

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 12, 2022

Z. Ali  
C. Filsfils  
P. Camarillo  
Cisco Systems  
D. Voyer  
Bell Canada  
S. Matsushima  
Softbank  
R. Rokui  
Nokia  
July 12, 2021

Building blocks for Network Slice Realization  
in Segment Routing Network

draft-ali-teas-spring-ns-building-blocks-01.txt

Abstract

This document describes how to realize the IETF network slice using the Segment Routing based technology. It explains how the building blocks specified for the Segment Routing can be used for this purpose.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 12, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.



## Table of Contents

|      |  |    |
|------|--|----|
| 1    | Introduction.....                                      | 2  |
| 2    | Segment Routing Policy.....                            | 3  |
| 2.1  | Flex-Algorithm Based SR Policies .....                 | 4  |
| 2.2  | On-demand SR policy .....                              | 5  |
| 2.3  | Automatic Steering .....                               | 6  |
| 2.4  | Inter-domain Considerations .....                      | 6  |
| 3    | TI-LFA and Microloop Avoidance.....                    | 7  |
| 4    | SR VPN.....  | 7  |
| 5    | Stateless Service Programming.....                     | 7  |
| 6    | Operations, Administration, and Maintenance (OAM)..... | 8  |
| 7    | QoS.....   | 8  |
| 8    | Stateless Network Slice Identification.....            | 9  |
| 8.1  | Stateless Slice Identification in SRv6.....            | 9  |
| 8.2  | Stateless Slice Identification in SR-MPLS.....         | 10 |
| 8    | IETF Network Slice Controller (NSC).....               | 10 |
| 9    | Illustration.....                                      | 10 |
| 10   | Security Considerations .....                          | 10 |
| 11   | IANA Considerations .....                              | 11 |
| 12   | References .....                                       | 11 |
| 12.1 | Normative References .....                             | 11 |
| 13   | Acknowledgments .....                                  | 11 |
| 14   | Contributors .....                                     | 11 |

## 1 Introduction

As more and more Service Providers and Enterprises operate a single network infrastructure to support an ever-increasing number of services, the ability to custom fit transport to application needs is critically important. This includes creating network slices with different characteristics can coexist on top of the shared network infrastructure.

Network Slicing is meant to create (end-to-end) partitioned network infrastructure that can be used to provide differentiated connectivity behaviors to fulfill the requirements of a diverse set of services. Services belonging to different Network slices can be wholly disjoint or can share different parts of the network infrastructure.

The definition of network slice for use within the IETF and the characteristics of IETF network slice are specified in [I-D.ietf-teas-ietf-network-slice-definition]. A framework for reusing IETF VPN and traffic-engineering technologies to realize IETF network slices is discussed in [I-D.ietf-teas-ietf-network-slices]. These documents also discuss the function of an IETF Network Slice Controller and the requirements on its northbound and southbound interfaces.

Segment Routing enables Service Providers to support realization of the Network Slicing in IP/MPLS transport network. The network as a whole, in a distributed and entirely automated manner, can share a single infrastructure resource along multiple virtual services (slices). For example, one IETF network slice is optimized continuously for low-cost

transport; a second one is optimized continuously for low-latency  
transport; a third one is orchestrated to support disjoint

services, etc. The optimization objective of each of these slices is programmable by the operator.

The Segment Routing specification already contains the various building blocks required to create network slices. This includes the following.

- . SR Policy with or without Flexible Algorithm.
- . TI-LFA with O(50 msec) protection in the slice underlay.
- . SR VPN.
- . SR Service Programming (NFV, SFC).
- . Operation, Administration and Management (OAM) and Performance Management (PM).
- . QoS using DiffServ.
- . Stateless Network Slice Identification
- . Orchestration at the Controller.

Each of these building blocks works independently of each other. Their functionality can be combined to satisfy service provider's requirement for the Network Slicing. An external controller plays an important role to orchestrate these building blocks into a Slicing service (see I-D.ietf-teas-ietf-network-slice-definition)).

This document elaborates on the attributes of each of these building blocks for Network Slicing in IP and/or MPLS underlay network. The document also highlights how each IETF network Slice can benefit from traffic engineering, network function virtualization/ service chaining (service programming), OAM, performance management, SDN readiness, O (50 msec) TI-LFA protection, etc. features of SR while respecting resource partitioning employed over the common networking infrastructure.

The document equally applicable to the SR-MPLS and SRv6 instantiations of segment routing.

The following subsection elaborates on each of these build blocks.

## 2 Segment Routing Policy

Segment Routing (SR) allows a headend node to steer a packet flow along any path without creating intermediate per-flow states [I-D.ietf-spring-segment-routing-policy]. The headend node steers a flow into a Segment Routing Policy (SR Policy). I.e., the SR Policy can be used to steer traffic along any arbitrary path in the network. This allows operators to enforce low-latency and / or disjoint paths, regardless of the normal forwarding paths.

The SR policy is able to support various optimization objectives [I-D.draft-filsfils-spring-sr-policy-considerations]. The optimization objectives can be instantiated for the IGP metric ([RFC1195] [RFC2328] [RFC5340]) xor the TE metric ([RFC5305], [RFC3630]) xor the latency extended TE metric ([RFC7810] [RFC7471]). In addition, an SR policy is able to various constraints, including inclusion and/or exclusion of TE affinity, inclusion and/or exclusion of IP address, inclusion and/or exclusion of SRLG, inclusion and/or exclusion of admin-tag, maximum accumulated metric (IGP, TE, and latency), maximum number of SIDs in the solution SID-List, maximum number of weighted SID-Lists in the solution set, diversity to another service instance (e.g., link, node, or SRLG disjoint paths originating from different head-ends), etc. [I-D.draft-filsfils-spring-sr-policy-considerations]. The supports for various optimization objectives and constraints enables SR policy to create Slices in the network.

SR policy can be instantiated with or without IGP Flexible Algorithm feature. The following subsection describes the SR Flexible Algorithm feature and how SR policy can utilize this feature.

## 2.1 Flex-Algorithm

Flexible Algorithm enriches the SR Policy solution by adding additional segments having different properties than the IGP Prefix segments. Flex Algo adds flexible, user-defined segments to the SRTE toolbox. Specifically, it allows for association of the "intent" to Prefix SIDs. [I-D.ietf-lsr-flex-algo] defines the IGP based Flex-Algorithm solution which allows IGPs themselves to compute paths constraint by the "intent" represented by the Flex-Algorithm.

The Flex-Algorithm has the following attributes:

- . Algorithm associate to the SID a specific TE intent expressed as an optimization objective (an algorithm) [I-D.ietf-lsr-flex-algo].
- . Flexibility includes the ability of network operators to define the intent of each algorithm they implement.
- . By design the mapping between the Flex-Algorithm and its meaning is flexible and is defined by the user.
- . Flexibility also includes ability for operators to make the decision to exclude some specific links from the shortest path computation, e.g.,

- o operator 1 may define Algo 128 to compute the shortest path for TE metric and exclude red affinity links.
- o operator 2 may define Algo 128 to compute the shortest path for latency metric and exclude blue affinity links.

An IETF Network Slice can be realized by associating of a Flexible-Algorithm value with the Slice via IETF network slice controller (NSC).

Flex Alg leverages SR on-demand next hop (ODN) and Automated Steering for intent-based instantiation of traffic engineered paths described in the following sub-sections. Specifically, as specified in [RFC8402] the IGP Flex Algo Prefix SIDs can also be used as segments within SR Policies thereby leveraging the underlying IGP Flex Algo solution.

## 2.2 On-demand SR policy

Segment Routing On-Demand Next-hop (ODN) functionality enables on-demand creation of SR Policies for service traffic. Using a Path Computation Element (PCE), end-to-end SR Policy paths can be computed to provide end-to-end Segment Routing connectivity, even in multi-domain networks running with or without IGP Flexible-Algorithm [I-D.draft-ietf-spring-segment-routing-policy].

The On-Demand Next-hop functionality provides optimized service paths to meet customer and application SLAs (such as latency, disjointness) without any pre-configured TE tunnel and with the automatic steering of the service traffic on the SR Policy without a static route, autoroute-announce, or policy-based routing.

With this functionality, the IETF Network Slice Controller can realize the IETF network slice based on their requirements. The head-end router requests the PCE to compute the path for the service and then instantiates an SR Policy with the computed path and steers the service traffic into that SR Policy. If the topology changes, the stateful PCE updates the SR Policy path. This happens seamlessly, while TI-LFA protects the traffic in case the topology change happened due to a failure.

### 2.3 Automatic Steering

Automatically steering traffic into an IETF Network Slice is one of the fundamental requirement for Slicing. That is made possible by the "Automated Steering" functionality of SR. Specifically, SR policy can be used for traffic engineer paths within a slice, "automatically steer" traffic to the right slice and connect IGP Flex-Algorithm domains sharing the same "intent".

A headend can steer a packet flow into a valid SR Policy within a slice in various ways [I-D.draft-ietf-spring-segment-routing-policy]:

- . Incoming packets have an active SID matching a local Binding SID (BSID) at the headend.
- . Per-destination Steering: incoming packets match a BGP/Service route which recurses on an SR policy.
- . Per-flow Steering: incoming packets match or recurse on a forwarding array of where some of the entries are SR Policies.
- . Policy-based Steering: incoming packets match a routing policy which directs them on an SR policy.

### 2.4 Inter-domain Considerations

The network slicing needs to be extended across multiple domains such that each domain can satisfy the intent consistently. SR has native inter-domain mechanisms, e.g., SR policies are designed to span multiple domains using a PCE based solution [I-D.ietf-spring-segment-routing], [I-D.ietf-spring-segment-routing-central-epe]. An edge router upon service configuration automatically requests to the Segment Routing PCE an inter-domain path to the remote service endpoint. The path can either be for simple best-effort inter-domain reachability or for reachability with an SLA contract and can be restricted to a Network Slice.

The SR native mechanisms for inter-domain are easily extendable to include the case when different IGP Flex-Algorithm values are used to represent the same intent. E.g., in domain1 Service Provider 1 (SP1) may use flex-algo 128 to indicate low latency Slice and in domain2 Service Provider 2 (SP2) may use flex-algo 129 to indicate low latency Slice. When an automation system at a PE1 in SP1 network configures a service with next hop (PE2) in SP2 network, SP1 contacts a Path Computation Element (PCE) to find a route to PE2. In the request, the PE1 also indicates the intent (i.e., the Flex-Algo 128) in the PCEP message. As the PCE has a complete understanding of both Domains, it can understand the path computation in Domain1 needs to be performed for Algorithm 128 and path computation in Domain2 needs to be



performed for Algorithm 129 (i.e., in the Low Latency Network Slice in both domains).

### 3 TI-LFA and Microloop Avoidance

The Segment Routing-based fast-reroute solution, TI-LFA, can provide per-destination sub-50msec protection upon any single link, node or SRLG failure regardless of the topology. The traffic is rerouted straight to the post-convergence path, hence avoiding any intermediate flap via an intermediate path. The primary and backup path computation is completely automatic by the IGP.

[I-D.draft-bashandy-rtgwg-segment-routing-ti-lfa] proposes a Topology Independent Loop-free Alternate Fast Re-route (TI-LFA), aimed at protecting node and adjacency segments within  $O(50 \text{ msec})$  in the Segment Routing networks. Furthermore, [I-D. draft-bashandy-rtgwg-segment-routing-uloop] provides a mechanism leveraging Segment Routing to ensure loop-freeness during the IGP reconvergence process following a link-state change event.

As mentioned earlier, Network Slicing in Segment Routing works seamlessly with all the other components of the Segment Routing. This, of course, includes TI-LFA and microloop avoidance within a Slice, with the added benefit that backup path only uses resources available to the Slice. For example, when Flexible Algorithm is used, the TI-LFA backup path computation is performed such that it is optimized per Flexible-Algorithm. The backup path shares the same properties as the primary path. The backup path does not use a resource outside the Slice of the primary path it is protecting.

### 4 SR VPN

Virtual Private Networks (VPNs) provides a mean for creating a logically separated network to a different set of users access to a common network. Segment Routing is equipped with the rich multi-service virtual private network (VPN) capabilities, including Layer 3 VPN (L3VPN), Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS), and Ethernet VPN (EVPN). The ability of Segment Routing to support different VPN technologies is one of the fundamental building blocks for creating slicing an SR network.

## 5 Stateless Service Programming

An important part of an IETF Network Slicing is the orchestration of virtualized service containers. [I-D.draft-xuclad-spring-sr-service-chaining] describes how to implement service segments and achieve stateless service programming in SR-MPLS and SRv6 networks. It introduces the notion of service segments. The ability of encoding the service segments along with the topological segment enables service providers to forward packets along a specific network path, but also steer them through VNFs or physical service appliances available in the network.

In an SR network, each of the service, running either on a physical appliance or in a virtual environment, is associated with a segment identifier (SID) for the service. These service SIDs are then leveraged as part of a SID-list to steer packets through the corresponding services. Service SIDs may be combined with topological SIDs to achieve service programming while steering the traffic through a specific topological path in the network. In this fashion, SR provides a fully integrated solution for overlay, underlay and service programming building blocks needed to satisfy network slicing requirements.

## 6 Operations, Administration, and Maintenance (OAM)

There are various OAM elements that are critical to satisfy Network Slicing requirements. These includes but not limited to the following:

- . Measuring per-link TE Matric.
- . Flooding per-link TE Matric.
- . Taking TE Matric into account during path calculation.
- . Taking TE Matric bound into account during path calculation.
- . SLA Monitoring: Service Provider can monitor each SR Policy in a Slice to Monitor SLA offered by the Policy using technique described in [I-D.draft-gandhi-spring-udp-pm]. This includes monitoring end-to-end delays on all ECMP paths of the Policy as well as monitoring traffic loss on a Policy. Remedial mechanisms can be used to ensure that the SR policy conforms to the SLA contract.

## 7 QoS

Segment Routing relies on MPLS and IP Differentiated Services. Differentiated services enhancements are intended to enable scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. [RFC2475] defines

an architecture for implementing scalable service differentiation in the Internet. This architecture is composed of many functional elements implemented in network nodes, including a small set of per-hop forwarding behaviors, packet classification functions, and traffic conditioning functions including metering, marking, shaping, and policing.

The DiffServ architecture achieves scalability by implementing complex classification and conditioning functions only at network boundary nodes, and by applying per-hop behaviors to aggregates of traffic depending on the traffic marker. Specifically, the node at the ingress of the DiffServ domain conditions, classifies and marks the traffic into a limited number of traffic classes. The function is used to ensure that the slice's traffic conforms to the contract associated with the slice.

Per-hop behaviors are defined to permit a reasonably granular means of allocating buffer and bandwidth resources at each node among competing traffic streams. Specifically, per class scheduling and queuing control mechanisms are applied at each IP hop to the traffic classes depending on packet's marking. Techniques such as queue management and a variety of scheduling mechanisms are used to get the required packet behavior to meet the slice's SLA.

## 8 Stateless Network Slice Identification

Some use-cases require a slice identifier (SLID) in the packet to provide differentiated treatment of the packets belonging to different network slices.

The network slice instantiation using the SLID in the packet is required to work with the building blocks described in the previous sections. For example, the QoS/ DiffServ needs to be observed on a per slice basis. The slice identification needs to be topologically independent and stateless.

### 8.1 Stateless Slice Identification in SRv6

[I-D.draft-filsfils-spring-srv6-stateless-slice-id] describes a stateless encoding of slice identification in the outer IPv6 header of an SRv6 domain. As defined in RFC8754 [RFC8754], when an ingress PE receives a packet that traverses the SR domain, it encapsulates the packet in an outer IPv6 header and an optional SRH. Based on a local policy of the SR domain, the Flow Label field of the outer IPv6 header carries the SLID. Specifically, the SLID is added in the 8 most significant bits of the Flow Label field of the outer IPv6 header. The remaining 12 bits of the Flow Label field are set as described in section 5.5 of [RFC8754] for inter-domain packets. Based on the local policy of the SR domain, the draft also uses one of the bits in the Traffic Class field of the outer IPv6 header to indicate that the entropy label contains the SLID.

The network slicing mechanism described in [I-D.draft-filsfils-spring-srv6-stateless-slice-id] works seamlessly with the building blocks described in the previous sections. For example, the slice identification is independent of topology and the network's QoS/DiffServ policy. It enables scalable network slicing for SRv6 overlays.

## 8.2 Stateless Slice Identification in SR-MPLS

[I-D.draft-decraene-mpls-slid-encoded-entropy-label-id] describes a similar stateless encoding of slice identification in the SR-MPLS domain. Specifically, the document extends the use of the Entropy Label to carry the SLID. The number of bits to be used for encoding the SLID in the Entropy Label is governed by a local policy of the SR domain. Based on the local policy of the SR domain, the draft uses one of the bits in the TTL field of the Entropy Label to indicate that the Entropy Label contains the SLID.

The network slicing mechanism described in [I-D.draft-decraene-mpls-slid-encoded-entropy-label-id] works seamlessly with the building blocks described in the previous sections. For example, the slice identification is independent of topology and the network's QoS/DiffServ policy. It enables scalable network slicing for SR-MPLS overlays.

## 8 IETF Network Slice Controller (NSC)

The role of IETF Network Slice Controller (NSC) is described in I-D.ietf-teas-ietf-network-slice-definition]. It plays a vital role in realization the IETF network slice using the SR building blocks discussed above. The NSC also performs admission control and traffic placement for slice management at the transport layer.

The SDN friendliness of the SR technology becomes handy to realize the orchestration. The controller may use PCEP or Netconf to interact with the routers. The router implements Yang model for SR-based network slicing.

Specification of the controller technology for orchestrating Network Slices, services and admission control for the services is outside the scope of this draft.

## 9 Illustration

To be added in a later revision.

## 10 Security Considerations

This document does not impose any additional security challenges.

## 11 IANA Considerations

This document does not define any new protocol or any extension to an existing protocol.

## 12 References

### 12.1 Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

### 7.2. Informative References

- [I-D.ietf-teas-ietf-network-slices] Farrel, A., et al, "Framework for IETF Network Slices", draft-ietf-teas-ietf-network-slices (work in progress)
- [I-D.ietf-teas-ietf-network-slice-definition] Rokui, R. et al, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition (work in progress).
- [I-D.ietf-spring-segment-routing-policy] Filsfils, C., Sivabalan, et al, "Segment Routing Policy For Traffic Engineering", draft-ietf-spring-segment-routing-policy (work in progress).
- [I-D.ietf-lsr-flex-algo] P. Psenak, et al, draft-ietf-lsr-flex-algo, work in progress.
- [RFC8402] Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC8402.
- [I-D.draft-filsfils-spring-sr-policy-considerations] Filsfils, C., et al. draft-filsfils-spring-sr-policy-considerations (work in progress)
- [RFC8754] Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-16 (work in progress), February 2019.
- [I-D.draft-filsfils-spring-srv6-stateless-slice-id] Filsfils, C., et al. draft-filsfils-spring-srv6-stateless-slice-id, work in progress.
- [I-D.draft-decraene-mpls-slid-encoded-entropy-label-id] Decraene B., Filsfils, C., Henderickx W., Saad T., Beeram V., work in progress.

13 Acknowledgments

14 Contributors

Francois Clad  
Cisco Systems, Inc.  
fclad@cisco.com

Internet-Draft      Network Slice Realization in SR Network

Authors' Addresses

Zafar Ali  
Cisco Systems, Inc.  
Email: zali@cisco.com

Clarence Filsfils  
Cisco Systems, Inc.  
Email: cf@cisco.com



Pablo Camarillo Garvia  
Cisco Systems, Inc.  
Email: [pcamaril@cisco.com](mailto:pcamaril@cisco.com)

Daniel Voyer  
Bell Canada  
Email: daniel.voyer@bell.ca

Satoru Matsushima  
Softbank  
Email: satoru.matsushima@g.softbank.co.jp

Reza Rokui  
Nokia  
Canada  
Email: reza.rokui@nokia.com

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 2, 2022

Z. Ali  
C. Filsfils  
P. Camarillo  
Cisco Systems  
D. Voyer  
Bell Canada  
S. Matsushima  
Softbank  
R. Rokui  
Ciena  
A. Dhamija  
Rakuten  
P. Maheshwari  
Airtel  
February 3, 2022

Building blocks for Network Slice Realization  
in Segment Routing Network

draft-ali-teas-spring-ns-building-blocks-02.txt

Abstract

This document describes how to realize the IETF network slice using the Segment Routing based technology. It explains how the building blocks specified for the Segment Routing can be used for this purpose.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 2, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.



## Table of Contents

|      |  |    |
|------|--|----|
| 1    | Introduction.....                                      | 2  |
| 2    | Segment Routing Policy.....                            | 3  |
| 2.1  | Flex-Algorithm Based SR Policies .....                 | 4  |
| 2.2  | On-demand SR policy .....                              | 5  |
| 2.3  | Automatic Steering .....                               | 6  |
| 2.4  | Inter-domain Considerations .....                      | 6  |
| 3    | TI-LFA and Microloop Avoidance.....                    | 7  |
| 4    | SR VPN.....  | 7  |
| 5    | Stateless Service Programming.....                     | 7  |
| 6    | Operations, Administration, and Maintenance (OAM)..... | 8  |
| 7    | QoS.....   | 8  |
| 8    | Stateless Network Slice Identification.....            | 9  |
| 8.1  | Stateless Slice Identification in SRv6.....            | 9  |
| 8.2  | Stateless Slice Identification in SR-MPLS.....         | 10 |
| 8    | IETF Network Slice Controller (NSC).....               | 10 |
| 9    | Illustration.....                                      | 10 |
| 10   | Security Considerations .....                          | 10 |
| 11   | IANA Considerations .....                              | 11 |
| 12   | References .....                                       | 11 |
| 12.1 | Normative References .....                             | 11 |
| 13   | Acknowledgments .....                                  | 11 |
| 14   | Contributors .....                                     | 11 |

## 1 Introduction

As more and more Service Providers and Enterprises operate a single network infrastructure to support an ever-increasing number of services, the ability to custom fit transport to application needs is critically important. This includes creating network slices with different characteristics can coexist on top of the shared network infrastructure.

Network Slicing is meant to create (end-to-end) partitioned network infrastructure that can be used to provide differentiated connectivity behaviors to fulfill the requirements of a diverse set of services. Services belonging to different Network slices can be wholly disjoint or can share different parts of the network infrastructure.

The definition of network slice for use within the IETF and the characteristics of IETF network slice are specified in [I-D.ietf-teas-ietf-network-slice-definition]. A framework for reusing IETF VPN and traffic-engineering technologies to realize IETF network slices is discussed in [I-D.ietf-teas-ietf-network-slices]. These documents also discuss the function of an IETF Network Slice Controller and the requirements on its northbound and southbound interfaces.

Segment Routing enables Service Providers to support realization of the Network Slicing in IP/MPLS transport network. The network as a whole, in a distributed and entirely automated manner, can share a single infrastructure resource along multiple virtual services (slices). For example, one IETF network slice is optimized continuously for low-cost

transport; a second one is optimized continuously for low-latency  
transport; a third one is orchestrated to support disjoint

services, etc. The optimization objective of each of these slices is programmable by the operator.

The Segment Routing specification already contains the various building blocks required to create network slices. This includes the following.

- . SR Policy with or without Flexible Algorithm.
- . TI-LFA with O(50 msec) protection in the slice underlay.
- . SR VPN.
- . SR Service Programming (NFV, SFC).
- . Operation, Administration and Management (OAM) and Performance Management (PM).
- . QoS using DiffServ.
- . Stateless Network Slice Identification
- . Orchestration at the Controller.

Each of these building blocks works independently of each other. Their functionality can be combined to satisfy service provider's requirement for the Network Slicing. An external controller plays an important role to orchestrate these building blocks into a Slicing service (see I-D.ietf-teas-ietf-network-slice-definition)).

This document elaborates on the attributes of each of these building blocks for Network Slicing in IP and/or MPLS underlay network. The document also highlights how each IETF network Slice can benefit from traffic engineering, network function virtualization/ service chaining (service programming), OAM, performance management, SDN readiness, O (50 msec) TI-LFA protection, etc. features of SR while respecting resource partitioning employed over the common networking infrastructure.

The document equally applicable to the SR-MPLS and SRv6 instantiations of segment routing.

The following subsection elaborates on each of these build blocks.

## 2 Segment Routing Policy

Segment Routing (SR) allows a headend node to steer a packet flow along any path without creating intermediate per-flow states [I-D.ietf-spring-segment-routing-policy]. The headend node steers a flow into a Segment Routing Policy (SR Policy). I.e., the SR Policy can be used to steer traffic along any arbitrary path in the network. This allows operators to enforce low-latency and / or disjoint paths, regardless of the normal forwarding paths.

The SR policy is able to support various optimization objectives [I-D.draft-filsfils-spring-sr-policy-considerations]. The optimization objectives can be instantiated for the IGP metric ([RFC1195] [RFC2328] [RFC5340]) xor the TE metric ([RFC5305], [RFC3630]) xor the latency extended TE metric ([RFC7810] [RFC7471]). In addition, an SR policy is able to various constraints, including inclusion and/or exclusion of TE affinity, inclusion and/or exclusion of IP address, inclusion and/or exclusion of SRLG, inclusion and/or exclusion of admin-tag, maximum accumulated metric (IGP, TE, and latency), maximum number of SIDs in the solution SID-List, maximum number of weighted SID-Lists in the solution set, diversity to another service instance (e.g., link, node, or SRLG disjoint paths originating from different head-ends), etc. [I-D.draft-filsfils-spring-sr-policy-considerations]. The supports for various optimization objectives and constraints enables SR policy to create Slices in the network.

SR policy can be instantiated with or without IGP Flexible Algorithm feature. The following subsection describes the SR Flexible Algorithm feature and how SR policy can utilize this feature.

## 2.1 Flex-Algorithm

Flexible Algorithm enriches the SR Policy solution by adding additional segments having different properties than the IGP Prefix segments. Flex Algo adds flexible, user-defined segments to the SRTE toolbox. Specifically, it allows for association of the "intent" to Prefix SIDs. [I-D.ietf-lsr-flex-algo] defines the IGP based Flex-Algorithm solution which allows IGPs themselves to compute paths constraint by the "intent" represented by the Flex-Algorithm.

The Flex-Algorithm has the following attributes:

- . Algorithm associate to the SID a specific TE intent expressed as an optimization objective (an algorithm) [I-D.ietf-lsr-flex-algo].
- . Flexibility includes the ability of network operators to define the intent of each algorithm they implement.
- . By design the mapping between the Flex-Algorithm and its meaning is flexible and is defined by the user.
- . Flexibility also includes ability for operators to make the decision to exclude some specific links from the shortest path computation, e.g.,



- o operator 1 may define Algo 128 to compute the shortest path for TE metric and exclude red affinity links.
- o operator 2 may define Algo 128 to compute the shortest path for latency metric and exclude blue affinity links.

An IETF Network Slice can be realized by associating of a Flexible-Algorithm value with the Slice via IETF network slice controller (NSC).

Flex Alg leverages SR on-demand next hop (ODN) and Automated Steering for intent-based instantiation of traffic engineered paths described in the following sub-sections. Specifically, as specified in [RFC8402] the IGP Flex Algo Prefix SIDs can also be used as segments within SR Policies thereby leveraging the underlying IGP Flex Algo solution.

## 2.2 On-demand SR policy

Segment Routing On-Demand Next-hop (ODN) functionality enables on-demand creation of SR Policies for service traffic. Using a Path Computation Element (PCE), end-to-end SR Policy paths can be computed to provide end-to-end Segment Routing connectivity, even in multi-domain networks running with or without IGP Flexible-Algorithm [I-D.draft-ietf-spring-segment-routing-policy].

The On-Demand Next-hop functionality provides optimized service paths to meet customer and application SLAs (such as latency, disjointness) without any pre-configured TE tunnel and with the automatic steering of the service traffic on the SR Policy without a static route, autoroute-announce, or policy-based routing.

With this functionality, the IETF Network Slice Controller can realize the IETF network slice based on their requirements. The head-end router requests the PCE to compute the path for the service and then instantiates an SR Policy with the computed path and steers the service traffic into that SR Policy. If the topology changes, the stateful PCE updates the SR Policy path. This happens seamlessly, while TI-LFA protects the traffic in case the topology change happened due to a failure.

### 2.3 Automatic Steering

Automatically steering traffic into an IETF Network Slice is one of the fundamental requirement for Slicing. That is made possible by the "Automated Steering" functionality of SR. Specifically, SR policy can be used for traffic engineer paths within a slice, "automatically steer" traffic to the right slice and connect IGP Flex-Algorithm domains sharing the same "intent".

A headend can steer a packet flow into a valid SR Policy within a slice in various ways [I-D.draft-ietf-spring-segment-routing-policy]:

- . Incoming packets have an active SID matching a local Binding SID (BSID) at the headend.
- . Per-destination Steering: incoming packets match a BGP/Service route which recurses on an SR policy.
- . Per-flow Steering: incoming packets match or recurse on a forwarding array of where some of the entries are SR Policies.
- . Policy-based Steering: incoming packets match a routing policy which directs them on an SR policy.

### 2.4 Inter-domain Considerations

The network slicing needs to be extended across multiple domains such that each domain can satisfy the intent consistently. SR has native inter-domain mechanisms, e.g., SR policies are designed to span multiple domains using a PCE based solution [I-D.ietf-spring-segment-routing], [I-D.ietf-spring-segment-routing-central-epe]. An edge router upon service configuration automatically requests to the Segment Routing PCE an inter-domain path to the remote service endpoint. The path can either be for simple best-effort inter-domain reachability or for reachability with an SLA contract and can be restricted to a Network Slice.

The SR native mechanisms for inter-domain are easily extendable to include the case when different IGP Flex-Algorithm values are used to represent the same intent. E.g., in domain1 Service Provider 1 (SP1) may use flex-algo 128 to indicate low latency Slice and in domain2 Service Provider 2 (SP2) may use flex-algo 129 to indicate low latency Slice. When an automation system at a PE1 in SP1 network configures a service with next hop (PE2) in SP2 network, SP1 contacts a Path Computation Element (PCE) to find a route to PE2. In the request, the PE1 also indicates the intent (i.e., the Flex-Algo 128) in the PCEP message. As the PCE has a complete understanding of both Domains, it can understand the path computation in Domain1 needs to be performed for Algorithm 128 and path computation in Domain2 needs to be

performed for Algorithm 129 (i.e., in the Low Latency Network Slice in both domains).

### 3 TI-LFA and Microloop Avoidance

The Segment Routing-based fast-reroute solution, TI-LFA, can provide per-destination sub-50msec protection upon any single link, node or SRLG failure regardless of the topology. The traffic is rerouted straight to the post-convergence path, hence avoiding any intermediate flap via an intermediate path. The primary and backup path computation is completely automatic by the IGP.

[I-D.draft-bashandy-rtgwg-segment-routing-ti-lfa] proposes a Topology Independent Loop-free Alternate Fast Re-route (TI-LFA), aimed at protecting node and adjacency segments within  $O(50 \text{ msec})$  in the Segment Routing networks. Furthermore, [I-D. draft-bashandy-rtgwg-segment-routing-uloop] provides a mechanism leveraging Segment Routing to ensure loop-freeness during the IGP reconvergence process following a link-state change event.

As mentioned earlier, Network Slicing in Segment Routing works seamlessly with all the other components of the Segment Routing. This, of course, includes TI-LFA and microloop avoidance within a Slice, with the added benefit that backup path only uses resources available to the Slice. For example, when Flexible Algorithm is used, the TI-LFA backup path computation is performed such that it is optimized per Flexible-Algorithm. The backup path shares the same properties as the primary path. The backup path does not use a resource outside the Slice of the primary path it is protecting.

### 4 SR VPN

Virtual Private Networks (VPNs) provides a mean for creating a logically separated network to a different set of users access to a common network. Segment Routing is equipped with the rich multi-service virtual private network (VPN) capabilities, including Layer 3 VPN (L3VPN), Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS), and Ethernet VPN (EVPN). The ability of Segment Routing to support different VPN technologies is one of the fundamental building blocks for creating slicing an SR network.

## 5 Stateless Service Programming

An important part of an IETF Network Slicing is the orchestration of virtualized service containers. [I-D.draft-xuclad-spring-sr-service-chaining] describes how to implement service segments and achieve stateless service programming in SR-MPLS and SRv6 networks. It introduces the notion of service segments. The ability of encoding the service segments along with the topological segment enables service providers to forward packets along a specific network path, but also steer them through VNFs or physical service appliances available in the network.

In an SR network, each of the service, running either on a physical appliance or in a virtual environment, is associated with a segment identifier (SID) for the service. These service SIDs are then leveraged as part of a SID-list to steer packets through the corresponding services. Service SIDs may be combined with topological SIDs to achieve service programming while steering the traffic through a specific topological path in the network. In this fashion, SR provides a fully integrated solution for overlay, underlay and service programming building blocks needed to satisfy network slicing requirements.

## 6 Operations, Administration, and Maintenance (OAM)

There are various OAM elements that are critical to satisfy Network Slicing requirements. These includes but not limited to the following:

- . Measuring per-link TE Matric.
- . Flooding per-link TE Matric.
- . Taking TE Matric into account during path calculation.
- . Taking TE Matric bound into account during path calculation.
- . SLA Monitoring: Service Provider can monitor each SR Policy in a Slice to Monitor SLA offered by the Policy using technique described in [I-D.draft-gandhi-spring-udp-pm]. This includes monitoring end-to-end delays on all ECMP paths of the Policy as well as monitoring traffic loss on a Policy. Remedial mechanisms can be used to ensure that the SR policy conforms to the SLA contract.

## 7 QoS

Segment Routing relies on MPLS and IP Differentiated Services. Differentiated services enhancements are intended to enable scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. [RFC2475] defines

an architecture for implementing scalable service differentiation in the Internet. This architecture is composed of many functional elements implemented in network nodes, including a small set of per-hop forwarding behaviors, packet classification functions, and traffic conditioning functions including metering, marking, shaping, and policing.

The DiffServ architecture achieves scalability by implementing complex classification and conditioning functions only at network boundary nodes, and by applying per-hop behaviors to aggregates of traffic depending on the traffic marker. Specifically, the node at the ingress of the DiffServ domain conditions, classifies and marks the traffic into a limited number of traffic classes. The function is used to ensure that the slice's traffic conforms to the contract associated with the slice.

Per-hop behaviors are defined to permit a reasonably granular means of allocating buffer and bandwidth resources at each node among competing traffic streams. Specifically, per class scheduling and queuing control mechanisms are applied at each IP hop to the traffic classes depending on packet's marking. Techniques such as queue management and a variety of scheduling mechanisms are used to get the required packet behavior to meet the slice's SLA.

## 8 Stateless Network Slice Identification

Some use-cases require a slice identifier (SLID) in the packet to provide differentiated treatment of the packets belonging to different network slices.

The network slice instantiation using the SLID in the packet is required to work with the building blocks described in the previous sections. For example, the QoS/ DiffServ needs to be observed on a per slice basis. The slice identification needs to be topologically independent and stateless.

### 8.1 Stateless Slice Identification in SRv6

[I-D.draft-filsfils-spring-srv6-stateless-slice-id] describes a stateless encoding of slice identification in the outer IPv6 header of an SRv6 domain. As defined in RFC8754 [RFC8754], when an ingress PE receives a packet that traverses the SR domain, it encapsulates the packet in an outer IPv6 header and an optional SRH. Based on a local policy of the SR domain, the Flow Label field of the outer IPv6 header carries the SLID. Specifically, the SLID is added in the 8 most significant bits of the Flow Label field of the outer IPv6 header. The remaining 12 bits of the Flow Label field are set as described in section 5.5 of [RFC8754] for inter-domain packets. Based on the local policy of the SR domain, the draft also uses one of the bits in the Traffic Class field of the outer IPv6 header to indicate that the entropy label contains the SLID.

The network slicing mechanism described in [I-D.draft-filsfils-spring-srv6-stateless-slice-id] works seamlessly with the building blocks described in the previous sections. For example, the slice identification is independent of topology and the network's QoS/DiffServ policy. It enables scalable network slicing for SRv6 overlays.

## 8.2 Stateless Slice Identification in SR-MPLS

[I-D.draft-decraene-mpls-slid-encoded-entropy-label-id] describes a similar stateless encoding of slice identification in the SR-MPLS domain. Specifically, the document extends the use of the Entropy Label to carry the SLID. The number of bits to be used for encoding the SLID in the Entropy Label is governed by a local policy of the SR domain. Based on the local policy of the SR domain, the draft uses one of the bits in the TTL field of the Entropy Label to indicate that the Entropy Label contains the SLID.

The network slicing mechanism described in [I-D.draft-decraene-mpls-slid-encoded-entropy-label-id] works seamlessly with the building blocks described in the previous sections. For example, the slice identification is independent of topology and the network's QoS/DiffServ policy. It enables scalable network slicing for SR-MPLS overlays.

## 8 IETF Network Slice Controller (NSC)

The role of IETF Network Slice Controller (NSC) is described in I-D.ietf-teas-ietf-network-slice-definition]. It plays a vital role in realization the IETF network slice using the SR building blocks discussed above. The NSC also performs admission control and traffic placement for slice management at the transport layer.

The SDN friendliness of the SR technology becomes handy to realize the orchestration. The controller may use PCEP or Netconf to interact with the routers. The router implements Yang model for SR-based network slicing.

Specification of the controller technology for orchestrating Network Slices, services and admission control for the services is outside the scope of this draft.

## 9 Illustration

To be added in a later revision.

## 10 Security Considerations

This document does not impose any additional security challenges.

## 11 IANA Considerations

This document does not define any new protocol or any extension to an existing protocol.

## 12 References

### 12.1 Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

### 7.2. Informative References

- [I-D.ietf-teas-ietf-network-slices] Farrel, A., et al, "Framework for IETF Network Slices", draft-ietf-teas-ietf-network-slices (work in progress)
- [I-D.ietf-teas-ietf-network-slice-definition] Rokui, R. et al, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition (work in progress).
- [I-D.ietf-spring-segment-routing-policy] Filsfils, C., Sivabalan, et al, "Segment Routing Policy For Traffic Engineering", draft-ietf-spring-segment-routing-policy (work in progress).
- [I-D.ietf-lsr-flex-algo] P. Psenak, et al, draft-ietf-lsr-flex-algo, work in progress.
- [RFC8402] Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC8402.
- [I-D.draft-filsfils-spring-sr-policy-considerations] Filsfils, C., et al. draft-filsfils-spring-sr-policy-considerations (work in progress)
- [RFC8754] Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-16 (work in progress), February 2019.
- [I-D.draft-filsfils-spring-srv6-stateless-slice-id] Filsfils, C., et al. draft-filsfils-spring-srv6-stateless-slice-id, work in progress.
- [I-D.draft-decraene-mpls-slid-encoded-entropy-label-id] Decraene B., Filsfils, C., Henderickx W., Saad T., Beeram V., work in progress.



13 Acknowledgments

14 Contributors

Francois Clad  
Cisco Systems, Inc.  
fclad@cisco.com

Internet-Draft      Network Slice Realization in SR Network

Authors' Addresses

Zafar Ali  
Cisco Systems, Inc.  
Email: zali@cisco.com

Clarence Filsfils  
Cisco Systems, Inc.  
Email: cf@cisco.com

Pablo Camarillo Garvia  
Cisco Systems, Inc.  
Email: [pcamaril@cisco.com](mailto:pcamaril@cisco.com)

Daniel Voyer  
Bell Canada  
Email: daniel.voyer@bell.ca

Satoru Matsushima  
Softbank  
Email: satoru.matsushima@g.softbank.co.jp

Reza Rokui  
Ciena  
Canada  
Email: rrokui@Ciena.com

Amit Dhamija  
Rakuten  
Email: amit.dhamija@rakuten.com

Praveen Maheshwari  
Airtel  
Email: Praveen.Maheshwari@airtel.com

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 13 January 2022

S. Barguil  
L.M. Contreras  
Telefonica  
V. Lopez  
R. Rokui  
Nokia  
O. Gonzalez de Dios  
Telefonica  
12 July 2021

Instantiation of IETF Network Slices in Service Providers Networks  
draft-barguil-teas-network-slices-instantiation-02

Abstract

Network Slicing (NS) is an integral part of Service Provider networks. The IETF has produced several YANG data models to support the Software-Defined Networking and network slice architecture and YANG-based service models for network slice (NS) instantiation.

This document describes the relationship between IETF Network Slice models for requesting the IETF Network Slices and (e.g., Layer-3 Service Model, Layer-2 Service Model) and Network Models (e.g., Layer-3 Network Model, Layer-2 Network Model) used during their realizations. In addition, this document describes the communication between the IETF Network Slice Controller and the network controllers for the realization of IETF network slices.

The IETF Network Slice YANG model provides the customer-oriented view of the network slice. Thus, once the IETF Network Slice controller (NSC) receives a request, it needs to map it to accomplish the specific parameters expected by the network controllers. The network models are analyzed to satisfy the IETF Network Slice requirements, and the gaps in existing models are reported.

The document also provides operational and security considerations when deploying network slices in Service Provider networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 13 January 2022.

#### Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .   | 3  |
| 1.1. Terminology . . . . .  | 3  |
| 2. Reference Architecture and Components . . . . .                                | 3  |
| 3. IETF Network Slice Requirements and Data Models . . . . .                      | 7  |
| 4. IETF Network Slice Procedure . . . . .   | 8  |
| 5. Network Controller Operation . . . . .   | 8  |
| 5.1. LxVPN Network Models . . . . .   | 9  |
| 5.2. Traffic Engineering Models . . . . .   | 9  |
| 5.3. Traffic Engineering Service Mapping . . . . .                                | 10 |
| 6. Operational Considerations . . . . .   | 10 |
| 6.1. Availability . . . . .   | 10 |
| 6.2. Downlink throughput / Uplink throughput. . . . .                             | 10 |
| 6.3. Protection scheme . . . . .  | 11 |
| 6.4. Delay . . . . .  | 11 |
| 6.5. Packet loss rate . . . . .   | 11 |
| 7. Network Slice Procedure . . . . .  | 12 |
| 7.1. IETF Network Slice requested to Hierarchical Network<br>Controller . . . . . | 13 |
| 7.2. IETF Network Slice requested to Network Slice<br>Controller . . . . .        | 14 |
| 7.3. Network Slice Controller as part of the domain<br>controller . . . . .       | 15 |
| 8. Security Considerations . . . . .  | 17 |
| 9. IANA Considerations . . . . .  | 18 |
| 10. Conclusions . . . . .   | 18 |

|                                    |    |
|------------------------------------|----|
| 11. Contributors . . . . .         | 18 |
| 12. Acknowledgements . . . . .     | 19 |
| 13. Normative References . . . . . | 19 |
| Authors' Addresses . . . . .       | 21 |

## 1. Introduction

The IETF has produced several YANG data models to support the Software-Defined Networking and network slice architecture.

The IETF Network Slice YANG service model provides the customer-oriented view of the network slice. Once the IETF Network Slice controller (NSC) receives a request, it needs to map it to accomplish the specific parameters expected by the network controller.

Several Service Models and Network Models, including Layer-3 Service Model (L3SM), Layer-2 Service Model (L2SM) and Network Models which may be utilized for IETF Network Slicing, are analyzed can satisfy the IETF Network Slice requirements. In addition, identified gaps on existing models are reported.

This document describes the architecture and communication process between the Network Slice Controller and a network controller for IETF network slice creation.

Editor's Note: the terminology in this draft will be aligned with the final terminology selected for describing the notion of IETF Network Slice when applied to IETF technologies, which is currently under discussion. By now same terminology as used in [I-D.ietf-teas-ietf-network-slice-definition] and [I-D.nsd-ietf-teas-ns-framework] is primarily used here. Consensus to use "IETF Network Slice" term has been reached.

### 1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

## 2. Reference Architecture and Components

As described in [I-D.ietf-teas-ietf-network-slice-definition], the IETF Network Slice Controller (NSC) is a functional entity for control and management of IETF network slices. As shown in Figure A, NSC from its North Bound Interface (NBI) exposes set of APIs that allow a higher level system to request an IETF network slice. The NSC NBI supports the request for enablement of an IETF Network Slice (i.e., creation, modification or deletion). Upon receiving a request

from its NBI, NSC finds the resources needed for realization of the IETF Network Slice and in turn interfaces from its South Bound Interface (SBI) with one or more Network Controllers for the realization of the requested IETF Network Slice.

This document focuses on how IETF Network Slice Controller (NSC) can be implemented in operator's network.

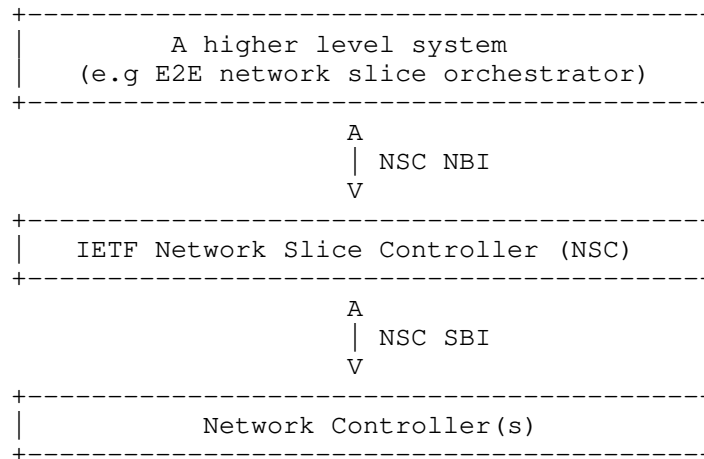


Figure 1 Network Slice Controller as a module of the Hierarchical SDN controller.

Several architectural definitions have arisen on the IETF to support SDN and network slicing deployments. The architectural proposal defined in [I-D.ietf-teas-ietf-network-slice-definition] includes a three-level hierarchy and expresses how each level relates with the ACTN architecture framework.

Figure 2 defines depicts a possible architecture using those concepts. It starts from a top consumer or high-level operational systems. Next, the IETF Network Slice Controller function might be part of the Hierarchical network controller (e.g., as the MDSC in the ACTN context [RFC8453]) as a modular function. At the bottom, two network controllers, each one can handle multiple or single underlay technologies.



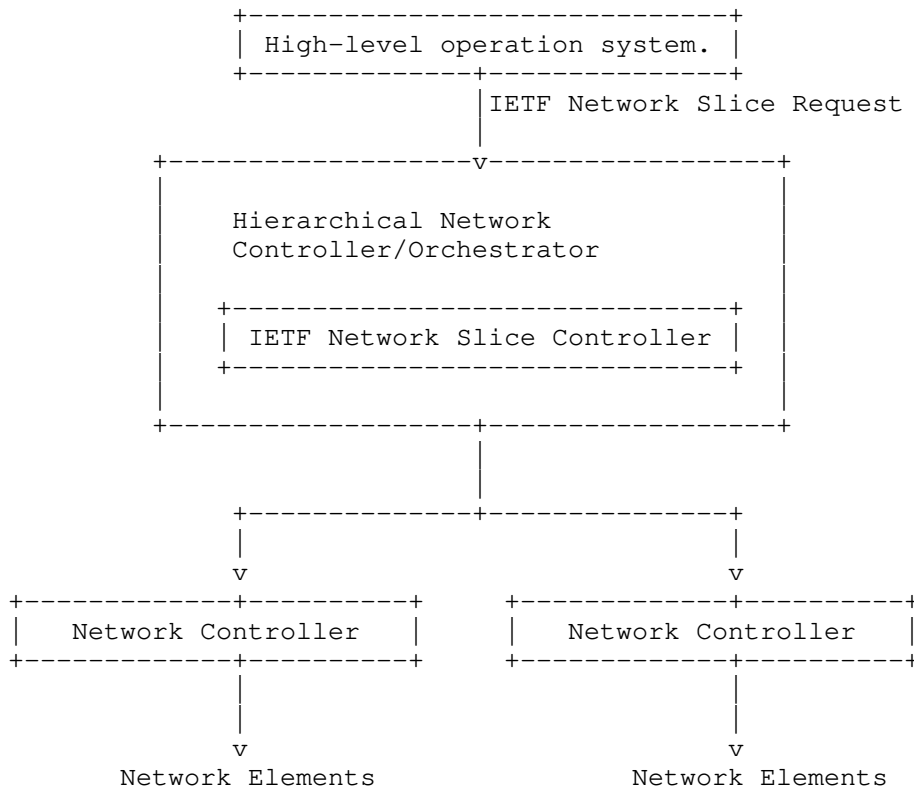


Figure 2 IETF Network Slice Controller as a module of the Hierarchical SDN controller.

In other implementations, the IETF Network Slice Controller can be a stand-alone element and directly interact with the network controller, as depicted in Figure 2. In this scenario, the services request follows a data-enrichment path, where each entity adds more information to the service request. This document describes how the available service models and network models interact to deliver the network slices in a service provider environment.

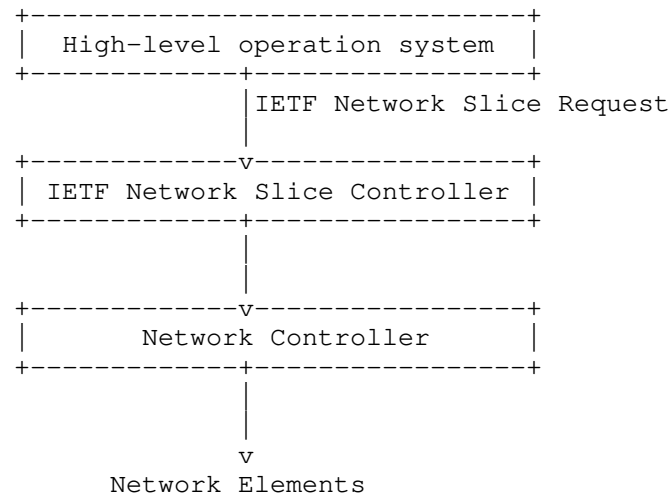


Figure 3 The IETF Network Slice Controller as a stand-alone entity.

As another implementation possibility, the IETF Network Slice Controller can be integrated with the Network controller and directly realize the network slice using device data models to configure the network devices. The sample architecture is depicted in Figure 4.

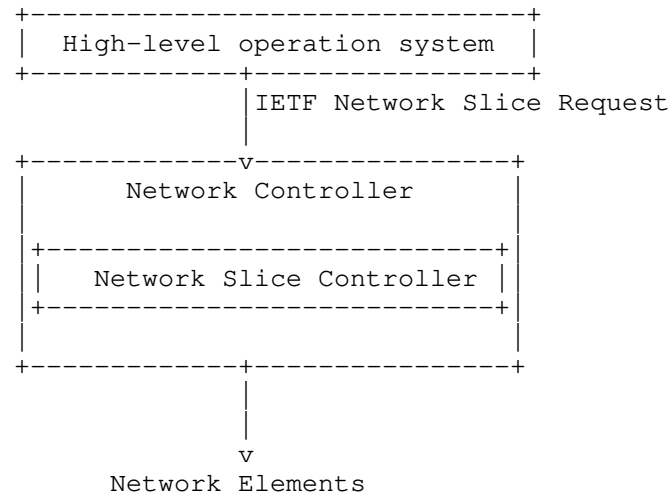


Figure 4 IETF Network Slice Controller as a module of the Network controller.

### 3. IETF Network Slice Requirements and Data Models

The main set of requirements for the IETF Slice, based on the high-level slice requirements from multiple organizations and use cases, are compiled in [I-D.contreras-teas-slice-nbi] and reproduced bellow the slice use cases reported:

|   |
|---|
| Network Slice Requirements for 5G service |
| Availability                              |
| Deterministic communication               |
| Downlink throughput per network slice     |
| Energy efficiency                         |
| Group communication support               |
| Isolation level                           |
| Maximum supported packet size             |
| Mission critical support                  |
| Performance monitoring                    |
| Slice quality of service parameters       |
| Support for non-IP traffic                |
| Uplink throughput per network slice       |
| User data access                          |
| Delay tolerance                           |
| NFV-based services                        |
| Incoming and outgoing bandwidth           |
| Qos metrics                               |
| Directionality                            |
| MTU                                       |
| Protection scheme                         |
| Connectivity mode                         |
| Network sharing                           |
| Maximum and Guaranteed Bit Rate           |
| Bounded latency                           |
| Packet loss rate                          |
| IP addressing                             |
| L2/L3 reachability                        |
| Recovery time                             |
| Secure connection                         |

To accomplish those requirements, a set of YANG data models have been proposed. Those Yang models, summarized in table xx, could be used by an IETF Network Slice Controller to manage CRUD operations on the IETF Network Slice. That is, these models aim capturing the requirements from the consumer of the slice point of view and avoid entering into the detail of how the slice is actually created.

- \* [draft-wd-teas-ietf-network-slice-nbi-yang]: A Yang Data Model for IETF Network Slice NBI.
- \* [draft-liu-teas-transport-network-slice-yang]: Transport Network Slice YANG Data Model.

#### 4. IETF Network Slice Procedure

An IETF Network Slice may use several underlying technologies. The creation of a new IETF Network Slice will be initiated with following three steps:

1. A higher level system requests connections with specific characteristics via the NBI.
2. This request will be processed by an IETF NSC which specifies a mapping between northbound request to any IETF Services, Tunnels, and paths models.
3. A series of requests for creation of services, tunnels and paths will be sent to the network to realize the transport slice.

#### 5. Network Controller Operation

As a functional entity responsible for managing a network domain, the network controller, can expose its northbound interface based on YANG models. The IETF Network Slice Controller can use the network controller's NBI during the realization of IETF Network Slice. The following network models can be used for realization of IETF Network slices:

- \* LxVPN Network models:
  - These models describe a VPN service from the network point of view. It supports the creation of Layer 3 and Layer 2 services using several control planes.
- \* Traffic Engineering models:

- These models allow to manipulate Traffic Engineering tunnels within the network segment. Technology-specific extensions allow to work with a desired technology (e.g. MPLS RSVP-TE tunnels, Segment Routing paths, OTN tunnels, etc.)
- \* TE Service Mapping extensions:
  - These extensions allow to specify for LxVPN the details of an underlay based on TE.
- \* ACLs and routing policies models:
  - Even though ACLs and routing policies are device models, it's exposure in the NBI of a domain controller allows to provide an additional granularity that the network domain controller is not able to infer on its own.

#### 5.1. LxVPN Network Models

The framework defined in [RFC8969] compiles a set of YANG data models for automating network services. The data models can be used during the service and network management life cycle (e.g., service instantiation, service provisioning, service optimization, service monitoring, service diagnosing, and service assurance). The so called Network models could be reused for the realization of Network slice requests.

The following models are examples of Network models that describe services.

- \* [I-D.ietf-opsawg-l3sm-l3nm]: A Layer 3 VPN Network YANG Model
- \* [I-D.ietf-opsawg-l2nm]: A Layer 2 VPN Network YANG Model

#### 5.2. Traffic Engineering Models

TEAS has defined a collection of models to allow the management of Traffic Engineering tunnels.

- \* [I-D.ietf-teas-yang-te]: A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces. The model allows to instantiate paths in a TE enabled network. Note that technology augmented models are require to particular per-technology instantiations.

### 5.3. Traffic Engineering Service Mapping

The IETF has defined a YANG model to set up the procedure to map VPN service/network models to the TE models. This model, known as service mapping, allows the network controller to assign/retrieve transport resources allocated to specific services. At the moment there is just one service mapping model [I-D.ietf-teas-te-service-mapping-yang]. The "Traffic Engineering (TE) and Service Mapping Yang Model" augments the VPN service and network models.

## 6. Operational Considerations

This section outlines the compliance and operational aspects of Network Controller models with IETF Network slice requirements. Section presented the requirements of the IETF Network slice. In this subsection it is analyzed how available YANG models that can be used by a Network Controller can satisfy those requirements and identify gaps.

### 6.1. Availability

As per [draft-ietf-teas-te-service-mapping-yang], Availability is a probabilistic measure of the length of time that a VPN/VN instance functions without a network failure. As per RFC 8330, The parameter "availability", as described in [G.827], [F.1703], and [P.530], is often used to describe the link capacity. The availability is a time scale, representing a proportion of the operating time that the requested bandwidth is ensured".

The calculation of the availability is not trivial and would need to be clearly scoped to avoid misunderstandings.

The set of Yang models proposed today allow to request tunnels/paths with different resiliency requirements in terms of protection and restoration. However, none of them include the possibility of requesting a specific availability (e.g. 99.9999%).

### 6.2. Downlink throughput / Uplink throughput.

The LxVPN Models ([I-D.ietf-opsawg-l3sm-l3nm] and [I-D.ietf-opsawg-l2nm]) allow to specify the bandwidth at the interface level between the slice and the customer. In addition, the Service Mapping model [draft-ietf-teas-te-service-mapping-yang] allows to bind a VPN to a given LSP, which have its bandwidth requirements. Additionally, TE models can force a give bandwidth in the connection between Provider Edges.

Previous comment applies to the incoming and outgoing bandwidth parameters required for the NFV-based services use case in [I-D.contreras-teas-slice-nbi]. The Network sharing use case has Maximum and Guaranteed Bit Rate parameters. These parameters can be mapped to the TE tunnel models when setting up LSPs [draft-ietf-teas-yang-te].

### 6.3. Protection scheme

Protection schemes are mechanisms to define how to setup resources for a given connection. TE tunnel models [draft-ietf-teas-yang-te] includes protection and restoration as two main attributes. The parameters included in the containers for protection and restoration cover the requirements of the IETF NS related with protection schemes. Similarly, TE models cover the parameter 'recovery time' for the network sharing use case.

### 6.4. Delay

Delay is a critical parameter for several IETF NS types. Every use-case defined in [I-D.contreras-teas-slice-nbi] contains delay constraints. 5G use cases require 'delay tolerance', NFV-based services have the delay information within 'QoS metrics' and 'Bounded latency' in the network sharing use case.

During the realization of the IETF Network Slice, these parameters are part of the requirements of a TE tunnel configuration [draft-ietf-teas-yang-te]. They can be included within the 'path-metric-bounds' parameter, so the created LSP fulfils the given metrics bounds like 'path-metric-delay-average' or 'path-metric-delay-minimum'.

### 6.5. Packet loss rate

The packet loss rate indicates the maximum rate for lost packets that the service tolerates in the link. During the realization of the IETF Network Slice, this attribute will influence the tunnel selection and the value is included in the [draft-ietf-teas-yang-te] document as the 'path-metric-loss'. The 'path-metric-loss' is a metric type, which measures the percentage of packet loss of all links traversed by a P2P path. This parameter is required for 5G services and network sharing use-case, while it is part of the 'QoS metrics' for the NFV-based services.

## 7. Network Slice Procedure

Draft [draft-contreras-teas-slice-controller-models] shows the internal structure of an IETF Network Slice Controller which can be divided into two components:

- \* IETF Network Slice Mapper: this high-level component processes the customer request, putting it into the context of the overall IETF Network Slices in the network.
- \* IETF Network Slice Realizer: this high-level component processes the complete view of transport slices including the one requested by the customer, decides the proper technologies for realizing the IETF Network Slice and triggers its realization.

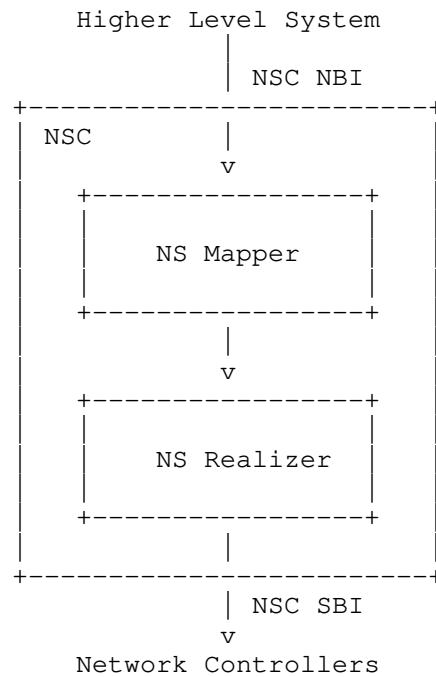


Figure 8: IETF Network Slice Controller Structure

The details of IETF network slice mapper and realize are provided below for various implementation of NCS.



### 7.1. IETF Network Slice requested to Hierarchical Network Controller

Referring to Figure 1 in an integrated architecture, the IETF Network Slice Controller (NCS) is part of a Hierarchical SDN controller module, the NSC's and the Hierarchical Network Controller should share the same internal data and the same NBI. Thus, the H-SDN module must be able to:

- \* **Map:** The customer request received using the [draft-wd-teas-ietf-network-slice-nbi-yang] must be processed by the NCS. The mapping process takes the network-slice SLAs selected by the customer to available Routing Policies and Forwarding policies.
- \* **Realize:** Create necessary network requests. The slice's realization can be translated into one or several LXNM Network requests, depending on the number of underlay controllers. Thus, the NCS must have a complete view of the network to map the orders and distribute them across domains. The realization should include the expansion/selection of Forwarding Policies, Routing Policies, VPN policies, and Underlay transport preference.

To maintain the data coherence between the control layers, the IETF Network Slice ID "ns-id" used of the [draft-wd-teas-ietf-network-slice-nbi-yang] must be directly mapped to the "transport-instance-id" at the VPN-Node level.

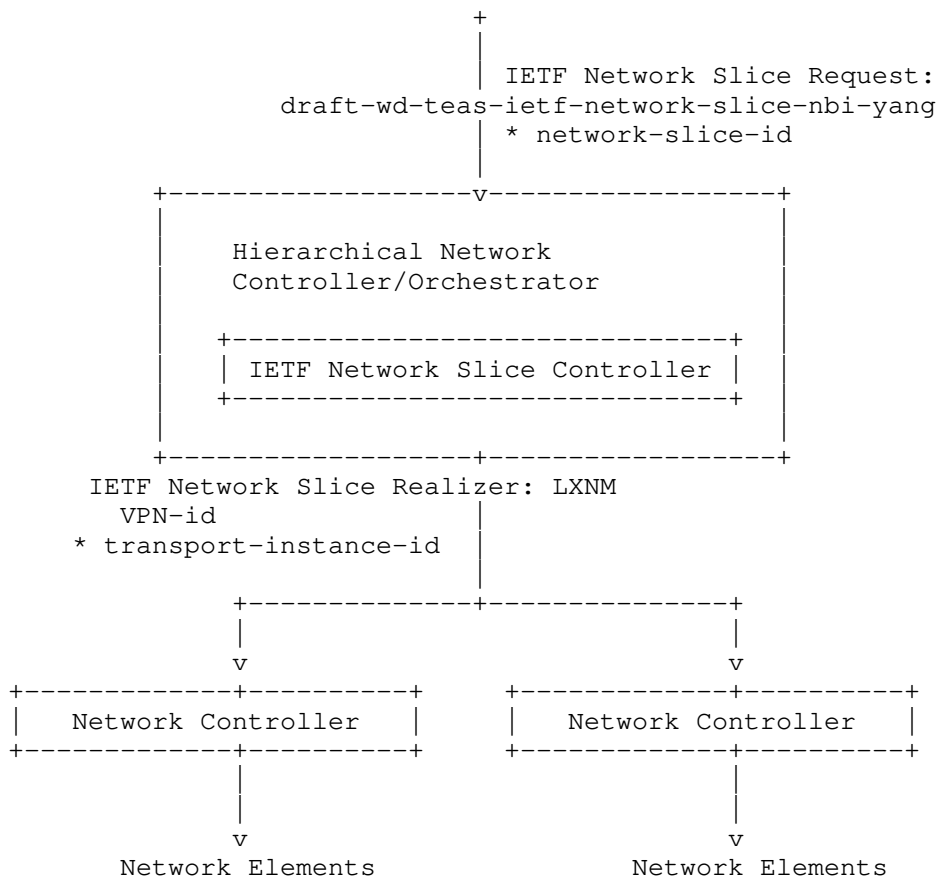


Figure 9 Workflow for the slice request in an integrated architecture.

## 7.2. IETF Network Slice requested to Network Slice Controller

Referring to Figure 2 when the Network Slice Controller is a stand-alone controller module, the NSC's should perform the same two tasks described in section 6.1:

- \* Map: Process the customer request. The customer request can be sent using the [draft-liu-teas-transport-network-slice-yang]. This draft allows the topology mapping of the Slice request.

- \* **Realize:** Create necessary network requests. The slice's realization will be translated into one LXNM Network request. As the NCS has a topological view of the network, the realization can include the customer's traffic engineering transport preferences and policies.

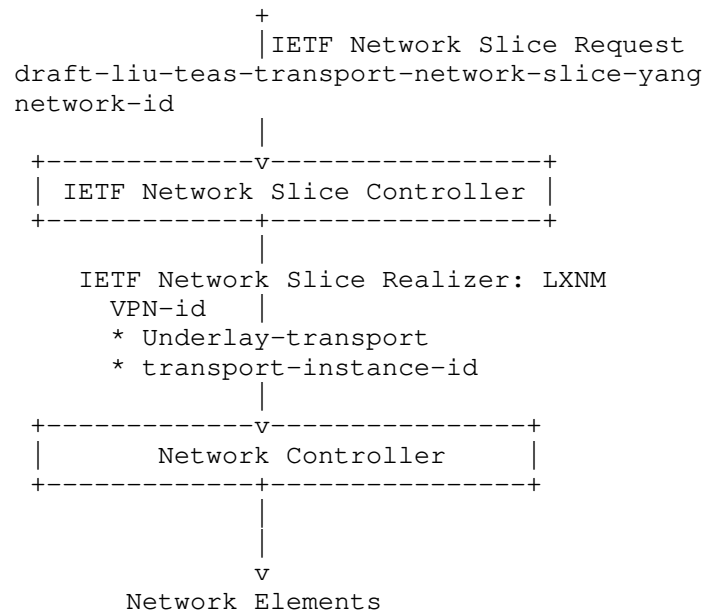


Figure 10 Workflow for the slice request in an stand-alone architecture.

### 7.3. Network Slice Controller as part of the domain controller

The Network Slice Controller can be a module of the Network controller. In that case, two options are available. One is to share the same device data model in the NBI and SBI of the SDN controller. The direct translation would reduce the service logic implemented at the SDN controller level, grouping the mapping and translation into a single task:

- \* **Realize:** As the device models are part of the network controller's NBI thus, the realization can be done by the network controller applying a simple service logic to send the Network elements.

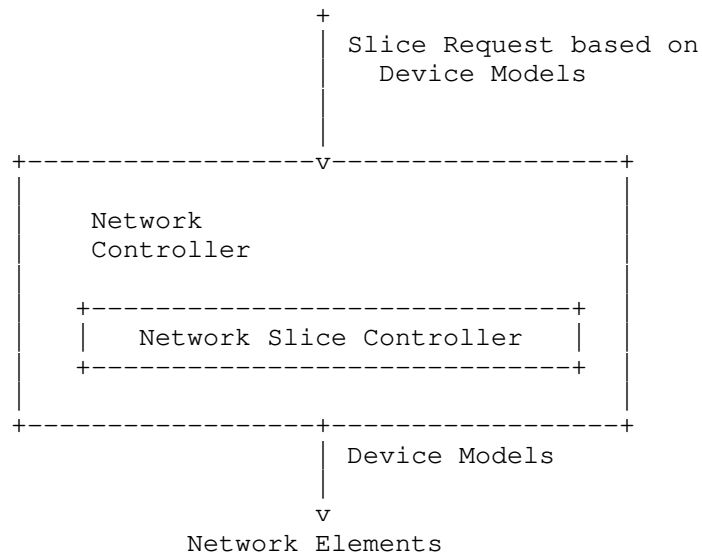


Figure 11 Workflow for the slice request in an stand-alone architecture.

A second option introduces a more complex logic in the network controller and creates an abstraction layer to process the transport slices. In that case, the controller should receive network slices creation requests and maintain the whole set of implemented slices:

- \* Map & Realize: The mapping and realization can be done by the Domain controller applying the service logic to create policies directly on the Network elements.

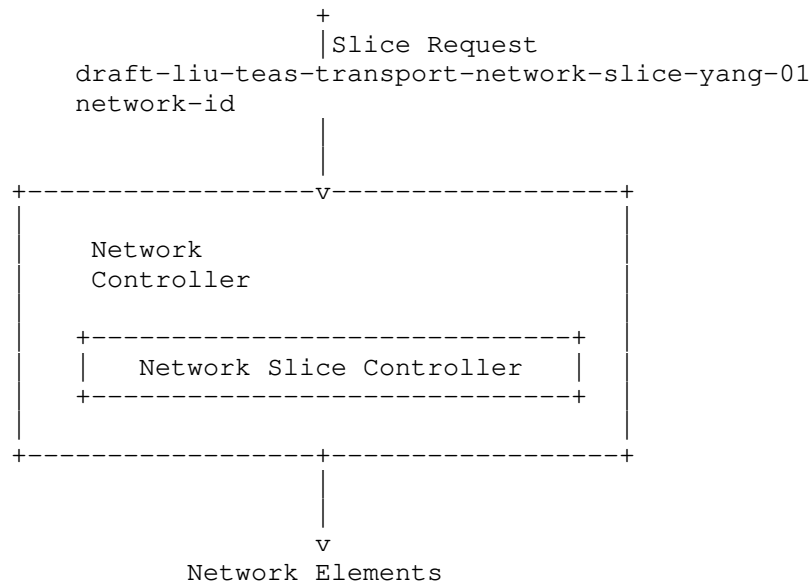


Figure 12 Workflow for the slice request in an stand-alone architecture.

## 8. Security Considerations

There are two main aspects to consider. On the one hand, the IETF Network Slice has a set of security related requirements, such as hard isolation of the slice, or encryption of the communications through the slice. All those requirements need to be analyzed in detailed and clearly mapped to the Network Controller and device interfaces.

On the other hand, the communication between the IETF network slicer and the network controller (or controllers or hierarchy of controllers) need to follow the same security considerations as with the network models.

The network YANG modules defines schemas for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040].

The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242].

The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8466].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

The following summarizes the foreseen risks of using the Network Models to instantiate IETF network Slices:

- \* Malicious clients attempting to delete or modify VPN services that implements an IETF network slice. The malicious client could manipulate security related aspects of the network configuration that impact the requirements of the slice, failing to satisfy the customer requirement.
- \* Unauthorized clients attempting to create/modify/delete a VPN hat implements an IETF network slice service.
- \* Unauthorized clients attempting to read VPN services related information hat implements an IETF network slice
- \* Malicious clients attempting to leak traffic of the slice.

## 9. IANA Considerations

This document is informational and does not require IANA allocations.

## 10. Conclusions

A wide variety of yang models are currently under definition in IETF that can be used by Network Controllers to instantiate IETF network slices. Some of the IETF slice requirements can be satisfied by multiple means, as there are multiple choices available. However, other requirements are still not covered by the existing models. A more detailed definition of those uncovered requirements would be needed. Finally a consensus on the set of models to be exposed by Network Controllers would facilitate the deployment of IETF network slices.

## 11. Contributors

Daniel King:daniel@olddog.co.uk>

Figure 1

## 12. Acknowledgements

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).

## 13. Normative References

[I-D.contreras-teas-slice-nbi]

Contreras, L. M., Homma, S., and J. A. Ordonez-Lucena, "IETF Network Slice Use Cases and Attributes for Northbound Interface of IETF Network Slice Controllers", Work in Progress, Internet-Draft, draft-contreras-teas-slice-nbi-04, 22 February 2021, <<https://datatracker.ietf.org/doc/html/draft-contreras-teas-slice-nbi-04>>.

[I-D.ietf-opsawg-l2nm]

Barguil, S., Dios, O. G. D., Boucadair, M., and L. A. Munoz, "A Layer 2 VPN Network YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-l2nm-02, 30 April 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-l2nm-02>>.

[I-D.ietf-opsawg-l3sm-l3nm]

Barguil, S., Dios, O. G. D., Boucadair, M., Munoz, L. A., and A. Aguado, "A Layer 3 VPN Network YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-l3sm-l3nm-08, 22 April 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-l3sm-l3nm-08>>.

[I-D.ietf-teas-ietf-network-slice-definition]

Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Definition of IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slice-definition-01, 22 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-ietf-network-slice-definition-01>>.

[I-D.ietf-teas-te-service-mapping-yang]

Lee, Y., Dhody, D., Fioccola, G., Wu, Q., Ceccarelli, D., and J. Tantsura, "Traffic Engineering (TE) and Service Mapping Yang Model", Work in Progress, Internet-Draft, draft-ietf-teas-te-service-mapping-yang-07, 21 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-te-service-mapping-yang-07>>.

- [I-D.ietf-teas-yang-te]  
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I.,  
and O. G. D. Dios, "A YANG Data Model for Traffic  
Engineering Tunnels, Label Switched Paths and Interfaces",  
Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-  
26, 22 February 2021,  
<[https://datatracker.ietf.org/doc/html/draft-ietf-teas-  
yang-te-26](https://datatracker.ietf.org/doc/html/draft-ietf-teas-yang-te-26)>.
- [I-D.nsd-t-teas-ns-framework]  
Gray, E. and J. Drake, "Framework for IETF Network  
Slices", Work in Progress, Internet-Draft, draft-nsdt-  
teas-ns-framework-05, 2 February 2021,  
<[https://datatracker.ietf.org/doc/html/draft-nsdt-teas-ns-  
framework-05](https://datatracker.ietf.org/doc/html/draft-nsdt-teas-ns-framework-05)>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,  
and A. Bierman, Ed., "Network Configuration Protocol  
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,  
<<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure  
Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,  
<<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF  
Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,  
<<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration  
Access Control Model", STD 91, RFC 8341,  
DOI 10.17487/RFC8341, March 2018,  
<<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for  
Abstraction and Control of TE Networks (ACTN)", RFC 8453,  
DOI 10.17487/RFC8453, August 2018,  
<<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG  
Data Model for Layer 2 Virtual Private Network (L2VPN)  
Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October  
2018, <<https://www.rfc-editor.org/info/rfc8466>>.



[RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and  
L. Geng, "A Framework for Automating Service and Network  
Management with YANG", RFC 8969, DOI 10.17487/RFC8969,  
January 2021, <<https://www.rfc-editor.org/info/rfc8969>>.

Authors' Addresses

Samier Barguil  
Telefonica  
Distrito T  
28050 Madrid  
Spain

Email: [samier.barguilgiraldo.ext@telefonica.com](mailto:samier.barguilgiraldo.ext@telefonica.com)

Luis Miguel Contreras  
Telefonica  
Distrito T  
28050 Madrid  
Spain

Email: [luismiguel.contrerasmurillo@telefonica.com](mailto:luismiguel.contrerasmurillo@telefonica.com)

Victor Lopez  
Nokia  
Calle de María Tubau, 9  
28050 Madrid  
Spain

Email: [victor.lopez@nokia.com](mailto:victor.lopez@nokia.com)

Reza Rokui  
Nokia  
Canada

Email: [reza.rokui@nokia.com](mailto:reza.rokui@nokia.com)

Oscar Gonzalez de Dios  
Telefonica  
Distrito T  
28050 Madrid  
Spain

Email: [oscar.gonzalezdedios@telefonica.com](mailto:oscar.gonzalezdedios@telefonica.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 8 September 2022

S. Barguil  
L.M. Contreras  
Telefonica  
V. Lopez  
Nokia  
R. Rokui  
Ciena  
O. Gonzalez de Dios  
Telefonica  
7 March 2022

Instantiation of IETF Network Slices in Service Providers Networks  
draft-barguil-teas-network-slices-instantiation-03

Abstract

Network Slicing (NS) is an integral part of Service Provider networks. The IETF has produced several YANG data models to support the Software-Defined Networking and network slice architecture and YANG-based service models for network slice (NS) instantiation.

This document describes the relationship between IETF Network Slice models for requesting the IETF Network Slices and (e.g., Layer-3 Service Model, Layer-2 Service Model) and Network Models (e.g., Layer-3 Network Model, Layer-2 Network Model) used during their realizations. In addition, this document describes the communication between the IETF Network Slice Controller and the network controllers for the realization of IETF network slices.

The IETF Network Slice YANG model provides the customer-oriented view of the network slice. Thus, once the IETF Network Slice controller (NSC) receives a request, it needs to map it to accomplish the specific parameters expected by the network controllers. The network models are analyzed to satisfy the IETF Network Slice requirements, and the gaps in existing models are reported.

The document also provides operational and security considerations when deploying network slices in Service Provider networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 September 2022.

#### Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

#### Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .   | 3  |
| 1.1. Terminology . . . . .  | 3  |
| 2. Reference Architecture and Components . . . . .  | 4  |
| 2.1. Possible architectural options for IETF Network Slice<br>Controller . . . . .            | 4  |
| 2.2. Possible relationship of IETF Network Slice service model<br>with other models . . . . . | 7  |
| 3. IETF Network Slice Requirements and Data Models . . . . .                                  | 8  |
| 4. IETF Network Slice Procedure . . . . .   | 9  |
| 5. Network Controller Operation . . . . .   | 10 |
| 5.1. LxVPN Service Models . . . . .   | 10 |
| 5.2. LxVPN Network Models . . . . .   | 11 |
| 5.3. Traffic Engineering Models . . . . .   | 11 |
| 5.4. Traffic Engineering Service Mapping . . . . .  | 11 |
| 6. Operational Considerations . . . . .   | 11 |
| 6.1. Availability . . . . .   | 12 |
| 6.2. Downlink throughput / Uplink throughput. . . . .   | 12 |
| 6.3. Protection scheme . . . . .  | 12 |
| 6.4. Delay . . . . .  | 13 |
| 6.5. Packet loss rate . . . . .   | 13 |

|   |    |
|---|----|
| 7. Network Slice Procedure . . . . .  | 13 |
| 7.1. IETF Network Slice requested to Hierarchical Network<br>Controller . . . . .                       | 14 |
| 7.2. IETF Network Slice requested to Network Slice<br>Controller . . . . .                              | 16 |
| 7.3. Network Slice Controller as part of the domain<br>controller . . . . .                             | 17 |
| 8. Security Considerations . . . . .  | 18 |
| 9. IANA Considerations . . . . .  | 19 |
| 10. Conclusions . . . . .   | 19 |
| 11. Contributors . . . . .  | 19 |
| 12. Acknowledgements . . . . .  | 20 |
| 13. Normative References . . . . .  | 20 |
| Annex. Example of relationship between IETF NBI model parameters<br>and L3SM model parameters . . . . . | 22 |
| Authors' Addresses . . . . .  | 25 |

## 1. Introduction

The IETF has produced several YANG data models to support the Software-Defined Networking and network slice architecture.

The IETF Network Slice YANG service model provides the customer-oriented view of the network slice. Once the IETF Network Slice controller (NSC) receives a request, it needs to map it to accomplish the specific parameters expected by the network controller.

Several Service Models and Network Models, including Layer-3 Service Model (L3SM), Layer-2 Service Model (L2SM) and Network Models which may be utilized for IETF Network Slicing, are analyzed can satisfy the IETF Network Slice requirements. In addition, identified gaps on existing models are reported.

This document describes the architecture and communication process between the Network Slice Controller and a network controller for IETF network slice creation.

Editor's Note: the terminology in this draft will be aligned with the final terminology selected for describing the notion of IETF Network Slice when applied to IETF technologies, as being defined in [I-D.ietf-teas-ietf-network-slices].

### 1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

## 2. Reference Architecture and Components

As described in [I-D.ietf-teas-ietf-network-slices], the IETF Network Slice Controller (NSC) is a functional entity for control and management of IETF network slices. As shown in Figure A, NSC from its Northbound Interface (NBI) exposes set of APIs that allow a higher level system to request an IETF network slice. The NSC NBI supports the request for enabling of an IETF Network Slice (i.e., creation, modification or deletion). Upon receiving a request from its NBI, NSC finds the resources needed for realization of the IETF Network Slice and in turn interfaces from its Southbound Interface (SBI) with one or more Network Controllers for the realization of the requested IETF Network Slice.

This document focuses on how IETF Network Slice Controller (NSC) can be implemented in the operator's network.

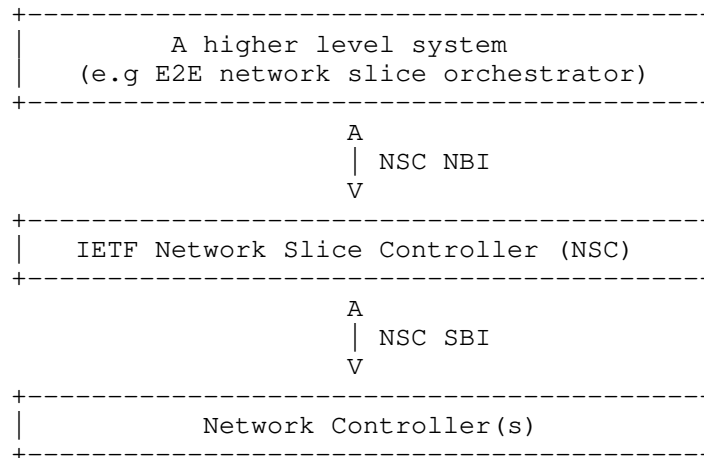


Figure 1 Network Slice Controller as a module of the Hierarchical SDN controller.

### 2.1. Possible architectural options for IETF Network Slice Controller

Several architectural definitions have arisen on the IETF to support SDN and network slicing deployments. The architectural proposal defined in [I-D.ietf-teas-ietf-network-slices] includes a three-level hierarchy and expresses how each level relates with the ACTN architecture framework.

Figure 2 defines depicts a possible architecture using those concepts. It starts from a top consumer or high-level operational systems. Next, the IETF Network Slice Controller function might be

part of the Hierarchical network controller (e.g., as the MDSC in the ACTN context [RFC8453]) as a modular function. At the bottom, two network controllers, each one can handle multiple or single underlay technologies.

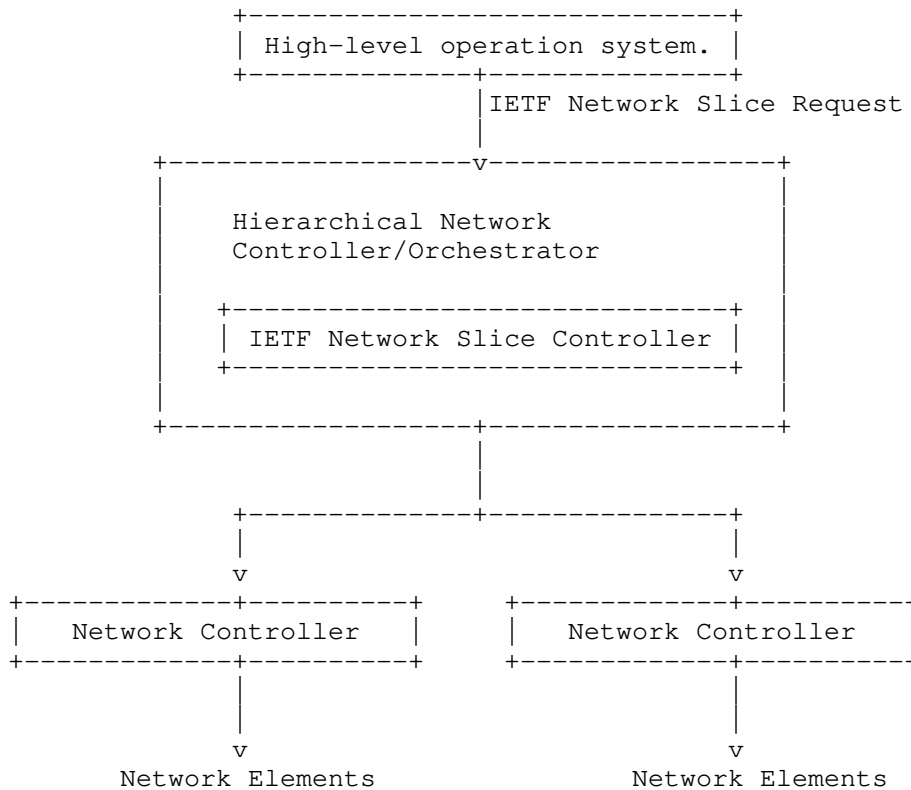


Figure 2 IETF Network Slice Controller as a module of the Hierarchical SDN controller.

In other implementations, the IETF Network Slice Controller can be a stand-alone element and directly interact with the network controller, as depicted in Figure 2. In this scenario, the services request follows a data-enrichment path, where each entity adds more information to the service request. This document describes how the available service models and network models interact to deliver the network slices in a service provider environment.

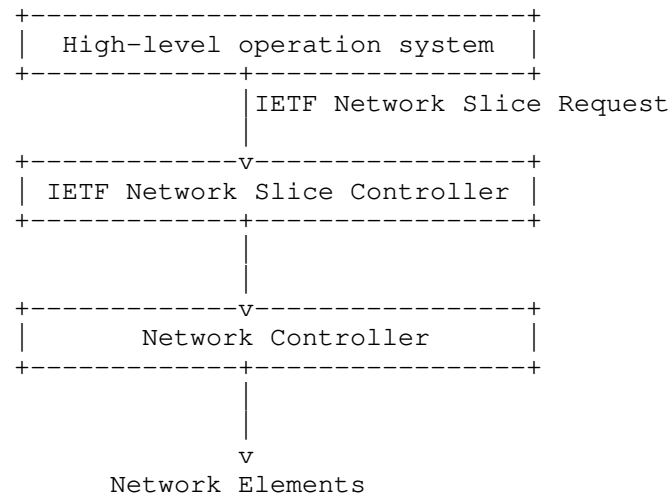


Figure 3 The IETF Network Slice Controller as a stand-alone entity.

As another implementation possibility, the IETF Network Slice Controller can be integrated with the Network controller and directly realize the network slice using device data models to configure the network devices. The sample architecture is depicted in Figure 4.

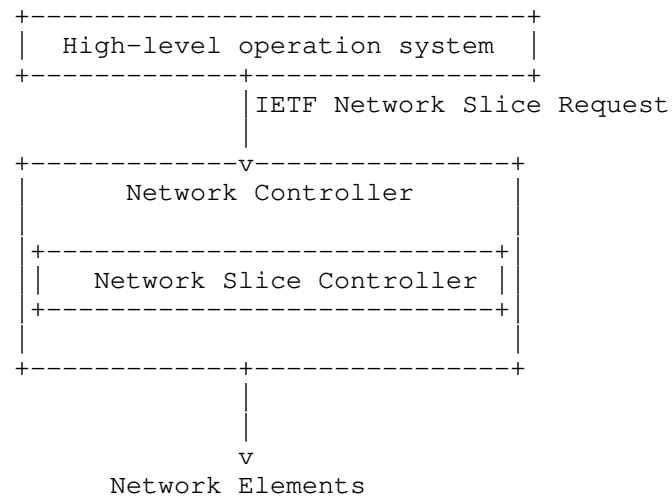


Figure 4 IETF Network Slice Controller as a module of the Network controller.

## 2.2. Possible relationship of IETF Network Slice service model with other models

IETF Network Slice service is expected to serve as input from where deriving some other models in the network. According to the architectural options before, different relationships could be considered. Figure 5 reflects a couple of options.

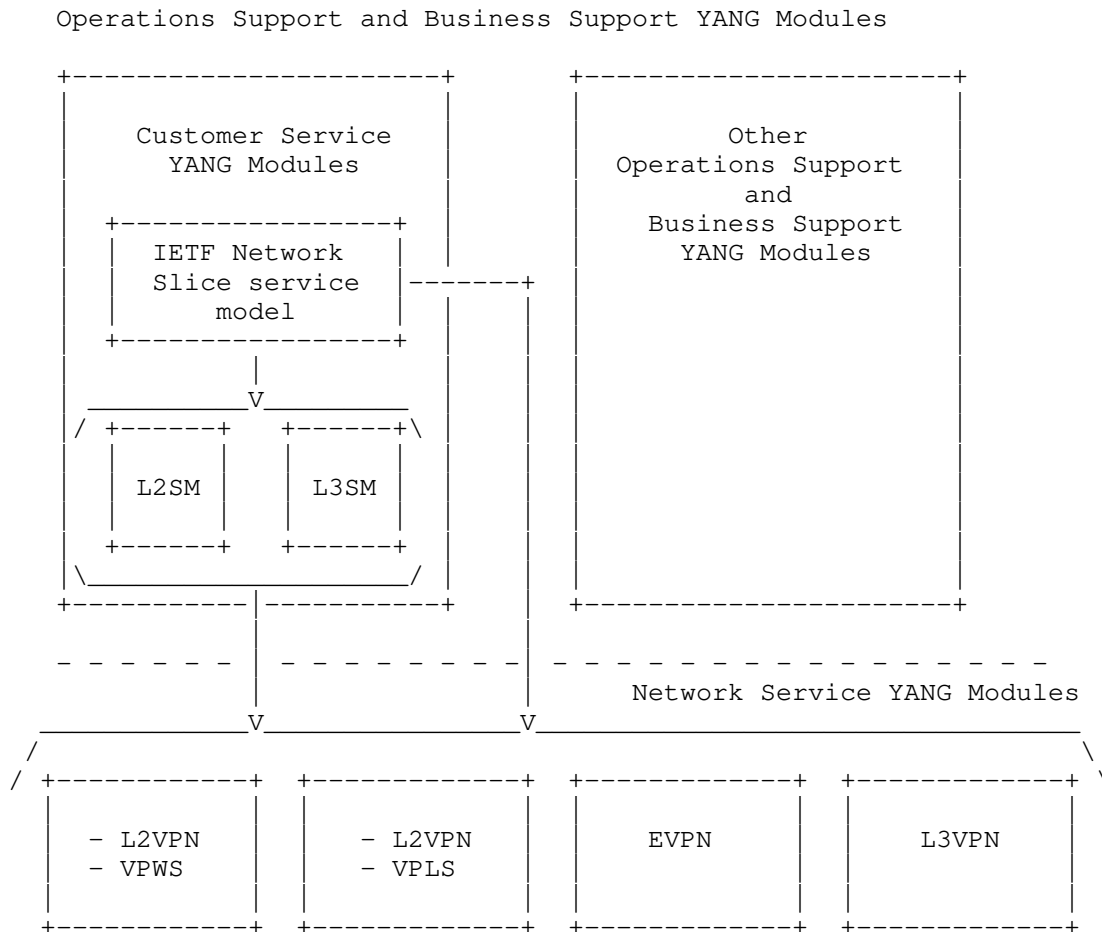


Figure 5 Possible relationships between models.

Thus, the IETF Network Slice model (e.g., as defined in [RefNBIdraft]) could feed existing service models, such as L2SM or L3SM, or could feed existing network models (e.g., EVPN, L3VPN, etc). Existing models both for service or network level could require some



extensions themselves, or their application in conjunction with some other complementary models (e.g., TE model) to accomplish the service objectives and expectations as declared in the IETF Network Slice model.

### 3. IETF Network Slice Requirements and Data Models

The main set of requirements for the IETF Slice, based on the high-level slice requirements from multiple organizations and use cases, are compiled in [I-D.contreras-teas-slice-nbi] and reproduced bellow the slice use cases reported:

| Network Slice Requirements for 5G service   |
|---|
| Availability<br>Deterministic communication<br>Downlink throughput per network slice<br>Energy efficiency<br>Group communication support<br>Isolation level<br>Maximum supported packet size<br>Mission critical support<br>Performance monitoring<br>Slice quality of service parameters<br>Support for non-IP traffic<br>Uplink throughput per network slice<br>User data access<br>Delay tolerance |
| NFV-based services  |
| Incoming and outgoing bandwidth<br>Qos metrics<br>Directionality<br>MTU<br>Protection scheme<br>Connectivity mode   |

|                                 |
|---------------------------------|
| Network sharing                 |
| Maximum and Guaranteed Bit Rate |
| Bounded latency                 |
| Packet loss rate                |
| IP addressing                   |
| L2/L3 reachability              |
| Recovery time                   |
| Secure connection               |

To accomplish those requirements, a set of YANG data models have been proposed. Those Yang models, summarized in table xx, could be used by an IETF Network Slice Controller to manage CRUD operations on the IETF Network Slice. That is, these models aim capturing the requirements from the consumer of the slice point of view and avoid entering into the detail of how the slice is actually created.

- \* [draft-wd-teas-ietf-network-slice-nbi-yang]: A Yang Data Model for IETF Network Slice NBI.
- \* [draft-liu-teas-transport-network-slice-yang]: Transport Network Slice YANG Data Model.

#### 4. IETF Network Slice Procedure

An IETF Network Slice may use several underlying technologies. The creation of a new IETF Network Slice will be initiated with following three steps:

1. A higher level system requests connections with specific characteristics via the NBI.
2. This request will be processed by an IETF NSC which specifies a mapping between northbound request to any IETF Services, Tunnels, and paths models.
3. A series of requests for creation of services, tunnels and paths will be sent to the network to realize the transport slice.

## 5. Network Controller Operation

As a functional entity responsible for managing a network domain, the network controller, can expose its northbound interface based on YANG models. The IETF Network Slice Controller can use the network controller's NBI during the realization of IETF Network Slice. The following network models can be used for realization of IETF Network slices:

- \* LxVPN Network models:

- These models describe a VPN service from the network point of view. It supports the creation of Layer 3 and Layer 2 services using several control planes.

- \* Traffic Engineering models:

- These models allow to manipulate Traffic Engineering tunnels within the network segment. Technology-specific extensions allow to work with a desired technology (e.g. MPLS RSVP-TE tunnels, Segment Routing paths, OTN tunnels, etc.)

- \* TE Service Mapping extensions:

- These extensions allow to specify for LxVPN the details of an underlay based on TE.

- \* ACLs and routing policies models:

- Even though ACLs and routing policies are device models, it's exposure in the NBI of a domain controller allows to provide an additional granularity that the network domain controller is not able to infer on its own.

### 5.1. LxVPN Service Models

The framework defined in [RFC8969] compiles a set of YANG data models for automating network services. The data models can be used during the service and network management life cycle (e.g., service instantiation, service provisioning, service optimization, service monitoring, service diagnosing, and service assurance). The Service models could be a realization of IETF Network slice requests.

The following models are examples of Network models that describe services.

- \* [RFC8049]: YANG Data Model for L3VPN Service Delivery

- \* [RFC8466]: A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery

## 5.2. LxVPN Network Models

Similar to the Service Models, the framework defined in [RFC8969] compiles a set of YANG data models for automating network services. The Network models could be reused for the realization of Network slice requests.

The following models are examples of Network models that describe services.

- \* [I-D.ietf-opsawg-l3sm-l3nm]: A Layer 3 VPN Network YANG Model
- \* [I-D.ietf-opsawg-l2nm]: A Layer 2 VPN Network YANG Model

## 5.3. Traffic Engineering Models

TEAS has defined a collection of models to allow the management of Traffic Engineering tunnels.

- \* [I-D.ietf-teas-yang-te]: A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces. The model allows to instantiate paths in a TE enabled network. Note that technology augmented models are require to particular per-technology instantiations.

## 5.4. Traffic Engineering Service Mapping

The IETF has defined a YANG model to set up the procedure to map VPN service/network models to the TE models. This model, known as service mapping, allows the network controller to assign/retrieve transport resources allocated to specific services. At the moment there is just one service mapping model [I-D.ietf-teas-te-service-mapping-yang]. The "Traffic Engineering (TE) and Service Mapping Yang Model" augments the VPN service and network models.

## 6. Operational Considerations

This section outlines the compliance and operational aspects of Network Controller models with IETF Network slice requirements. Section presented the requirements of the IETF Network slice. In this subsection it is analyzed how available YANG models that can be used by a Network Controller can satisfy those requirements and identify gaps.

### 6.1. Availability

As per [draft-ietf-teas-te-service-mapping-yang], Availability is a probabilistic measure of the length of time that a VPN/VN instance functions without a network failure. As per RFC 8330, The parameter "availability", as described in [G.827], [F.1703], and [P.530], is often used to describe the link capacity. The availability is a time scale, representing a proportion of the operating time that the requested bandwidth is ensured".

The calculation of the availability is not trivial and would need to be clearly scoped to avoid misunderstandings.

The set of Yang models proposed today allow to request tunnels/paths with different resiliency requirements in terms of protection and restoration. However, none of them include the possibility of requesting a specific availability (e.g. 99.9999%).

### 6.2. Downlink throughput / Uplink throughput.

The LxVPN Models ([I-D.ietf-opsawg-l3sm-l3nm] and [I-D.ietf-opsawg-l2nm]) allow to specify the bandwidth at the interface level between the slice and the customer. In addition, the Service Mapping model [draft-ietf-teas-te-service-mapping-yang] allows to bind a VPN to a given LSP, which have its bandwidth requirements. Additionally, TE models can force a give bandwidth in the connection between Provider Edges.

Previous comment applies to the incoming and outgoing bandwidth parameters required for the NFV-based services use case in [I-D.contreras-teas-slice-nbi]. The Network sharing use case has Maximum and Guaranteed Bit Rate parameters. These parameters can be mapped to the TE tunnel models when setting up LSPs [draft-ietf-teas-yang-te].

### 6.3. Protection scheme

Protection schemes are mechanisms to define how to setup resources for a given connection. TE tunnel models [draft-ietf-teas-yang-te] includes protection and restoration as two main attributes. The parameters included in the containers for protection and restoration cover the requirements of the IETF NS related with protection schemes. Similarly, TE models cover the parameter 'recovery time' for the network sharing use case.

#### 6.4. Delay

Delay is a critical parameter for several IETF NS types. Every use-case defined in [I-D.contreras-teas-slice-nbi] contains delay constraints. 5G use cases require 'delay tolerance', NFV-based services have the delay information within 'QoS metrics' and 'Bounded latency' in the network sharing use case.

During the realization of the IETF Network Slice, these parameters are part of the requirements of a TE tunnel configuration [draft-ietf-teas-yang-te]. They can be included within the 'path-metric-bounds' parameter, so the created LSP fulfils the given metrics bounds like 'path-metric-delay-average' or 'path-metric-delay-minimum'.

#### 6.5. Packet loss rate

The packet loss rate indicates the maximum rate for lost packets that the service tolerates in the link. During the realization of the IETF Network Slice, this attribute will influence the tunnel selection and the value is included in the [draft-ietf-teas-yang-te] document as the 'path-metric-loss'. The 'path-metric-loss' is a metric type, which measures the percentage of packet loss of all links traversed by a P2P path. This parameter is required for 5G services and network sharing use-case, while it is part of the 'QoS metrics' for the NFV-based services.

#### 7. Network Slice Procedure

Draft [draft-contreras-teas-slice-controller-models] shows the internal structure of an IETF Network Slice Controller which can be divided into two components:

- \* IETF Network Slice Mapper: this high-level component processes the customer request, putting it into the context of the overall IETF Network Slices in the network.
- \* IETF Network Slice Realizer: this high-level component processes the complete view of transport slices including the one requested by the customer, decides the proper technologies for realizing the IETF Network Slice and triggers its realization.

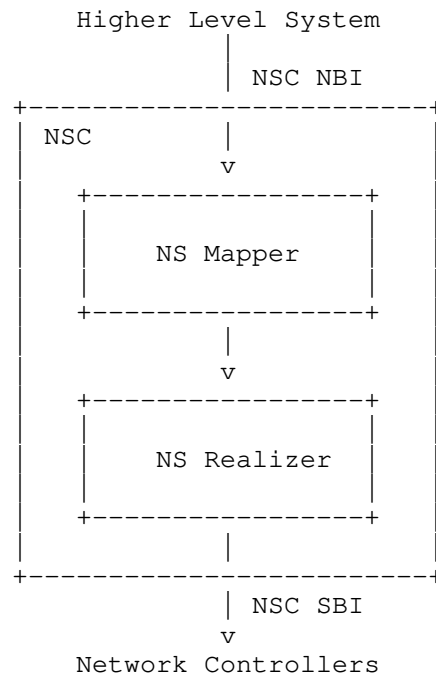


Figure 8: IETF Network Slice Controller Structure

The details of IETF network slice mapper and realize are provided below for various implementation of NCS.

#### 7.1. IETF Network Slice requested to Hierarchical Network Controller

Referring to Figure 1 in an integrated architecture, the IETF Network Slice Controller (NCS) is part of a Hierarchical SDN controller module, the NSC's and the Hierarchical Network Controller should share the same internal data and the same NBI. Thus, the H-SDN module must be able to:

- \* Map: The customer request received using the [draft-wd-teas-ietf-network-slice-nbi-yang] must be processed by the NCS. The mapping process takes the network-slice SLAs selected by the customer to available Routing Policies and Forwarding policies.

- \* **Realize:** Create necessary network requests. The slice's realization can be translated into one or several LXNM Network requests, depending on the number of underlay controllers. Thus, the NCS must have a complete view of the network to map the orders and distribute them across domains. The realization should include the expansion/selection of Forwarding Policies, Routing Policies, VPN policies, and Underlay transport preference.

To maintain the data coherence between the control layers, the IETF Network Slice ID ns-id used of the [draft-wd-teas-ietf-network-slice-nbi-yang] must be directly mapped to the transport-instance-id at the VPN-Node level.

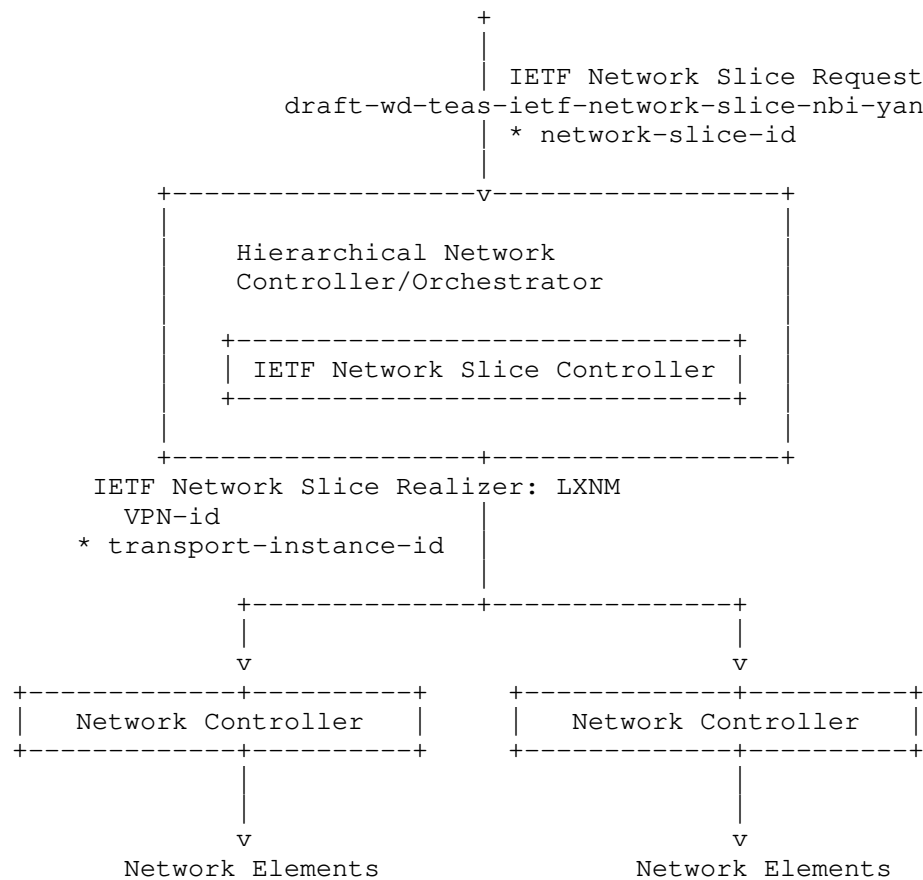


Figure 9 Workflow for the slice request in an integrated architecture.



## 7.2. IETF Network Slice requested to Network Slice Controller

Referring to Figure 2 when the Network Slice Controller is a stand-alone controller module, the NSC's should perform the same two tasks described in section 6.1:

- \* **Map:** Process the customer request. The customer request can be sent using the [draft-liu-teas-transport-network-slice-yang]. This draft allows the topology mapping of the Slice request.
- \* **Realize:** Create necessary network requests. The slice's realization will be translated into one LXNM Network request. As the NCS has a topological view of the network, the realization can include the customer's traffic engineering transport preferences and policies.

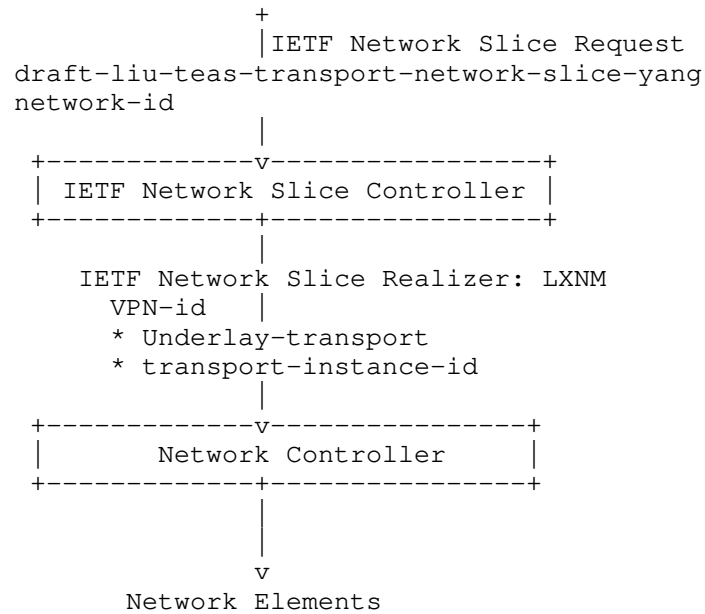


Figure 10 Workflow for the slice request in an stand-alone architecture.

### 7.3. Network Slice Controller as part of the domain controller

The Network Slice Controller can be a module of the Network controller. In that case, two options are available. One is to share the same device data model in the NBI and SBI of the SDN controller. The direct translation would reduce the service logic implemented at the SDN controller level, grouping the mapping and translation into a single task:

- \* **Realize:** As the device models are part of the network controller's NBI thus, the realization can be done by the network controller applying a simple service logic to send the Network elements.

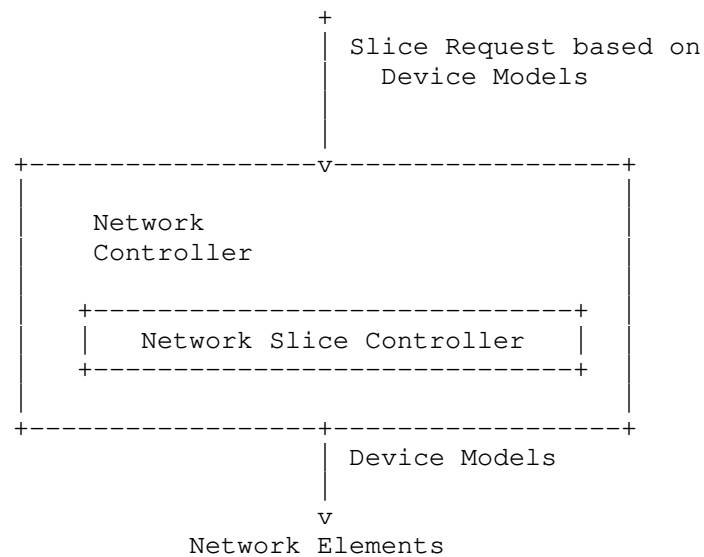


Figure 11 Workflow for the slice request in an stand-alone architecture.

A second option introduces a more complex logic in the network controller and creates an abstraction layer to process the transport slices. In that case, the controller should receive network slices creation requests and maintain the whole set of implemented slices:

- \* **Map & Realize:** The mapping and realization can be done by the Domain controller applying the service logic to create policies directly on the Network elements.

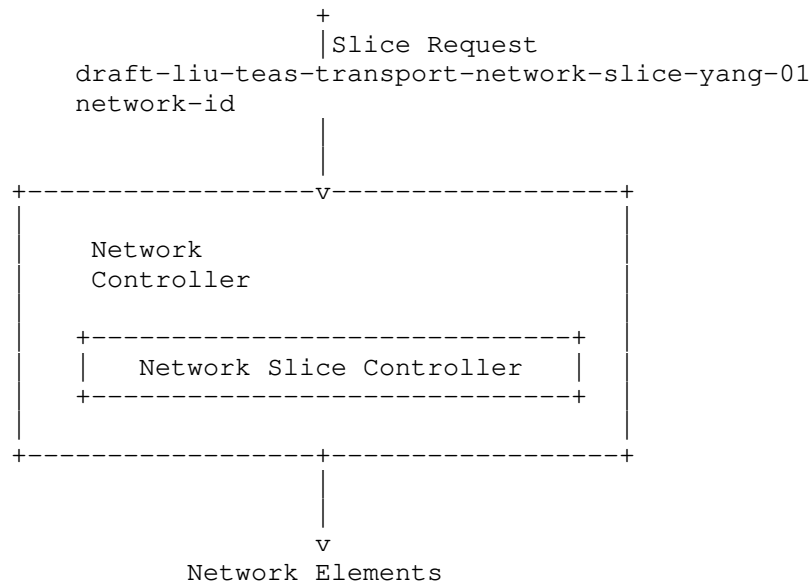


Figure 12 Workflow for the slice request in an stand-alone architecture.

## 8. Security Considerations

There are two main aspects to consider. On the one hand, the IETF Network Slice has a set of security related requirements, such as hard isolation of the slice, or encryption of the communications through the slice. All those requirements need to be analyzed in detailed and clearly mapped to the Network Controller and device interfaces.

On the other hand, the communication between the IETF network slicer and the network controller (or controllers or hierarchy of controllers) need to follow the same security considerations as with the network models.

The network YANG modules defines schemas for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040].

The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242].

The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8466].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

The following summarizes the foreseen risks of using the Network Models to instantiate IETF network Slices:

- \* Malicious clients attempting to delete or modify VPN services that implements an IETF network slice. The malicious client could manipulate security related aspects of the network configuration that impact the requirements of the slice, failing to satisfy the customer requirement.
- \* Unauthorized clients attempting to create/modify/delete a VPN hat implements an IETF network slice service.
- \* Unauthorized clients attempting to read VPN services related information hat implements an IETF network slice
- \* Malicious clients attempting to leak traffic of the slice.

## 9. IANA Considerations

This document is informational and does not require IANA allocations.

## 10. Conclusions

A wide variety of yang models are currently under definition in IETF that can be used by Network Controllers to instantiate IETF network slices. Some of the IETF slice requirements can be satisfied by multiple means, as there are multiple choices available. However, other requirements are still not covered by the existing models. A more detailed definition of those uncovered requirements would be needed. Finally, a consensus on the set of models to be exposed by Network Controllers would facilitate the deployment of IETF network slices.

## 11. Contributors

Daniel King:daniel@olddog.co.uk>

Figure 1

## 12. Acknowledgements

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).

## 13. Normative References

[I-D.contreras-teas-slice-nbi]

Contreras, L. M., Homma, S., Ordonez-Lucena, J. A., Tantsura, J., and K. Szarkowicz, "IETF Network Slice Use Cases and Attributes for Northbound Interface of IETF Network Slice Controllers", Work in Progress, Internet-Draft, draft-contreras-teas-slice-nbi-05, 12 July 2021, <<https://datatracker.ietf.org/doc/html/draft-contreras-teas-slice-nbi-05>>.

[I-D.ietf-opsawg-l2nm]

Barguil, S., Dios, O. G. D., Boucadair, M., and L. A. Munoz, "A Layer 2 VPN Network YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-l2nm-12, 22 November 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-l2nm-12>>.

[I-D.ietf-opsawg-l3sm-l3nm]

Barguil, S., Dios, O. G. D., Boucadair, M., Munoz, L. A., and A. Aguado, "A YANG Network Data Model for Layer 3 VPNs", Work in Progress, Internet-Draft, draft-ietf-opsawg-l3sm-l3nm-18, 8 October 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-l3sm-l3nm-18>>.

[I-D.ietf-teas-ietf-network-slices]

Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-08, 6 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-ietf-network-slices-08>>.

[I-D.ietf-teas-te-service-mapping-yang]

Lee, Y., Dhody, D., Fioccola, G., Wu, Q., Ceccarelli, D., and J. Tantsura, "Traffic Engineering (TE) and Service Mapping YANG Model", Work in Progress, Internet-Draft, draft-ietf-teas-te-service-mapping-yang-09, 24 October 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-te-service-mapping-yang-09>>.

- [I-D.ietf-teas-yang-te]  
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I.,  
and O. G. D. Dios, "A YANG Data Model for Traffic  
Engineering Tunnels, Label Switched Paths and Interfaces",  
Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-  
29, 7 February 2022,  
<[https://datatracker.ietf.org/doc/html/draft-ietf-teas-  
yang-te-29](https://datatracker.ietf.org/doc/html/draft-ietf-teas-yang-te-29)>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,  
and A. Bierman, Ed., "Network Configuration Protocol  
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,  
<<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure  
Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,  
<<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF  
Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,  
<<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration  
Access Control Model", STD 91, RFC 8341,  
DOI 10.17487/RFC8341, March 2018,  
<<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for  
Abstraction and Control of TE Networks (ACTN)", RFC 8453,  
DOI 10.17487/RFC8453, August 2018,  
<<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG  
Data Model for Layer 2 Virtual Private Network (L2VPN)  
Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October  
2018, <<https://www.rfc-editor.org/info/rfc8466>>.
- [RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and  
L. Geng, "A Framework for Automating Service and Network  
Management with YANG", RFC 8969, DOI 10.17487/RFC8969,  
January 2021, <<https://www.rfc-editor.org/info/rfc8969>>.

#### Annex. Example of relationship between IETF NBI model parameters and L3SM model parameters

This annex presents an initial analysis of the relationship between IETF NBI model parameters and L3SM service model parameters.

The L3SM service parameters are defined in section 6.2 of RFC 8299. The following parameters are considered, so far:

- \* Bandwidth. This parameter indicates the bandwidth requirement between each CE and PE participating in the service, then referring essentially to the required WAN link bandwidth. It is expressed in terms of bits per second and individually specified for both input and output. Despite it is not stated in RFC 8299, this parameter can be interpreted as the CIR/PIR expected for the CE - PE connection.
- \* MTU. This parameter indicates the maximum PDU size expected for the layer-3 service. It is relevant since packets could be discarded in case the customer sends packets with longer MTU than the one expressed by this parameter.
- \* QoS. Regarding QoS, two different kind of parameters are detailed.
  - QoS classification policy. This policy is used to classify the traffic received from the customer, and it is expressed as a set of ordered rules. It is used for marking the input traffic (from CE to PE) when the customer flows match any of the rules in the list, setting the appropriate target class of service (target-class-id).
  - QoS profile. This profile defines the traffic-scheduling to be applied to the flows for either Site-to-WAN, WAN-to-Site, or both directions. It contains the following information per class of service: rate-limit, latency, jitter and guaranteed bandwidth.
- \* Multicast. This parameter identifies if the service is multicast, and if so, what is the role of the site in the customer multicast service topology (i.e., source, receiver, or both). It also defines the kind of multicast relationship with the customer (i.e., as a router requiring PIM, host requiring either IGMP or MLD, or both), as well as the support of IPv4, IPv6 or both.

On the other hand, the IETF NS NBI YANG model supports a number of SLOs and SLEs in the form of network slice service policy attributes. Such policy can apply to per-network slice, per-connection group or

per-connection individually (over-writing of attributes is allowed as more granular information is provided). The following SLO attributes are detailed:

- \* One-way / Two-way bandwidth, indicating the guaranteed minimum bandwidth between any two NSEs (unidirectional / bidirectional).
- \* One-way / Two-way latency, indicating the guaranteed minimum latency between any two NSEs (unidirectional / bidirectional).
- \* One-way / Two-way delay variation, indicating the maximum permissible delay variation of the slice (unidirectional / bidirectional).
- \* One-way / Two-way packet loss, indicating the maximum permissible packet loss rate between endpoints (unidirectional / bidirectional).

Additionally, the following SLEs are defined:

- \* MTU, referring to the the maximum PDU size that the customer may use.
- \* Security, indicating if encryption or other security measures are required between two endpoints.
- \* Isolation, as a way of indicating the isolation level expected by the customer in the allocation of network resources.
- \* Maximum occupancy level, to express the amount of flows to be admitted (and optionally a maximum number of countable resource units such as IP or MAC addresses).

Thus, an initial mapping between L3SM and IETF NS NBI model can be performed as indicated in the following table.



| +                           |   |
|-----------------------------|---|
| L3SM (RFC 8299)             | IETF NSC NBI YANG model   |
| Bandwidth                   | Sum of bandwidth SLO per NSE counting all connections   |
| MTU                         | MTU attribute in SLE  |
| QoS                         |   |
| .....                       | .....   |
| - QoS classification policy | Defined in the model as network-access-qos-policy-name to be applied per access-point   |
| .....                       | .....   |
| - QoS profile               |   |
| - rate-limit                | Defined in the model as incoming/outgoing rate-limits per end-point (or access-point)   |
| - latency                   | One-way / Two-way latency SLO   |
| - jitter                    | One-way / Two-way delay variation SLO   |
| - bandwidth                 | One-way / Two-way bandwidth SLO   |
| Multicast                   | The need of replication can be inferred from ns-connectivity-type. Further details are not available (e.g. source or receiver role) |

Table 1 Mapping of IETF NS NBI and L3SM service attributes.

The following consideration can be made.

- \* While the QoS profile in L3SM applies per service class, the parameters in IETF NS NBI apply per connection. So if per-class granularity is required in an IETF network slice, then different connections have to be defined between the same end-points, one per service class.
- \* A number of attributes are not defined in L3SM such as packet loss, isolation or security. Then L3SM could not be sufficient to realize IETF network slices with such specific needs, unless those other objectives and expectations are provided by other means (e.g., realizing the L3SM thorough technologies guaranteeing dedicated resource allocation such as OTN).

Authors' Addresses

Samier Barguil  
Telefonica  
Distrito T  
28050 Madrid  
Spain  
Email: samier.barguilgiraldo.ext@telefonica.com

Luis M. Contreras  
Telefonica  
Distrito T  
28050 Madrid  
Spain  
Email: luismiguel.contrerasmurillo@telefonica.com

Victor Lopez  
Nokia  
Calle de María Tubau, 9  
28050 Madrid  
Spain  
Email: victor.lopez@nokia.com

Reza Rokui  
Ciena  
Canada  
Email: rrokui@ciena.com

Oscar Gonzalez de Dios  
Telefonica  
Distrito T  
28050 Madrid  
Spain  
Email: oscar.gonzalezdedios@telefonica.com

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 12, 2022

T. Saad  
V. Beeram  
Juniper Networks  
B. Wen  
Comcast  
D. Ceccarelli  
J. Halpern  
Ericsson  
S. Peng  
R. Chen  
ZTE Corporation  
X. Liu  
Volta Networks  
L. Contreras  
Telefonica  
R. Rokui  
Nokia  
July 11, 2021

Realizing Network Slices in IP/MPLS Networks  
draft-bestbar-teas-ns-packet-03

Abstract

Network slicing provides the ability to partition a physical network into multiple logical networks of varying sizes, structures, and functions so that each slice can be dedicated to specific services or customers. Network slices need to operate in parallel while providing slice elasticity in terms of network resource allocation. The Differentiated Service (Diffserv) model allows for carrying multiple services on top of a single physical network by relying on compliant nodes to apply specific forwarding treatment (scheduling and drop policy) on to packets that carry the respective Diffserv code point. This document proposes a solution based on the Diffserv model to realize network slicing in IP/MPLS networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 12, 2022.

#### Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                               | 3  |
| 1.1. Terminology . . . . .                              | 4  |
| 1.2. Acronyms and Abbreviations . . . . .               | 6  |
| 2. Network Resource Slicing Membership . . . . .        | 6  |
| 2.1. Dedicated Network Resources . . . . .              | 6  |
| 2.2. Shared Network Resources . . . . .                 | 7  |
| 3. Path Selection . . . . .                             | 7  |
| 4. Slice Policy Modes . . . . .                         | 8  |
| 4.1. Data plane Slice Policy Mode . . . . .             | 8  |
| 4.2. Control Plane Slice Policy Mode . . . . .          | 9  |
| 4.3. Data and Control Plane Slice Policy Mode . . . . . | 11 |
| 5. Slice Policy Instantiation . . . . .                 | 12 |
| 5.1. Slice Policy Definition . . . . .                  | 13 |
| 5.1.1. Slice Policy Data Plane Selector . . . . .       | 14 |
| 5.1.2. Slice Policy Resource Reservation . . . . .      | 17 |
| 5.1.3. Slice Policy Per Hop Behavior . . . . .          | 18 |
| 5.1.4. Slice Policy Topology . . . . .                  | 19 |
| 5.2. Slice Policy Boundary . . . . .                    | 19 |
| 5.2.1. Slice Policy Edge Nodes . . . . .                | 19 |
| 5.2.2. Slice Policy Interior Nodes . . . . .            | 20 |
| 5.2.3. Slice Policy Incapable Nodes . . . . .           | 20 |
| 5.2.4. Combining Slice Policy Modes . . . . .           | 21 |
| 5.3. Mapping Traffic on Slice Aggregates . . . . .      | 22 |
| 6. Control Plane Extensions . . . . .                   | 22 |

|   |    |
|---|----|
| 7. Applicability to Path Control Technologies . . . . . | 23 |
| 8. IANA Considerations . . . . .                        | 23 |
| 9. Security Considerations . . . . .                    | 23 |
| 10. Acknowledgement . . . . .                           | 24 |
| 11. Contributors . . . . .                              | 24 |
| 12. References . . . . .                                | 24 |
| 12.1. Normative References . . . . .                    | 24 |
| 12.2. Informative References . . . . .                  | 26 |
| Authors' Addresses . . . . .                            | 27 |

## 1. Introduction

Network slicing allows a Service Provider to create independent and logical networks on top of a common or shared physical network infrastructure. Such network slices can be offered to customers or used internally by the Service Provider to facilitate or enhance their service offerings. A Service Provider can also use network slicing to structure and organize the elements of its infrastructure. This document provides a path control technology agnostic solution that a Service Provider can deploy to realize network slicing in IP/MPLS networks.

The definition of network slice for use within the IETF and the characteristics of IETF network slice are specified in [I-D.ietf-teas-ietf-network-slice-definition]. A framework for reusing IETF VPN and traffic-engineering technologies to realize IETF network slices is discussed in [I-D.nsd-t-teas-ns-framework]. These documents also discuss the function of an IETF Network Slice Controller and the requirements on its northbound and southbound interfaces.

This document introduces the notion of a slice aggregate which comprises of one or more IETF network slice traffic streams. It describes how a slice policy can be used to realize a slice aggregate by instantiating specific control and data plane behaviors on select topological elements in IP/MPLS networks. The onus is on the IETF Network Slice Controller to maintain the mapping between one or more IETF network slices and a slice aggregate. The mechanisms used by the controller to determine the mapping are outside the scope of this document. The focus of this document is on the mechanisms required at the device level to address the requirements of network slicing in packet networks.

In a Differentiated Service (Diffserv) domain [RFC2475], packets requiring the same forwarding treatment (scheduling and drop policy) are classified and marked with a Class Selector (CS) at domain ingress nodes. At transit nodes, the CS field inside the packet is inspected to determine the specific forwarding treatment to be

applied before the packet is forwarded further. Similar principles are adopted by this document to realize network slicing.

When logical networks representing slice aggregates are realized on top of a shared physical network infrastructure, it is important to steer traffic on the specific network resources allocated for the slice aggregate. In packet networks, the packets that traverse a specific slice aggregate MAY be identified by one or more specific fields carried within the packet. A slice policy ingress boundary node populates the respective field(s) in packets that enter a slice aggregate to allow interior slice policy nodes to identify those packets and apply the specific Per Hop Behavior (PHB) that is associated with the slice aggregate. The PHB defines the scheduling treatment and, in some cases, the packet drop probability.

The slice aggregate traffic may further carry a Diffserv CS to allow differentiation of forwarding treatments for packets within a slice aggregate. For example, when using MPLS as a dataplane, it is possible to identify packets belonging to the same slice aggregate by carrying a global MPLS label in the label stack that identifies the slice aggregate in each packet. Additional Diffserv classification may be indicated in the Traffic Class (TC) bits of the global MPLS label to allow further differentiation of forwarding treatments for traffic traversing the same slice aggregate network resources.

This document covers different modes of slice policy and discusses how each slice policy mode can ensure proper placement of slice aggregate paths and respective treatment of slice aggregate traffic.

### 1.1. Terminology

The reader is expected to be familiar with the terminology specified in [I-D.ietf-teas-ietf-network-slice-definition] and [I-D.nsd-t-teas-ns-framework].

The following terminology is used in the document:

IETF network slice:

a well-defined composite of a set of endpoints, the connectivity requirements between subsets of these endpoints, and associated requirements; the term 'network slice' in this document refers to 'IETF network slice' as defined in [I-D.ietf-teas-ietf-network-slice-definition].

IETF Network Slice Controller (NSC):

controller that is used to realize an IETF network slice [I-D.ietf-teas-ietf-network-slice-definition].

**Slice policy:**

a policy construct that enables instantiation of mechanisms in support of IETF network slice specific control and data plane behaviors on select topological elements; the enforcement of a slice policy results in the creation of a slice aggregate.

**Slice aggregate:**

a collection of packets that match a slice policy selection criteria and are given the same forwarding treatment; a slice aggregate comprises of one or more IETF network slice traffic streams; the mapping of one or more IETF network slices to a slice aggregate is maintained by the IETF Network Slice Controller.

**Slice policy capable node:**

a node that supports one of the slice policy modes described in this document.

**Slice policy incapable node:**

a node that does not support any of the slice policy modes described in this document.

**Slice aggregate traffic:**

traffic that is forwarded over network resources associated with a specific slice aggregate.

**Slice aggregate path:**

a path that is setup over network resources associated with a specific slice aggregate.

**Slice aggregate packet:**

a packet that traverses network resources associated with a specific slice aggregate.

**Slice policy topology:**

a set of topological elements associated with a slice policy.

**Slice aggregate aware TE:**

a mechanism for TE path selection that takes into account the available network resources associated with a specific slice aggregate.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 1.2. Acronyms and Abbreviations

BA: Behavior Aggregate

CS: Class Selector

SS: Slice Selector

S-PHB: Slice policy Per Hop Behavior as described in Section 5.1.3

SSL: Slice Selector Label as described in Section 5.1.1

SSLI: Slice Selector Label Indicator

SLA: Service Level Agreement

SLO: Service Level Objective

Diffserv: Differentiated Services

MPLS: Multiprotocol Label Switching

LSP: Label Switched Path

RSVP: Resource Reservation Protocol

TE: Traffic Engineering

SR: Segment Routing

VRF: VPN Routing and Forwarding

## 2. Network Resource Slicing Membership

A slice aggregate can be instantiated over parts of an IP/MPLS network (e.g., all or specific network resources in the access, aggregation, or core network), and can stretch across multiple domains administered by a provider. A slice policy topology may include all or a sub-set of the physical nodes and links of an IP/MPLS network; it may be comprised of dedicated and/or shared network resources (e.g., in terms of processing power, storage, and bandwidth).

### 2.1. Dedicated Network Resources

Physical network resources may be fully dedicated to a specific slice aggregate. For example, traffic belonging to a slice aggregate can traverse dedicated network resources without being subjected to



contention from traffic of other slice aggregates. Dedicated network resource slicing allows for simple partitioning of the physical network resources amongst slice aggregates without the need to distinguish packets traversing the dedicated network resources since only one slice aggregate traffic stream can traverse the dedicated resource at any time.

## 2.2. Shared Network Resources

To optimize network utilization, sharing of the physical network resources may be desirable. In such case, the same physical network resource capacity is divided among multiple slice aggregates. Shared network resources can be partitioned in the data plane (for example by applying hardware policers and shapers) and/or partitioned in the control plane by providing a logical representation of the physical link that has a subset of the network resources available to it.

## 3. Path Selection

Path selection in a network can be network state dependent, or network state independent as described in Section 5.1 of [I-D.ietf-teas-rfc3272bis]. The latter is the choice commonly used by IGPs when selecting a best path to a destination prefix, while the former is used by ingress TE routers, or Path Computation Engines (PCEs) when optimizing the placement of a flow based on the current network resource utilization.

For example, when steering traffic on a delay optimized path, the IGP can use its link state database's view of the network topology to compute a path optimizing for the delay metric of each link in the network resulting in a cumulative lowest delay path.

When path selection is network state dependent, the path computation can leverage Traffic Engineering mechanisms (e.g., as defined in [RFC2702]) to compute feasible paths taking into account the incoming traffic demand rate and current state of network. This allows avoiding overly utilized links, and reduces the chance of congestion on traversed links.

To enable TE path placement, the link state is advertised with current reservations, thereby reflecting the available bandwidth on each link. Such link reservations may be maintained centrally on a network wide network resource manager, or distributed on devices (as usually done with RSVP). TE extensions exist today to allow IGPs (e.g., [RFC3630] and [RFC5305]), and BGP-LS [RFC7752] to advertise such link state reservations.

When network resource reservations are also slice aggregate aware, the link state can carry per slice aggregate state (e.g., reservable bandwidth). This allows path computation to take into account the specific network resources available for a slice aggregate when determining the path for a specific flow. In this case, we refer to the process of path placement and path provisioning as slice aggregate aware TE.

#### 4. Slice Policy Modes

A slice policy can be used to dictate if the partitioning of the shared network resources amongst multiple slice aggregates can be achieved by realizing slice aggregates in:

- a) data plane only, or
- b) control plane only, or
- c) both control and data planes.

##### 4.1. Data plane Slice Policy Mode

The physical network resources can be partitioned on network devices by applying a Per Hop forwarding Behavior (PHB) onto packets that traverse the network devices. In the Diffserv model, a Class Selector (CS) is carried in the packet and is used by transit nodes to apply the PHB that determines the scheduling treatment and drop probability for packets.

When data plane slice policy mode is applied, packets need to be forwarded on the specific slice aggregate network resources and need to be applied a specific forwarding treatment that is dictated in the slice policy (refer to Section 5.1 below). A Slice Selector (SS) MUST be carried in each packet to identify the slice aggregate that it belongs to.

The ingress node of a slice policy domain, in addition to marking packets with a Diffserv CS, MAY also add an SS to each slice aggregate packet. The transit nodes within a slice policy domain MAY use the SS to associate packets with a slice aggregate and to determine the Slice policy Per Hop Behavior (S-PHB) that is applied to the packet (refer to Section 5.1.3 for further details). The CS MAY be used to apply a Diffserv PHB on to the packet to allow differentiation of traffic treatment within the same slice aggregate.

When data plane only slice policy mode is used, routers may rely on a network state independent view of the topology to determine the best paths to reach destinations. In this case, the best path selection

dictates the forwarding path of packets to the destination. The SS field carried in each packet determines the specific S-PHB treatment along the selected path.

For example, the Segment-Routing Flexible Algorithm [I-D.ietf-lsr-flex-algo] may be deployed in a network to steer packets on the IGP computed lowest cumulative delay path. A slice policy may be used to allow links along the least latency path to share its data plane resources amongst multiple slice aggregates. In this case, the packets that are steered on a specific slice policy carry the SS field that enables routers (along with the Diffserv CS) to determine the S-PHB and enforce slice aggregate traffic streams.

#### 4.2. Control Plane Slice Policy Mode

The physical network resources in the network can be logically partitioned by having a representation of network resources appear in a virtual topology. The virtual topology can contain all or a subset of the physical network resources by applying specific topology filters on the native topology. The logical network resources that appear in the virtual topology can reflect a part, whole, or in-excess of the physical network resource capacity (when oversubscription is desirable). For example, a physical link bandwidth can be divided into fractions, each dedicated to a slice aggregate. Each fraction of the physical link bandwidth MAY be represented as a logical link in a virtual topology that is used when determining paths associated with a specific slice aggregate. The virtual topology associated with the slice policy can be used by routing protocols, or by the ingress/PCE when computing slice aggregate aware TE paths.

To perform network state dependent path computation in this mode (slice aggregate aware TE), the resource reservation on each link needs to be slice aggregate aware. Details of required IGP extensions to support SA-TE are described in [I-D.bestbar-lsr-slice-aware-te].

The same physical link may be member of multiple slice policies that instantiate different slice aggregates. The slice aggregate network resource availability on such a link is updated (and may be advertised) whenever new paths are placed in the network. The slice aggregate resource reservation, in this case, MAY be maintained on each device or off the device on a resource reservation manager that holds reservation states for those links in the network.

Multiple slice aggregates can form a group and share the available network resources allocated to each slice aggregate. In this case, a node can update the reservable bandwidth for each slice aggregate to

take into consideration the available bandwidth from other slice aggregates in the same group.

For illustration purposes, the diagram below represents bandwidth isolation or sharing amongst a group of slice aggregates. In Figure 1a, the slice aggregates: S\_AGG1, S\_AGG2, S\_AGG3 and S\_AGG4 are not sharing any bandwidths between each other. In Figure 1b, the slice aggregates: S\_AGG1 and S\_AGG2 can share the available bandwidth portion allocated to each amongst them. Similarly, S\_AGG3 and S\_AGG4 can share amongst themselves any available bandwidth allocated to them, but they cannot share available bandwidth allocated to S\_AGG1 or S\_AGG2. In both cases, the Max Reservable Bandwidth may exceed the actual physical link resource capacity to allow for over subscription.

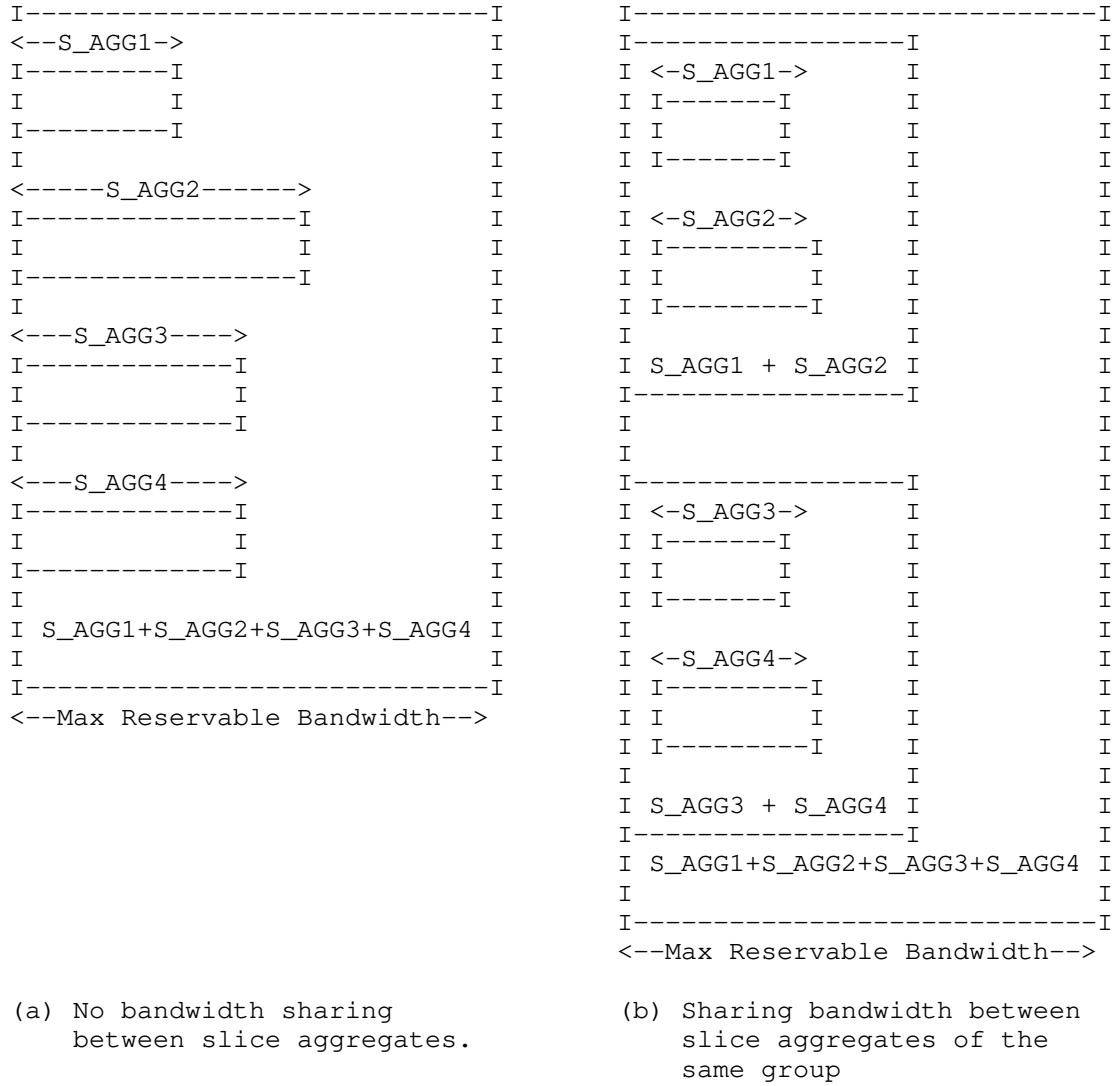


Figure 1: Bandwidth Isolation/Sharing.

#### 4.3. Data and Control Plane Slice Policy Mode

In order to support strict guarantees for slice aggregates, the network resources can be partitioned in both the control plane and data plane.

The control plane partitioning allows the creation of customized topologies per slice aggregate that routers or a Path Computation

Engine (PCE) can use to determine optimal path placement for specific demand flows (Slice aggregate aware TE).

The data plane partitioning protects slice aggregate traffic from network resource contention that could occur due to bursts in traffic from other slice aggregates traversing the same shared network resource.

## 5. Slice Policy Instantiation

A network slice can span multiple technologies and multiple administrative domains. Depending on the network slice consumer's requirements, a network slice can be differentiated from other network slices in terms of data, control or management planes.

The consumer of a network slice expresses their intent by specifying requirements rather than mechanisms to realize the slice. The requirements for a network slice can vary and can be expressed in terms of connectivity needs between end-points (point-to-point, point-to-multipoint or multipoint-to-multipoint) with customizable network capabilities that may include data speed, quality, latency, reliability, security, and services (refer to [I-D.ietf-teas-ietf-network-slice-definition] for more details). These capabilities are always provided based on a Service Level Agreement (SLA) between the network slice consumer and the provider.

The onus is on the network slice controller to consume the service layer slice intent and realize it with an appropriate slice policy. Multiple IETF network slices can be mapped to the same slice policy resulting in a slice aggregate. The network wide consistent slice policy definition is distributed to the devices in the network as shown in Figure 2. The specification of the network slice intent on the northbound interface of the controller and the mechanism used to map the network slice to a slice policy are outside the scope of this document.

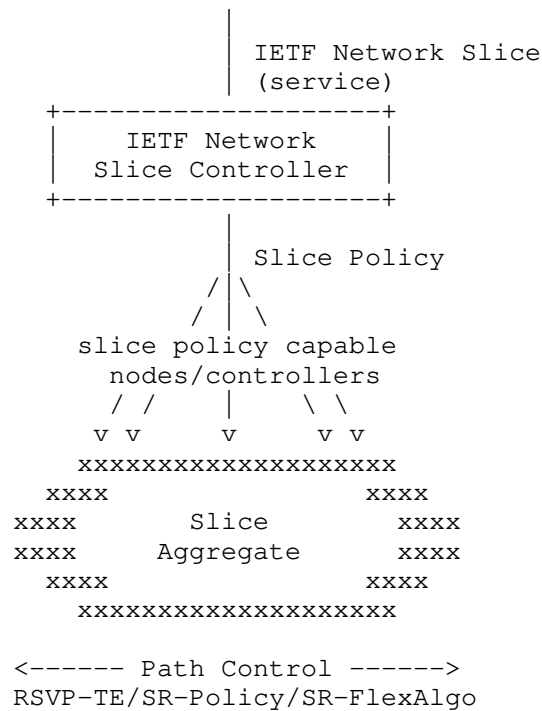


Figure 2: Slice Policy Instantiation.

### 5.1. Slice Policy Definition

The slice policy is network-wide construct that is consumed by network devices, and may include rules that control the following:

- o Data plane specific policies: This includes the SS, any firewall rules or flow-spec filters, and QoS profiles associated with the slice policy and any classes within it.
- o Control plane specific policies: This includes guaranteed bandwidth, any network resource sharing amongst slice policies, and reservation preference to prioritize any reservations of a specific slice policy over others.
- o Topology membership policies: This defines topology filter policies that dictate node/link/function network resource topology association for a specific slice policy.

There is a desire for flexibility in realizing network slices to support the services across networks consisting of products from multiple vendors. These networks may also be grouped into disparate

domains and deploy various path control technologies and tunnel techniques to carry traffic across the network. It is expected that a standardized data model for slice policy will facilitate the instantiation and management of slice aggregates on slice policy capable nodes. A YANG data model for the slice policy instantiation on network devices is described in [I-D.bestbar-teas-yang-slice-policy].

It is also possible to distribute the slice policy to network devices using several mechanisms, including protocols such as NETCONF or RESTCONF, or exchanging it using a suitable routing protocol that network devices participate in (such as IGP(s) or BGP). The extensions to enable specific protocols to carry a slice policy definition will be described in separate documents.

#### 5.1.1.1. Slice Policy Data Plane Selector

A router MUST be able to identify a packet belonging to a slice aggregate before it can apply the associated forwarding treatment or S-PHB. One or more fields within the packet MAY be used as an SS to do this.

##### Forwarding Address Based Slice Selector:

It is possible to assign a different forwarding address (or MPLS forwarding label in case of MPLS network) for each slice aggregate on a specific node in the network. [RFC3031] states in Section 2.1 that: 'Some routers analyze a packet's network layer header not merely to choose the packet's next hop, but also to determine a packet's "precedence" or "class of service"'. Assigning a unique forwarding address (or MPLS forwarding label) to each slice aggregate allows slice aggregate packets destined to a node to be distinguished by the destination address (or MPLS forwarding label) that is carried in the packet.

This approach requires maintaining per slice aggregate state for each destination in the network in both the control and data plane and on each router in the network. For example, consider a network slicing provider with a network composed of 'N' nodes, each with 'K' adjacencies to its neighbors. Assuming a node can be reached over 'M' different slice aggregates, the node assigns and advertises reachability to 'N' unique forwarding addresses, or MPLS forwarding labels. Similarly, each node assigns a unique forwarding address (or MPLS forwarding label) for each of its 'K' adjacencies to enable strict steering over the adjacency for each slice. The total number of control and data plane states that need to be stored and programmed in a router's forwarding is  $(N+K)*M$  states. Hence, as 'N', 'K', and 'M' parameters increase,



this approach suffers from scalability challenges in both the control and data planes.

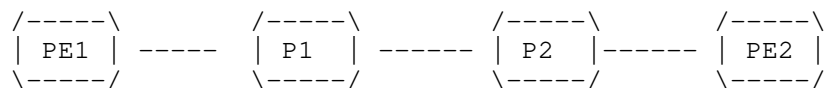
#### Global Identifier Based Slice Selector:

A slice policy MAY include a Global Identifier Slice Selector (GISS) field as defined in [I-D.kompella-mpls-mspl4fa] that is carried in each packet in order to associate it to a specific slice aggregate, independent of the forwarding address or MPLS forwarding label that is bound to the destination. Routers within the slice policy domain can use the forwarding address (or MPLS forwarding label) to determine the forwarding next-hop(s), and use the GISS field in the packet to infer the specific forwarding treatment that needs to be applied on the packet.

The GISS can be carried in one of multiple fields within the packet, depending on the dataplane used. For example, in MPLS networks, the GISS can be encoded within an MPLS label that is carried in the packet's MPLS label stack. All packets that belong to the same slice aggregate MAY carry the same GISS in the MPLS label stack. It is also possible to have multiple GISS's map to the same slice aggregate.

The GISS can be encoded in an MPLS label and may appear in several positions in the MPLS label stack. For example, the VPN service label may act as a GISS to allow VPN packets to be associated with a specific slice aggregate. In this case, a single VPN service label acting as a GISS MAY be allocated by all Egress PEs of a VPN. Alternatively, multiple VPN service labels MAY act as GISS's that map a single VPN to the same slice aggregate to allow for multiple Egress PEs to allocate different VPN service labels for a VPN. In other cases, a range of VPN service labels acting as multiple GISS's MAY map multiple VPN traffic to a single slice aggregate. An example of such deployment is shown in Figure 3.

SR Adj-SID:                      GISS (VPN service label) on PE2: 1001  
               9012: P1-P2  
               9023: P2-PE2



In

packet:

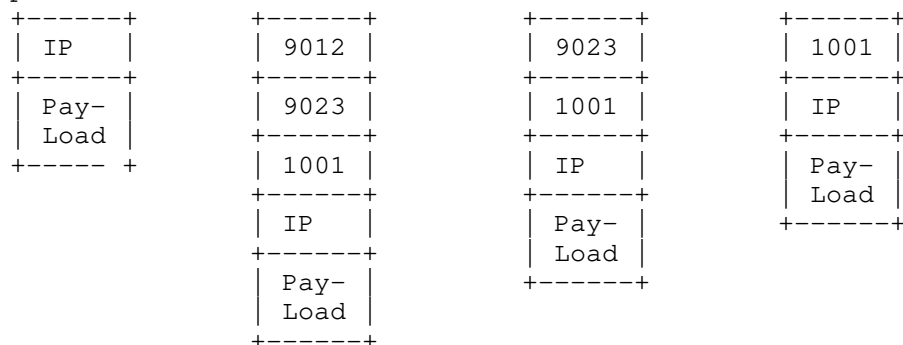


Figure 3: GISS or VPN label at bottom of label stack.

In some cases, the position of the GISS may not be at a fixed position in the MPLS label header. In this case, the GISS label can show up in any position in the MPLS label stack. To enable a transit router to identify the position of the GISS label, a special purpose label (ideally a base special purpose label (bSPL)) can be used as a GISS label indicator.

[I-D.kompella-mpls-mspl4fa] proposes a new bSPL called Forwarding Actions Identifier (FAI) that is assigned to alert of the presence of multiple actions and action data (including the presence of the GISS) that are carried within the MPLS label stack. The slice policy ingress boundary node, in this case, imposes two labels: the FAI label and a forwarding actions label that includes the GISS to identify the slice aggregate that packets belong to as shown in Figure 4.

[I-D.dekraene-mpls-slid-encoded-entropy-label-id] also proposes to repurpose the ELI/EL [RFC6790] to carry the Slice Identifier in order to minimize the size of the MPLS stack and ease incremental deployment.

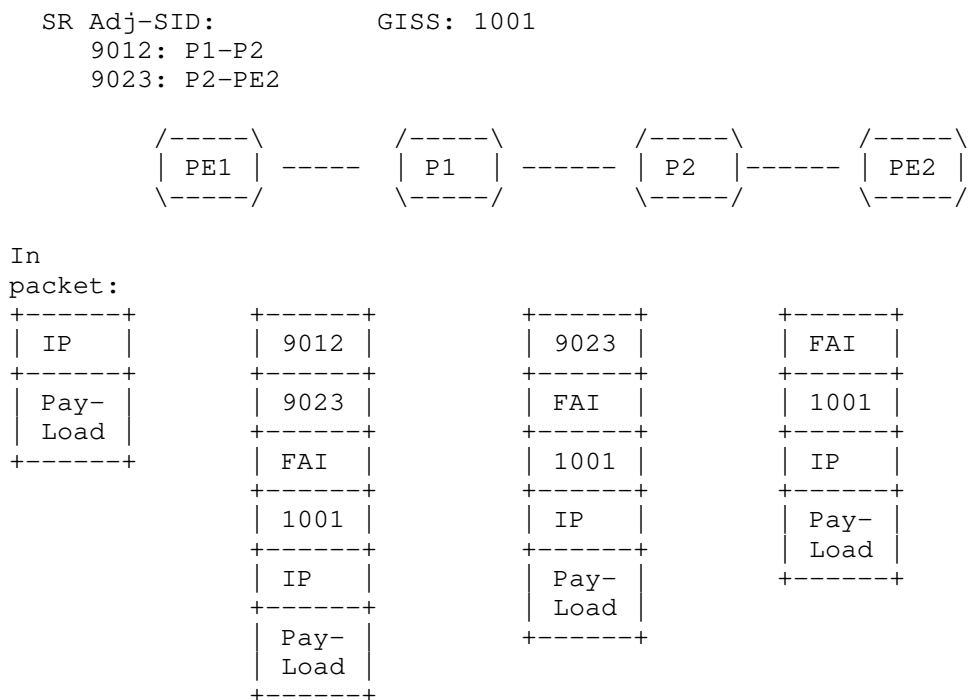


Figure 4: FAI and GISS label in the label stack.

When the slice is realized over an IP dataplane, the GISS can be encoded in the IP header. For example, the SSL can be encoded in portion of the IPv6 Flow Label field as described in [I-D.filsfils-spring-srv6-stateless-slice-id].

#### 5.1.2. Slice Policy Resource Reservation

Bandwidth and network resource allocation strategies for slice policies are essential to achieve optimal placement of paths within the network while still meeting the target SLOs.

Resource reservation allows for the managing of available bandwidth and for prioritization of existing allocations to enable preference-based preemption when contention on a specific network resource arises. Sharing of a network resource's available bandwidth amongst a group of slice policies may also be desirable. For example, a slice aggregate may not always be using all of its reservable bandwidth; this allows other slice policies in the same group to use the available bandwidth resources.

Congestion on shared network resources may result from sub-optimal placement of paths in different slice policies. When this occurs, preemption of some slice aggregate specific paths may be desirable to alleviate congestion. A preference based allocation scheme enables prioritization of slice aggregate paths that can be preempted.

Since network characteristics and its state can change over time, the slice policy topology and its state also need to be propagated in the network to enable ingress TE routers or Path Computation Engine (PCEs) to perform accurate path placement based on the current state of the slice policy network resources.

#### 5.1.3. Slice Policy Per Hop Behavior

In Diffserv terminology, the forwarding behavior that is assigned to a specific class is called a Per Hop Behavior (PHB). The PHB defines the forwarding precedence that a marked packet with a specific CS receives in relation to other traffic on the Diffserv-aware network.

A Slice policy Per Hop Behavior (S-PHB) is the externally observable forwarding behavior applied to a specific packet belonging to a slice aggregate. The goal of an S-PHB is to provide a specified amount of network resources for traffic belonging to a specific slice aggregate. A single slice policy may also support multiple forwarding treatments or services that can be carried over the same logical network.

The slice aggregate traffic may be identified at slice policy ingress boundary nodes by carrying a SS to allow routers to apply a specific forwarding treatment that guarantee the SLA(s).

With Differentiated Services (Diffserv) it is possible to carry multiple services over a single converged network. Packets requiring the same forwarding treatment are marked with a Class Selector (CS) at domain ingress nodes. Up to eight classes or Behavior Aggregates (BAs) may be supported for a given Forwarding Equivalence Class (FEC) [RFC2475]. To support multiple forwarding treatments over the same slice aggregate, a slice aggregate packet MAY also carry a Diffserv CS to identify the specific Diffserv forwarding treatment to be applied on the traffic belonging to the same slice policy.

At transit nodes, the CS field carried inside the packets are used to determine the specific PHB that determines the forwarding and scheduling treatment before packets are forwarded, and in some cases, drop probability for each packet.

#### 5.1.4. Slice Policy Topology

A key element of the slice policy is a customized topology that may include the full or subset of the physical network topology. The slice policy topology could also span multiple administrative domains and/or multiple dataplane technologies.

A slice policy topology can overlap or share a subset of links with another slice policy topology. A number of topology filtering policies can be defined as part of the slice policy to limit the specific topology elements that belong to a slice policy. For example, a topology filtering policy can leverage Resource Affinities as defined in [RFC2702] to include or exclude certain links for a specific slice aggregate. The slice policy may also include a reference to a predefined topology (e.g., derived from a Flexible Algorithm Definition (FAD) as defined in [I-D.ietf-lsr-flex-algo], or Multi-Topology ID as defined [RFC4915].

#### 5.2. Slice Policy Boundary

A network slice originates at the edge nodes of a network slice provider. Traffic that is steered over the corresponding slice aggregate may traverse slice policy capable interior nodes as well as slice policy incapable interior nodes.

The network slice may encompass one or more domains administered by a provider. For example, an organization's intranet or an ISP. The network provider is responsible for ensuring that adequate network resources are provisioned and/or reserved to support the SLAs offered by the network end-to-end.

##### 5.2.1. Slice Policy Edge Nodes

Slice policy edge nodes sit at the boundary of a network slice provider network and receive traffic that requires steering over network resources specific to a slice aggregate. These edge nodes are responsible for identifying slice aggregate specific traffic flows by possibly inspecting multiple fields from inbound packets (e.g., implementations may inspect IP traffic's network 5-tuple in the IP and transport protocol headers) to decide on which slice policy it can be steered.

Network slice ingress nodes may condition the inbound traffic at network boundaries in accordance with the requirements or rules of each service's SLAs. The requirements and rules for network slice services are set using mechanisms which are outside the scope of this document.

When data plane slice policy is applied, the slice policy ingress boundary nodes are responsible for adding a suitable SS onto packets that belong to specific slice aggregate. In addition, edge nodes MAY mark the corresponding Diffserv CS to differentiate between different types of traffic carried over the same slice aggregate.

#### 5.2.2. Slice Policy Interior Nodes

A slice policy interior node receives slice traffic and MAY be able to identify the packets belonging to a specific slice aggregate by inspecting the SS field carried inside each packet, or by inspecting other fields within the packet that may identify the traffic streams that belong to a specific slice aggregate. For example, when data plane slice policy is applied, interior nodes can use the SS carried within the packet to apply the corresponding S-PHB forwarding behavior. Nodes within the network slice provider network may also inspect the Diffserv CS within each packet to apply a per Diffserv class PHB within the slice policy, and allow differentiation of forwarding treatments for packets forwarded over the same slice aggregate network resources.

#### 5.2.3. Slice Policy Incapable Nodes

Packets that belong to a slice aggregate may need to traverse nodes that are slice policy incapable. In this case, several options are possible to allow the slice traffic to continue to be forwarded over such devices and be able to resume the slice policy forwarding treatment once the traffic reaches devices that are slice policy capable.

When data plane slice policy is applied, packets carry a SS to allow slice interior nodes to identify them. To enable end-to-end network slicing, the SS MUST be maintained in the packets as they traverse devices within the network - including slice policy incapable devices.

For example, when the SS is an MPLS label at the bottom of the MPLS label stack, packets can traverse over devices that are slice policy incapable without any further considerations. On the other hand, when the SSL is at the top of the MPLS label stack, packets can be bypassed (or tunneled) over the slice policy incapable devices towards the next device that supports slice policy as shown in Figure 5.

```
SR Node-SID:          SSL: 1001      @@@: slice policy enforced
1601: P1              ...: slice policy not enforced
1602: P2
1603: P3
1604: P4
1605: P5
```

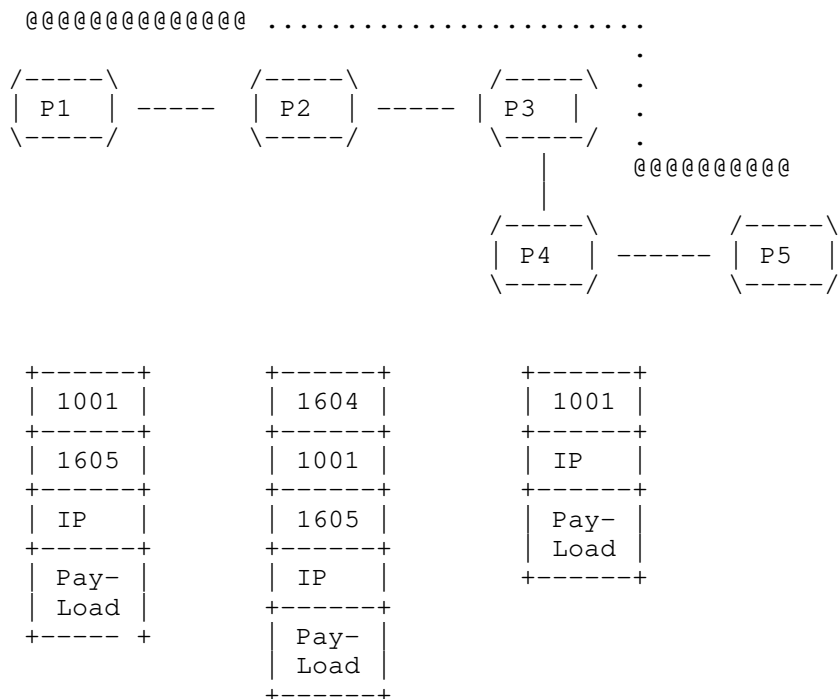


Figure 5: Extending network slice over slice policy incapable device(s).

#### 5.2.4. Combining Slice Policy Modes

It is possible to employ a combination of the slice policy modes that were discussed in Section 4 to realize a network slice. For example data and control plane slice policy mode can be employed in parts of a network, while control plane slice policy mode can be employed in the other parts of the network. The path selection, in such case, can take into account the slice aggregate specific available network resources. The SS carried within packets allow transit nodes to enforce the corresponding S-PHB on the parts of the network that apply the data plane slice policy mode. The SS can be maintained while traffic traverses nodes that do not enforce data plane slice

policy mode, and so slice PHB enforcement can resume once traffic traverses capable nodes.

### 5.3. Mapping Traffic on Slice Aggregates

The usual techniques to steer traffic onto paths can be applicable when steering traffic over paths established for a specific slice aggregate.

For example, one or more (layer-2 or layer-3) VPN services can be directly mapped to paths established for a slice aggregate. In this case, the per Virtual Routing and Forwarding (VRF) instance traffic that arrives on the Provider Edge (PE) router over external interfaces can be directly mapped to a specific slice aggregate path. External interfaces can be further partitioned (e.g., using VLANs) to allow mapping one or more VLANs to specific slice aggregate paths.

Another option is steer traffic to specific destinations directly over multiple slice policies. This allows traffic arriving on any external interface and targeted to such destinations to be directly steered over the slice paths.

A third option that can also be used is to utilize a data plane firewall filter or classifier to enable matching of several fields in the incoming packets to decide whether the packet is steered on a specific slice aggregate. This option allows for applying a rich set of rules to identify specific packets to be mapped to a slice aggregate. However, it requires data plane network resources to be able to perform the additional checks in hardware.

## 6. Control Plane Extensions

Routing protocols may need to be extended to carry additional per slice aggregate link state. For example, [RFC5305], [RFC3630], and [RFC7752] are ISIS, OSPF, and BGP protocol extensions to exchange network link state information to allow ingress TE routers and PCE(s) to do proper path placement in the network. The extensions required to support network slicing may be defined in other documents, and are outside the scope of this document.

The instantiation of a slice policy may need to be automated. Multiple options are possible to facilitate automation of distribution of a slice policy to capable devices.

For example, a YANG data model for the slice policy may be supported on network devices and controllers. A suitable transport (e.g., NETCONF [RFC6241], RESTCONF [RFC8040], or gRPC) may be used to enable configuration and retrieval of state information for slice policies



on network devices. The slice policy YANG data model is outside the scope of this document, and is defined in [I-D.bestbar-teas-yang-slice-policy].

## 7. Applicability to Path Control Technologies

The slice policy modes described in this document are agnostic to the technology used to setup paths that carry slice aggregate traffic. One or more paths connecting the endpoints of the mapped IETF network slices may be selected to steer the corresponding traffic streams over the resources allocated for the slice aggregate.

For example, once the feasible paths within a slice policy topology are selected, it is possible to use RSVP-TE protocol [RFC3209] to setup or signal the LSPs that would be used to carry the slice aggregate traffic. Specific extensions to RSVP-TE protocol to enable signaling of slice aggregate aware RSVP LSPs are outside the scope of this document.

Alternatively, Segment Routing (SR) [RFC8402] may be used and the feasible paths can be realized by steering over specific segments or segment-lists using an SR policy. Further details on how the slice policy modes presented in this document can be realized over an SR network is discussed in [I-D.bestbar-spring-scalable-ns], and [I-D.bestbar-lsr-spring-sa].

## 8. IANA Considerations

This document has no IANA actions.

## 9. Security Considerations

The main goal of network slicing is to allow for varying treatment of traffic from multiple different network slices that are utilizing a common network infrastructure and to allow for different levels of services to be provided for traffic traversing a given network resource.

A variety of techniques may be used to achieve this, but the end result will be that some packets may be mapped to specific resources and may receive different (e.g., better) service treatment than others. The mapping of network traffic to a specific slice policy is indicated primarily by the SS, and hence an adversary may be able to utilize resources allocated to a specific slice policy by injecting packets carrying the same SS field in their packets.

Such theft-of-service may become a denial-of-service attack when the modified or injected traffic depletes the resources available to forward legitimate traffic belonging to a specific slice policy.

The defense against this type of theft and denial-of-service attacks consists of a combination of traffic conditioning at slice policy domain boundaries with security and integrity of the network infrastructure within a slice policy domain.

## 10. Acknowledgement

The authors would like to thank Krzysztof Szarkowicz, Swamy SRK, Navaneetha Krishnan, Prabhu Raj Villadathu Karunakaran and Jie Dong for their review of this document, and for providing valuable feedback on it.

## 11. Contributors

The following individuals contributed to this document:

Colby Barth  
Juniper Networks  
Email: cbarth@juniper.net

Srihari R. Sangli  
Juniper Networks  
Email: ssangli@juniper.net

Chandra Ramachandran  
Juniper Networks  
Email: csekar@juniper.net

## 12. References

### 12.1. Normative References

- [I-D.bestbar-lsr-slice-aware-te]  
Britto, W., Shetty, R., Barth, C., Wen, B., Peng, S., and R. Chen, "IGP Extensions for Support of Slice Aggregate Aware Traffic Engineering", draft-bestbar-lsr-slice-aware-te-00 (work in progress), February 2021.
- [I-D.bestbar-lsr-spring-sa]  
Saad, T., Beeram, V. P., Chen, R., Peng, S., Wen, B., and D. Ceccarelli, "IGP Extensions for SR Slice Aggregate SIDs", draft-bestbar-lsr-spring-sa-00 (work in progress), February 2021.

- [I-D.bestbar-spring-scalable-ns]  
Saad, T., Beeram, V. P., Chen, R., Peng, S., Wen, B., and D. Ceccarelli, "Scalable Network Slicing over SR Networks", draft-bestbar-spring-scalable-ns-01 (work in progress), February 2021.
- [I-D.bestbar-teas-yang-slice-policy]  
Saad, T., Beeram, V. P., Wen, B., Ceccarelli, D., Peng, S., Chen, R., Contreras, L. M., and X. Liu, "YANG Data Model for Slice Policy", draft-bestbar-teas-yang-slice-policy-00 (work in progress), February 2021.
- [I-D.decraene-mpls-slid-encoded-entropy-label-id]  
Decraene, B., Filsfils, C., Henderickx, W., Saad, T., Beeram, V. P., and L. Jalil, "Using Entropy Label for Network Slice Identification in MPLS networks.", draft-decraene-mpls-slid-encoded-entropy-label-id-01 (work in progress), February 2021.
- [I-D.filsfils-spring-srv6-stateless-slice-id]  
Filsfils, C., Clad, F., Camarillo, P., and K. Raza, "Stateless and Scalable Network Slice Identification for SRv6", draft-filsfils-spring-srv6-stateless-slice-id-02 (work in progress), January 2021.
- [I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-15 (work in progress), April 2021.
- [I-D.kompella-mpls-mspl4fa]  
Kompella, K., Beeram, V. P., Saad, T., and I. Meilik, "Multi-purpose Special Purpose Label for Forwarding Actions", draft-kompella-mpls-mspl4fa-00 (work in progress), February 2021.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.

- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <<https://www.rfc-editor.org/info/rfc4915>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

## 12.2. Informative References

- [I-D.ietf-teas-ietf-network-slice-definition]  
Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-01 (work in progress), February 2021.

- [I-D.ietf-teas-rfc3272bis]  
Farrel, A., "Overview and Principles of Internet Traffic Engineering", draft-ietf-teas-rfc3272bis-11 (work in progress), April 2021.
- [I-D.nsdtd-teas-ns-framework]  
Gray, E. and J. Drake, "Framework for IETF Network Slices", draft-nsdt-teas-ns-framework-05 (work in progress), February 2021.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <<https://www.rfc-editor.org/info/rfc2475>>.
- [RFC2702] Awduche, D., Malcolm, J., Agoghua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <<https://www.rfc-editor.org/info/rfc2702>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

#### Authors' Addresses

Tarek Saad  
Juniper Networks

Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Vishnu Pavan Beeram  
Juniper Networks

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

Bin Wen  
Comcast

Email: [Bin\\_Wen@cable.comcast.com](mailto:Bin_Wen@cable.comcast.com)

Daniele Ceccarelli  
Ericsson

Email: daniele.ceccarelli@ericsson.com

Joel Halpern  
Ericsson

Email: joel.halpern@ericsson.com

Shaofu Peng  
ZTE Corporation

Email: peng.shaofu@zte.com.cn

Ran Chen  
ZTE Corporation

Email: chen.ran@zte.com.cn

Xufeng Liu  
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Luis M. Contreras  
Telefonica

Email: luismiguel.contrerasmurillo@telefonica.com

Reza Rokui  
Nokia

Email: reza.rokui@nokia.com

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 5 November 2022

T. Saad  
V. Beeram  
Juniper Networks  
J. Dong  
Huawei Technologies  
B. Wen  
Comcast  
D. Ceccarelli  
J. Halpern  
Ericsson  
S. Peng  
R. Chen  
ZTE Corporation  
X. Liu  
Volta Networks  
L. Contreras  
Telefonica  
R. Rokui  
Ciena  
L. Jalil  
Verizon  
4 May 2022

Realizing Network Slices in IP/MPLS Networks  
draft-bestbar-teas-ns-packet-10

Abstract

Realizing network slices may require the Service Provider to have the ability to partition a physical network into multiple logical networks of varying sizes, structures, and functions so that each slice can be dedicated to specific services or customers. Multiple network slices can be realized on the same network while ensuring slice elasticity in terms of network resource allocation. This document describes a scalable solution to realize network slicing in IP/MPLS networks by supporting multiple services on top of a single physical network by relying on compliant domains and nodes to provide forwarding treatment (scheduling, drop policy, resource usage) on to packets that carry identifiers that indicate the slicing service that is to be applied to the packets.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 5 November 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 3  |
| 1.1. Terminology . . . . .   | 5  |
| 1.2. Acronyms and Abbreviations . . . . .                                | 6  |
| 2. Network Resource Slicing Membership . . . . .                         | 7  |
| 3. IETF Network Slice Realization . . . . .                              | 8  |
| 3.1. Network Topology Filters . . . . .                                  | 9  |
| 3.2. IETF Network Slice Service Request . . . . .                        | 9  |
| 3.3. Slice-Flow Aggregation . . . . .                                    | 10 |
| 3.4. Path Placement over NRP Filter Topology . . . . .                   | 10 |
| 3.5. NRP Policy Installation . . . . .                                   | 10 |
| 3.6. Path Instantiation . . . . .  | 10 |
| 3.7. Service Mapping . . . . .   | 11 |
| 4. Network Resource Partition Modes . . . . .                            | 11 |
| 4.1. Data plane Network Resource Partition Mode . . . . .                | 11 |
| 4.2. Control Plane Network Resource Partition Mode . . . . .             | 12 |
| 4.3. Data and Control Plane Network Resource Partition Mode . . . . .    | 14 |
| 5. Network Resource Partition Instantiation . . . . .                    | 14 |
| 5.1. NRP Policy Definition . . . . .                                     | 14 |
| 5.1.1. Network Resource Partition - Flow-Aggregate<br>Selector . . . . . | 15 |



|        |  |    |
|--------|--|----|
| 5.1.2. | Network Resource Partition Resource Reservation . . .                            | 18 |
| 5.1.3. | Network Resource Partition Per Hop Behavior . . . . .                            | 19 |
| 5.1.4. | Network Resource Partition Topology . . . . .                                    | 20 |
| 5.2.   | Network Resource Partition Boundary . . . . .                                    | 20 |
| 5.2.1. | Network Resource Partition Edge Nodes . . . . .                                  | 20 |
| 5.2.2. | Network Resource Partition Interior Nodes . . . . .                              | 21 |
| 5.2.3. | Network Resource Partition Incapable Nodes . . . . .                             | 21 |
| 5.2.4. | Combining Network Resource Partition Modes . . . . .                             | 22 |
| 6.     | Mapping Traffic on Slice-Flow Aggregates . . . . .                               | 23 |
| 6.1.   | Network Slice-Flow Aggregate Relationships . . . . .                             | 23 |
| 7.     | Path Selection and Instantiation . . . . .                                       | 24 |
| 7.1.   | Applicability of Path Selection to Slice-Flow<br>Aggregates . . . . .            | 24 |
| 7.2.   | Applicability of Path Control Technologies to Slice-Flow<br>Aggregates . . . . . | 24 |
| 7.2.1. | RSVP-TE Based Slice-Flow Aggregate Paths . . . . .                               | 25 |
| 7.2.2. | SR Based Slice-Flow Aggregate Paths . . . . .                                    | 25 |
| 8.     | Network Resource Partition Protocol Extensions . . . . .                         | 25 |
| 9.     | Outstanding Issues . . . . .   | 26 |
| 10.    | IANA Considerations . . . . .  | 27 |
| 11.    | Security Considerations . . . . .  | 27 |
| 12.    | Acknowledgement . . . . .  | 27 |
| 13.    | Contributors . . . . .   | 27 |
| 14.    | References . . . . .   | 28 |
| 14.1.  | Normative References . . . . .   | 28 |
| 14.2.  | Informative References . . . . .   | 28 |
|        | Authors' Addresses . . . . .   | 30 |

## 1. Introduction

Network slicing allows a Service Provider to create independent and logical networks on top of a shared physical network infrastructure. Such network slices can be offered to customers or used internally by the Service Provider to enhance the delivery of their service offerings. A Service Provider can also use network slicing to structure and organize the elements of its infrastructure. The solution discussed in this document works with any path control technology (such as RSVP-TE, or SR) that can be used by a Service Provider to realize network slicing in IP/MPLS networks.

[I-D.ietf-teas-ietf-network-slices] provides the definition of a network slice for use within the IETF and discusses the general framework for requesting and operating IETF Network Slices, their characteristics, and the necessary system components and interfaces. It also discusses the function of an IETF Network Slice Controller and the requirements on its northbound and southbound interfaces.

This document introduces the notion of a Slice-Flow Aggregate which comprises of one or more IETF network slice traffic streams. It also describes the Network Resource Partition (NRP) and the NRP Policy that can be used to instantiate control and data plane behaviors on select topological elements associated with the NRP that supports a Slice-Flow Aggregate - refer Section 5.1 for further details.

The IETF Network Slice Controller is responsible for the aggregation of multiple IETF network traffic streams into a Slice-Flow Aggregate, and for maintaining the mapping required between them. The mechanisms used by the controller to determine the mapping of one or more IETF network slice to a Slice-Flow Aggregate are outside the scope of this document. The focus of this document is on the mechanisms required at the device level to address the requirements of network slicing in packet networks.

In a Diffserv (DS) domain [RFC2475], packets requiring the same forwarding treatment (scheduling and drop policy) are classified and marked with the respective Class Selector (CS) Codepoint (or the Traffic Class (TC) field for MPLS packets [RFC5462]) at the DS domain ingress nodes. Such packets are said to belong to a Behavior Aggregate (BA) that has a common set of behavioral characteristics or a common set of delivery requirements. At transit nodes, the CS is inspected to determine the specific forwarding treatment to be applied before the packet is forwarded. A similar approach is adopted in this document to realize network slicing. The solution proposed in this document does not mandate Diffserv to be enabled in the network to provide a specific forwarding treatment.

When logical networks associated with an NRP are realized on top of a shared physical network infrastructure, it is important to steer traffic on the specific network resources partition that is allocated for a given Slice-Flow Aggregate. In packet networks, the packets of a specific Slice-Flow Aggregate may be identified by one or more specific fields carried within the packet. An NRP ingress boundary node (where Slice-Flow Aggregate traffic enters the NRP) populates the respective field(s) in packets that are mapped to a Slice-Flow Aggregate in order to allow interior NRP nodes to identify and apply the specific Per NRP Hop Behavior (NRP-PHB) associated with the Slice-Flow Aggregate. The NRP-PHB defines the scheduling treatment and, in some cases, the packet drop probability.

If Diffserv is enabled within the network, the Slice-Flow Aggregate traffic can further carry a Diffserv CS to enable differentiation of forwarding treatments for packets within a Slice-Flow Aggregate.

For example, when using MPLS as a dataplane, it is possible to identify packets belonging to the same Slice-Flow Aggregate by carrying an identifier in an MPLS Label Stack Entry (LSE). Additional Diffserv classification may be indicated in the Traffic Class (TC) bits of the global MPLS label to allow further differentiation of forwarding treatments for traffic traversing the same NRP.

This document covers different modes of NRPs and discusses how each mode can ensure proper placement of Slice-Flow Aggregate paths and respective treatment of Slice-Flow Aggregate traffic.

### 1.1. Terminology

The reader is expected to be familiar with the terminology specified in [I-D.ietf-teas-ietf-network-slices].

The following terminology is used in the document:

IETF Network Slice:

refer to the definition of 'IETF network slice' in [I-D.ietf-teas-ietf-network-slices].

IETF Network Slice Controller (NSC):

refer to the definition in [I-D.ietf-teas-ietf-network-slices].

Network Resource Partition:

refer to the definition in [I-D.ietf-teas-ietf-network-slices].

Slice-Flow Aggregate:

a collection of packets that match an NRP Policy and are given the same forwarding treatment; a Slice-Flow Aggregate comprises of one or more IETF network slice traffic streams; the mapping of one or more IETF network slices to a Slice-Flow Aggregate is maintained by the IETF Network Slice Controller. The boundary nodes MAY also maintain a mapping of specific IETF network slice service(s) to a SFA.

Network Resource Partition Policy (NRP):

a policy construct that enables instantiation of mechanisms in support of IETF network slice specific control and data plane behaviors on select topological elements; the enforcement of an NRP Policy results in the creation of an NRP.

NRP Identifier (NRP-ID):

an identifier that is globally unique within an NRP domain and that can be used in the control or management plane to identify the resources associated with the NRP.

**NRP Capable Node:**

a node that supports one of the NRP modes described in this document.

**NRP Incapable Node:**

a node that does not support any of the NRP modes described in this document.

**Slice-Flow Aggregate Path:**

a path that is setup over the NRP that is associated with a specific Slice-Flow Aggregate.

**Slice-Flow Aggregate Packet:**

a packet that traverses over the NRP that is associated with a specific Slice-Flow Aggregate.

**NRP Filter Topology:**

a set of topological elements associated with a Network Resource Partition.

**NRP state aware TE (NRP-TE):**

a mechanism for TE path selection that takes into account the available network resources associated with a specific NRP.

## 1.2. Acronyms and Abbreviations

BA: Behavior Aggregate

CS: Class Selector

NRP-PHB: NRP Per Hop Behavior as described in Section 5.1.3

FAS: Flow Aggregate Selector

FASL: Flow Aggregate Selector Label as described in Section 5.1.1

SLA: Service Level Agreements

SLO: Service Level Objectives

SLE: Service Level Expectations

Diffserv: Differentiated Services

MPLS: Multiprotocol Label Switching

LSP: Label Switched Path

RSVP: Resource Reservation Protocol

TE: Traffic Engineering

SR: Segment Routing

VRF: VPN Routing and Forwarding

AC: Attachment Circuit

CE: Customer Edge

PE: Provider Edge

PCEP: Path Computation Element (PCE) Communication Protocol (PCEP)

## 2. Network Resource Slicing Membership

An NRP that supports a Slice-Flow Aggregate can be instantiated over parts of an IP/MPLS network (e.g., all or specific network resources in the access, aggregation, or core network), and can stretch across multiple domains administered by a provider. The NRP topology may be comprised of dedicated and/or shared network resources (e.g., in terms of processing power, storage, and bandwidth).

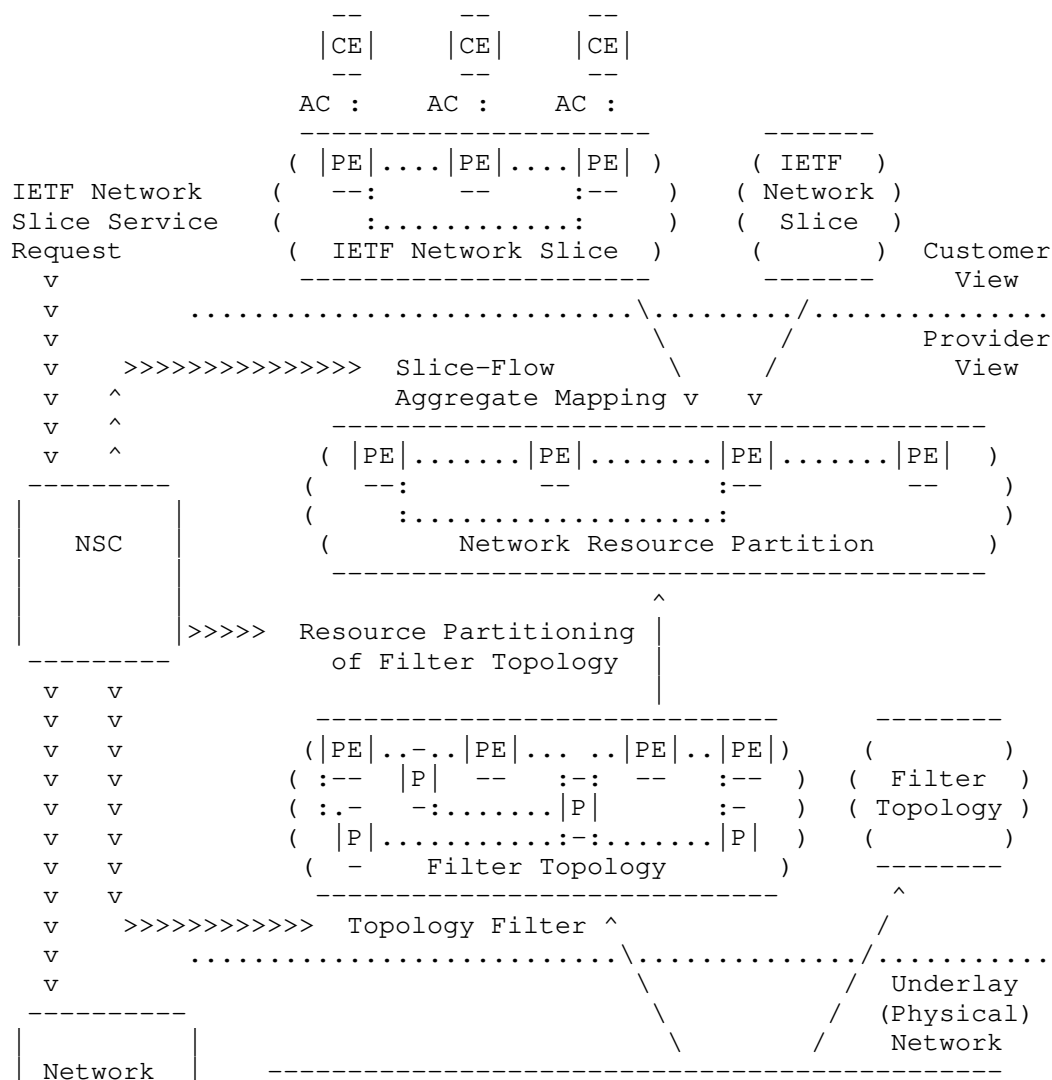
The physical network resources may be fully dedicated to a specific Slice-Flow Aggregate. For example, traffic belonging to a Slice-Flow Aggregate can traverse dedicated network resources without being subjected to contention from traffic of other Slice-Flow Aggregates. Dedicated physical network resource slicing allows for simple partitioning of the physical network resources amongst Slice-Flow Aggregates without the need to distinguish packets traversing the dedicated network resources since only one Slice-Flow Aggregate traffic stream can traverse the dedicated resource at any time.

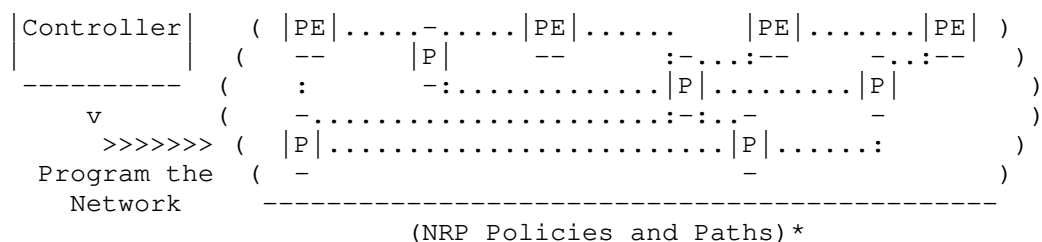
To optimize network utilization, sharing of the physical network resources may be desirable. In such case, the same physical network resource capacity is divided among multiple NRPs that support multiple Slice-Flow Aggregates. The shared physical network resources can be partitioned in the data plane (for example by applying hardware policers and shapers) and/or partitioned in the control plane by providing a logical representation of the physical link that has a subset of the network resources available to it.

### 3. IETF Network Slice Realization

Figure 1 describes the steps required to realize an IETF network slice service in a provider network using the solution proposed in this document. While Figure 4 of [I-D.ietf-teas-ietf-network-slices] provides an abstract architecture of an IETF Network Slice, this section intends to offer a realization of that architecture specific for IP/MPLS packet networks.

Each of the steps is further elaborated on in a subsequent section.





\* : NRP Policy installation and path placement can be centralized or distributed.

Figure 1: IETF network slice realization steps.

### 3.1. Network Topology Filters

The Physical Network may be filtered into a number of Filter Topologies. Filter actions may include selection of specific nodes and links according to their capabilities and are based on network-wide policies. The resulting topologies can be used to host IETF Network Slices and provide a useful way for the network operator to know that all of the resources they are using to plan a network slice meet specific SLOs. This step can be done offline during planning activity, or could be performed dynamically as new demands arise.

Section 5.1.4 describes how topology filters can be associated with the NRP instantiated by the NRP Policy.

### 3.2. IETF Network Slice Service Request

The customer requests an IETF Network Slice Service specifying the CE-AC-PE points of attachment, the connectivity matrix, and the SLOs/SLEs as described in [I-D.ietf-teas-ietf-network-slices]. These capabilities are always provided based on a Service Level Agreement (SLA) between the network slice costumer and the provider.

This defines the traffic flows that need to be supported when the slice is realized. Depending on the mechanism and encoding of the Attachment Circuit (AC), the IETF Network Slice Service may also include information that will allow the operator's controllers to configure the PEs to determine what customer traffic is intended for this IETF Network Slice.

IETF Network Slice Service Requests are likely to arrive at various times in the life of the network, and may also be modified.

### 3.3. Slice-Flow Aggregation

A network may be called upon to support very many IETF Network Slices, and this could present scaling challenges in the operation of the network. In order to overcome this, the IETF Network Slice streams may be aggregated into groups according to similar characteristics.

A Slice-Flow Aggregate is a construct that comprises the traffic flows of one or more IETF Network Slices. The mapping of IETF Network Slices into an Slice-Flow Aggregate is a matter of local operator policy is a function executed by the Controller. The Slice-Flow Aggregate may be preconfigured, created on demand, or modified dynamically.

### 3.4. Path Placement over NRP Filter Topology

Depending on the underlying network technology, the paths are selected in the network in order to best deliver the SLOs for the different services carried by the Slice-Flow Aggregate. The path placement function (carried on ingress node or by a controller) is performed on the Filter Topology that is selected to support the Slice-Flow Aggregate.

Note that this step may indicate the need to increase the capacity of the underlying Filter Topology or to create a new Filter Topology.

### 3.5. NRP Policy Installation

A Controller function programs the physical network with policies for handling the traffic flows belonging to the Slice-Flow Aggregate. These policies instruct underlying routers how to handle traffic for a specific Slice-Flow Aggregate: the routers correlate markers present in the packets that belong to the Slice-Flow Aggregate. The way in which the NRP Policy is installed in the routers and the way that the traffic is marked is implementation specific. The NRP Policy instantiation in the network is further described in Section 5.

### 3.6. Path Instantiation

Depending on the underlying network technology, a Controller function may install the forwarding state specific to the Slice-Flow Aggregate so that traffic is routed along paths derived in the Path Placement step described in Section 3.4. The way in which the paths are instantiated is implementation specific.



### 3.7. Service Mapping

The edge points can be configured to support the network slice service by mapping the customer traffic to Slice-Flow Aggregates, possibly using information supplied when the IETF network slice service was requested. The edge points may also be instructed to mark the packets so that the network routers will know which policies and routing instructions to apply. The steering of traffic onto Slice-Flow Aggregate paths is further described in Section 6.

## 4. Network Resource Partition Modes

An NRP Policy can be used to dictate if the network resource partitioning of the shared network resources among multiple Slice-Flow Aggregates can be achieved:

- a) in data plane only,
- b) in control plane only, or
- c) in both control and data planes.

### 4.1. Data plane Network Resource Partition Mode

The physical network resources can be partitioned on network devices by applying a Per Hop forwarding Behavior (PHB) onto packets that traverse the network devices. In the Diffserv model, a Class Selector (CS) codepoint is carried in the packet and is used by transit nodes to apply the PHB that determines the scheduling treatment and drop probability for packets.

When data plane NRP mode is applied, packets need to be forwarded on the specific NRP that supports the Slice-Flow Aggregate to ensure the proper forwarding treatment dictated in the NRP Policy is applied (refer to Section 5.1 below). In this case, a Flow Aggregate Selector (FAS) must be carried in each packet to identify the Slice-Flow Aggregate that it belongs to.

The ingress node of an NRP domain adds a FAS field if one is not already present in each Slice-Flow Aggregate packet. In the data plane NRP mode, the transit nodes within an NRP domain use the FAS to associate packets with a Slice-Flow Aggregate and to determine the Network Resource Partition Per Hop Behavior (NRP-PHB) that is applied to the packet (refer to Section 5.1.3 for further details). The CS is used to apply a Diffserv PHB on to the packet to allow differentiation of traffic treatment within the same Slice-Flow Aggregate.

When data plane only NRP mode is used, routers may rely on a network state independent view of the topology to determine the best paths. In this case, the best path selection dictates the forwarding path of packets to the destination. The FAS field carried in each packet determines the specific NRP-PHB treatment along the selected path.

#### 4.2. Control Plane Network Resource Partition Mode

Multiple NRPs can be realized over the same set of physical resources. Each NRP is identified by an identifier (NRP-ID) that is globally unique within the NRP domain. The NRP state reservations for each NRP can be maintained on the network element or on a controller.

The network reservation states for a specific partition can be represented in a topology that contains all or a subset of the physical network elements (nodes and links) and reflect the network state reservations in that NRP. The logical network resources that appear in the NRP topology can reflect a part, whole, or in-excess of the physical network resource capacity (e.g., when oversubscription is desirable).

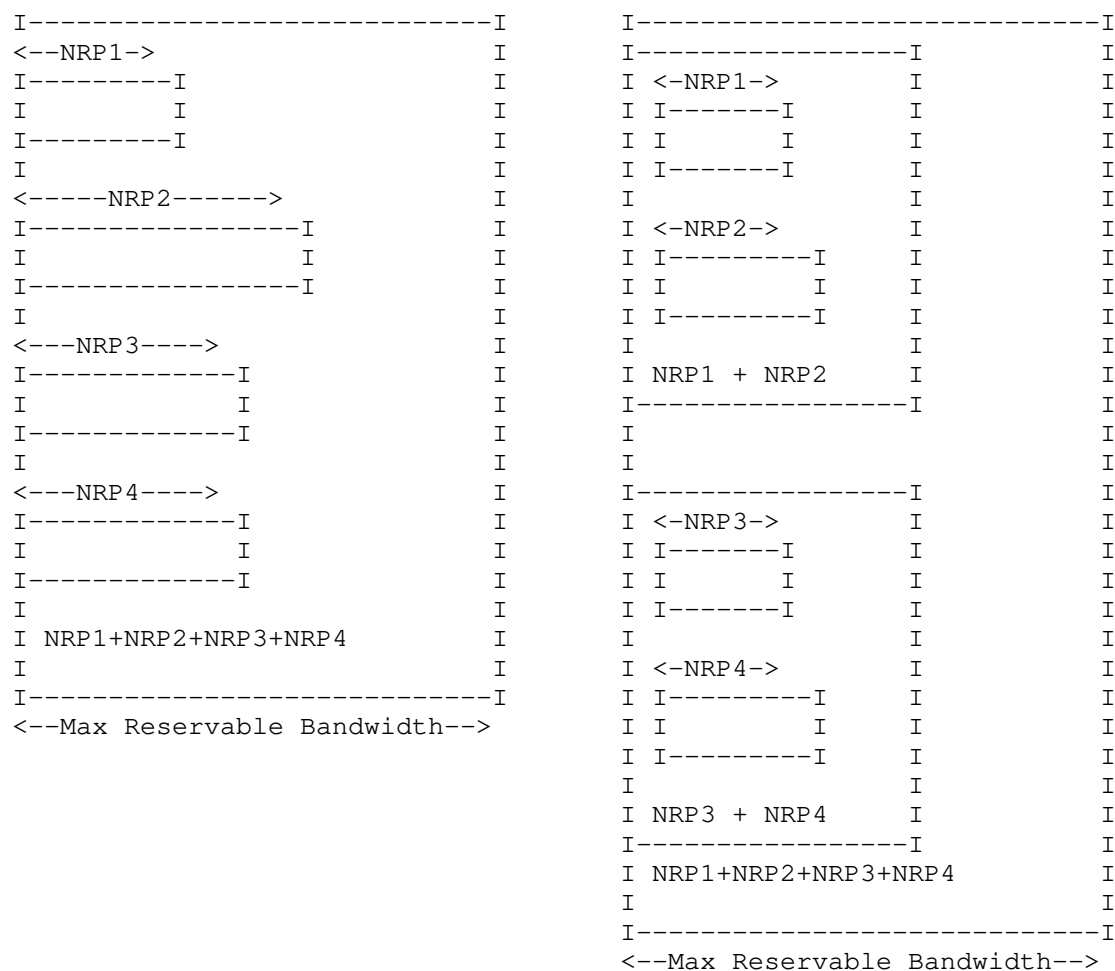
For example, the physical link bandwidth can be divided into fractions, each dedicated to an NRP that supports a Slice-Flow Aggregate. The topology associated with the NRP supporting a Slice-Flow Aggregate can be used by routing protocols, or by the ingress/PCE when computing NRP state aware TE paths.

To perform NRP state aware Traffic Engineering (NRP-TE), the resource reservation on each link needs to be NRP aware. The NRP reservations state can be managed locally on the device or off device (e.g. on a controller).

The same physical link may be member of multiple slice policies that instantiate different NRPs. The NRP reservable or utilized bandwidth on such a link is updated (and may be advertised) whenever new paths are placed in the network. The NRP reservation state, in this case, is maintained on each device or off the device on a resource reservation manager that holds reservation states for those links in the network.

Multiple NRPs that support Slice-Flow Aggregates can form a group and share the available network resources allocated to each. In this case, a node can update the reservable bandwidth for each NRP to take into consideration the available bandwidth from other NRPs in the same group.

For illustration purposes, Figure 2 describes bandwidth partitioning or sharing amongst a group of NRPs. In Figure 2a, the NRPs identified by the following NRP-IDs: NRP1, NRP2, NRP3 and NRP4 are not sharing any bandwidths between each other. In Figure 2b, the NRPs: NRP1 and NRP2 can share the available bandwidth portion allocated to each amongst them. Similarly, NRP3 and NRP4 can share amongst themselves any available bandwidth allocated to them, but they cannot share available bandwidth allocated to NRP1 or NRP2. In both cases, the Max Reservable Bandwidth may exceed the actual physical link resource capacity to allow for over subscription.



(a) No bandwidth sharing  
between NRPs.

(b) Sharing bandwidth between  
NRPs of the same group.

Figure 2: Bandwidth isolation/sharing among NRPs.

#### 4.3. Data and Control Plane Network Resource Partition Mode

In order to support strict guarantees for Slice-Flow Aggregates, the network resources can be partitioned in both the control plane and data plane.

The control plane partitioning allows the creation of customized topologies per NRP that each supports a Slice-Flow Aggregate. The ingress routers or a Path Computation Engine (PCE) may use the customized topologies and the NRP state to determine optimal path placement for specific demand flows using NRP-TE.

The data plane partitioning provides isolation for Slice-Flow Aggregate traffic, and protection when resource contention occurs due to bursts of traffic from other Slice-Flow Aggregate traffic that traverses the same shared network resource.

#### 5. Network Resource Partition Instantiation

A network slice can span multiple technologies and multiple administrative domains. Depending on the network slice customer requirements, a network slice can be differentiated from other network slices in terms of data, control, and management planes.

The customer of a network slice service expresses their intent by specifying requirements rather than mechanisms to realize the slice as described in Section 3.2.

The network slice controller is fed with the network slice service intent and realizes it with an appropriate Network Resource Partition Policy (NRP Policy). Multiple IETF network slices are mapped to the same Slice-Flow Aggregate as described in Section 3.3.

The network wide consistent NRP Policy definition is distributed to the devices in the network as shown in Figure 1. The specification of the network slice intent on the northbound interface of the controller and the mechanism used to map the network slice to a Slice-Flow Aggregate are outside the scope of this document and will be addressed in separate documents.

##### 5.1. NRP Policy Definition

The NRP Policy is network-wide construct that is supplied to network devices, and may include rules that control the following:

- \* Data plane specific policies: This includes the FAS, any firewall rules or flow-spec filters, and QoS profiles associated with the NRP Policy and any classes within it.
- \* Control plane specific policies: This includes bandwidth reservations, any network resource sharing amongst slice policies, and reservation preference to prioritize reservations of a specific NRP over others.
- \* Topology membership policies: This defines the topology filter policies that dictate node/link/function membership to a specific NRP.

There is a desire for flexibility in realizing network slices to support the services across networks consisting of implementations from multiple vendors. These networks may also be grouped into disparate domains and deploy various path control technologies and tunnel techniques to carry traffic across the network. It is expected that a standardized data model for NRP Policy will facilitate the instantiation and management of the NRP on the topological elements selected by the NRP Policy topology filter.

It is also possible to distribute the NRP Policy to network devices using several mechanisms, including protocols such as NETCONF or RESTCONF, or exchanging it using a suitable routing protocol that network devices participate in (such as IGP(s) or BGP). The extensions to enable specific protocols to carry an NRP Policy definition will be described in separate documents.

#### 5.1.1. Network Resource Partition - Flow-Aggregate Selector

A router should be able to identify a packet belonging to a Slice-Flow Aggregate before it can apply the associated dataplane forwarding treatment or NRP-PHB. One or more fields within the packet are used as an FAS to do this.

Forwarding Address Based FAS:

It is possible to assign a different forwarding address (or MPLS forwarding label in case of MPLS network) for each Slice-Flow Aggregate on a specific node in the network. [RFC3031] states in Section 2.1 that: 'Some routers analyze a packet's network layer header not merely to choose the packet's next hop, but also to determine a packet's "precedence" or "class of service"'. Assigning a unique forwarding address (or MPLS forwarding label) to each Slice-Flow Aggregate allows Slice-Flow Aggregate packets destined to a node to be distinguished by the destination address (or MPLS forwarding label) that is carried in the packet.

This approach requires maintaining per Slice-Flow Aggregate state for each destination in the network in both the control and data plane and on each router in the network. For example, consider a network slicing provider with a network composed of 'N' nodes, each with 'K' adjacencies to its neighbors. Assuming a node can be reached over 'M' different Slice-Flow Aggregates, the node assigns and advertises reachability to 'N' unique forwarding addresses, or MPLS forwarding labels. Similarly, each node assigns a unique forwarding address (or MPLS forwarding label) for each of its 'K' adjacencies to enable strict steering over the adjacency for each slice. The total number of control and data plane states that need to be stored and programmed in a router's forwarding is  $(N+K)*M$  states. Hence, as 'N', 'K', and 'M' parameters increase, this approach suffers from scalability challenges in both the control and data planes.

#### Global Identifier Based FAS:

An NRP Policy may include a Global Identifier FAS (G-FAS) field that is carried in each packet in order to associate it to the NRP supporting a Slice-Flow Aggregate, independent of the forwarding address or MPLS forwarding label that is bound to the destination. Routers within the NRP domain can use the forwarding address (or MPLS forwarding label) to determine the forwarding next-hop(s), and use the G-FAS field in the packet to infer the specific forwarding treatment that needs to be applied on the packet.

The G-FAS can be carried in one of multiple fields within the packet, depending on the dataplane used. For example, in MPLS networks, the G-FAS can be encoded within an MPLS label that is carried in the packet's MPLS label stack. All packets that belong to the same Slice-Flow Aggregate may carry the same G-FAS in the MPLS label stack. It is also possible to have multiple G-FAS's map to the same Slice-Flow Aggregate.

The G-FAS can be encoded in an MPLS label and may appear in several positions in the MPLS label stack. For example, the VPN service label may act as a G-FAS to allow VPN packets to be mapped to the Slice-Flow Aggregate. In this case, a single VPN service label acting as a G-FAS may be allocated by all Egress PEs of a VPN. Alternatively, multiple VPN service labels may act as G-FAS's that map a single VPN to the same Slice-Flow Aggregate to allow for multiple Egress PEs to allocate different VPN service labels for a VPN. In other cases, a range of VPN service labels acting as multiple G-FAS's may map multiple VPN traffic to a single Slice-Flow Aggregate. An example of such deployment is shown in Figure 3.

SR Adj-SID:                    G-FAS (VPN service label) on PE2: 1001  
9012: P1-P2  
9023: P2-PE2

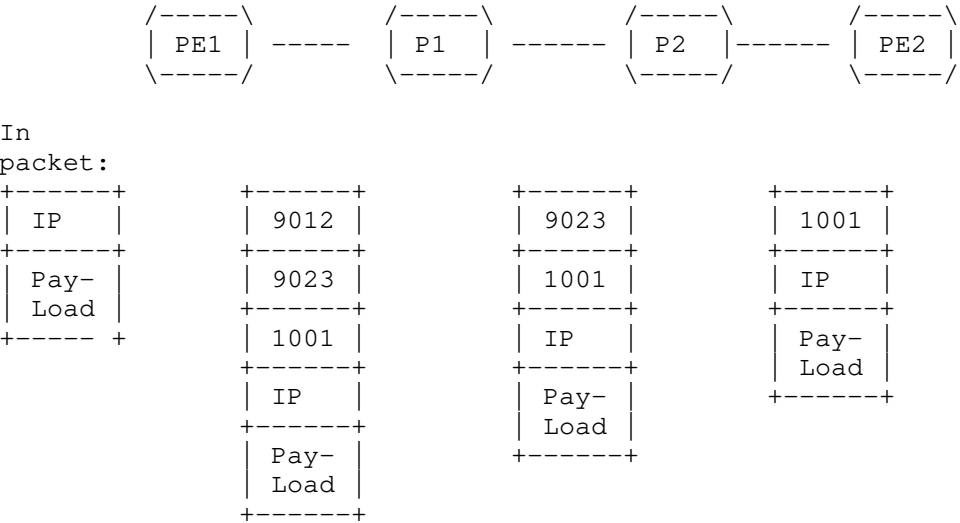


Figure 3: G-FAS or VPN label at bottom of label stack.

In some cases, the position of the G-FAS may not be at a fixed position in the MPLS label header. In this case, the G-FAS label can show up in any position in the MPLS label stack. To enable a transit router to identify the position of the G-FAS label, a special purpose label can be used to indicate the presence of a G-FAS in the MPLS label stack as shown in Figure 4.

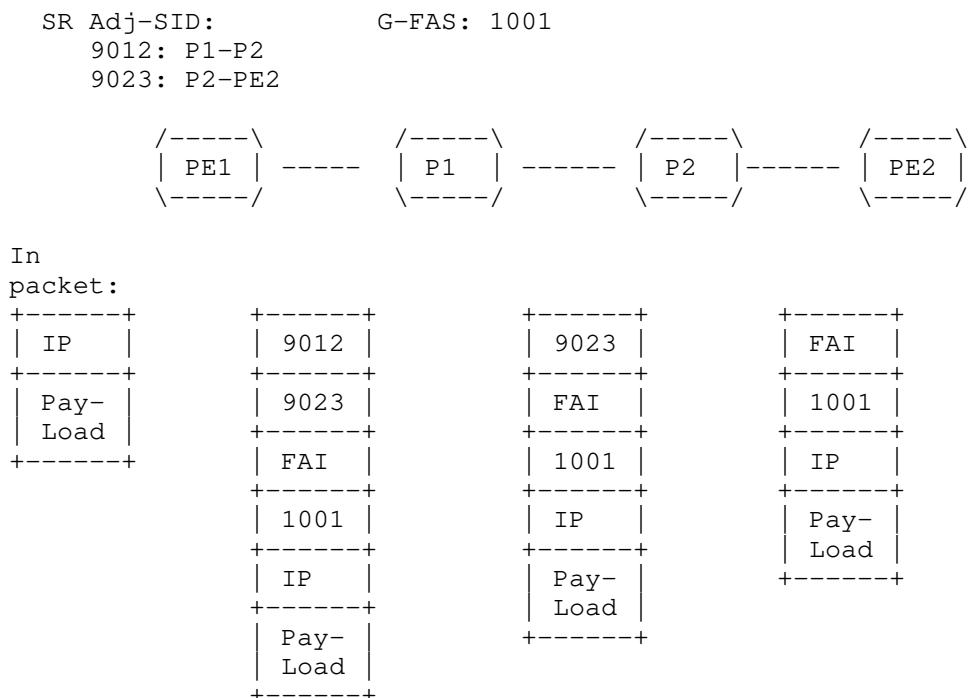


Figure 4: FAI and G-FAS label in the label stack.

When the slice is realized over an IP dataplane, the G-FAS can be encoded in the IP header (e.g. as an IPv6 option header).

#### 5.1.2. Network Resource Partition Resource Reservation

Bandwidth and network resource allocation strategies for slice policies are essential to achieve optimal placement of paths within the network while still meeting the target SLOs.

Resource reservation allows for the management of available bandwidth and the prioritization of existing allocations to enable preference-based preemption when contention on a specific network resource arises. Sharing of a network resource's available bandwidth amongst a group of NRPs may also be desirable. For example, a Slice-Flow Aggregate may not be using all of the NRP reservable bandwidth; this allows other NRPs in the same group to use the available bandwidth resources for other Slice-Flow Aggregates.



Congestion on shared network resources may result from sub-optimal placement of paths in different slice policies. When this occurs, preemption of some Slice-Flow Aggregate paths may be desirable to alleviate congestion. A preference-based allocation scheme enables prioritization of Slice-Flow Aggregate paths that can be preempted.

Since network characteristics and its state can change over time, the NRP topology and its network state need to be propagated in the network to enable ingress TE routers or Path Computation Engine (PCEs) to perform accurate path placement based on the current state of the NRP network resources.

#### 5.1.3. Network Resource Partition Per Hop Behavior

In Diffserv terminology, the forwarding behavior that is assigned to a specific class is called a Per Hop Behavior (PHB). The PHB defines the forwarding precedence that a marked packet with a specific CS receives in relation to other traffic on the Diffserv-aware network.

The NRP Per Hop Behavior (NRP-PHB) is the externally observable forwarding behavior applied to a specific packet belonging to a Slice-Flow Aggregate. The goal of an NRP-PHB is to provide a specified amount of network resources for traffic belonging to a specific Slice-Flow Aggregate. A single NRP may also support multiple forwarding treatments or services that can be carried over the same logical network.

The Slice-Flow Aggregate traffic may be identified at NRP ingress boundary nodes by carrying a FAS to allow routers to apply a specific forwarding treatment that guarantee the SLA(s).

With Differentiated Services (Diffserv) it is possible to carry multiple services over a single converged network. Packets requiring the same forwarding treatment are marked with a CS at domain ingress nodes. Up to eight classes or Behavior Aggregates (BAs) may be supported for a given Forwarding Equivalence Class (FEC) [RFC2475]. To support multiple forwarding treatments over the same Slice-Flow Aggregate, a Slice-Flow Aggregate packet may also carry a Diffserv CS to identify the specific Diffserv forwarding treatment to be applied on the traffic belonging to the same NRP.

At transit nodes, the CS field carried inside the packets are used to determine the specific PHB that determines the forwarding and scheduling treatment before packets are forwarded, and in some cases, drop probability for each packet.

#### 5.1.4. Network Resource Partition Topology

A key element of the NRP Policy is a customized topology that may include the full or subset of the physical network topology. The NRP topology could also span multiple administrative domains and/or multiple dataplane technologies.

An NRP topology can overlap or share a subset of links with another NRP topology. A number of topology filtering policies can be defined as part of the NRP Policy to limit the specific topology elements that belong to the NRP. For example, a topology filtering policy can leverage Resource Affinities as defined in [RFC2702] to include or exclude certain links that the NRP is instantiated on in supports of the Slice-Flow Aggregate.

The NRP Policy may also include a reference to a predefined topology (e.g., derived from a Flexible Algorithm Definition (FAD) as defined in [I-D.ietf-lsr-flex-algo], or Multi-Topology ID as defined [RFC4915]).

#### 5.2. Network Resource Partition Boundary

A network slice originates at the edge nodes of a network slice provider. Traffic that is steered over the corresponding NRP supporting a Slice-Flow Aggregate may traverse NRP capable as well as NRP incapable interior nodes.

The network slice may encompass one or more domains administered by a provider. For example, an organization's intranet or an ISP. The network provider is responsible for ensuring that adequate network resources are provisioned and/or reserved to support the SLAs offered by the network end-to-end.

##### 5.2.1. Network Resource Partition Edge Nodes

NRP edge nodes sit at the boundary of a network slice provider network and receive traffic that requires steering over network resources specific to a NRP that supports a Slice-Flow Aggregate. These edge nodes are responsible for identifying Slice-Flow Aggregate specific traffic flows by possibly inspecting multiple fields from inbound packets (e.g., implementations may inspect IP traffic's network 5-tuple in the IP and transport protocol headers) to decide on which NRP it can be steered.

Network slice ingress nodes may condition the inbound traffic at network boundaries in accordance with the requirements or rules of each service's SLAs. The requirements and rules for network slice services are set using mechanisms which are outside the scope of this document.

When data plane NRP mode is employed, the NRP ingress nodes are responsible for adding a suitable FAS onto packets that belong to specific Slice-Flow Aggregate. In addition, edge nodes may mark the corresponding Diffserv CS to differentiate between different types of traffic carried over the same Slice-Flow Aggregate.

#### 5.2.2. Network Resource Partition Interior Nodes

An NRP interior node receives slice traffic and may be able to identify the packets belonging to a specific Slice-Flow Aggregate by inspecting the FAS field carried inside each packet, or by inspecting other fields within the packet that may identify the traffic streams that belong to a specific Slice-Flow Aggregate. For example, when data plane NRP mode is applied, interior nodes can use the FAS carried within the packet to apply the corresponding NRP-PHB forwarding behavior. Nodes within the network slice provider network may also inspect the Diffserv CS within each packet to apply a per Diffserv class PHB within the NRP Policy, and allow differentiation of forwarding treatments for packets forwarded over the same NRP that supports the Slice-Flow Aggregate.

#### 5.2.3. Network Resource Partition Incapable Nodes

Packets that belong to a Slice-Flow Aggregate may need to traverse nodes that are NRP incapable. In this case, several options are possible to allow the slice traffic to continue to be forwarded over such devices and be able to resume the NRP forwarding treatment once the traffic reaches devices that are NRP-capable.

When data plane NRP mode is employed, packets carry a FAS to allow slice interior nodes to identify them. To support end-to-end network slicing, the FAS is maintained in the packets as they traverse devices within the network - including NRP capable and incapable devices.

For example, when the FAS is an MPLS label at the bottom of the MPLS label stack, packets can traverse over devices that are NRP incapable without any further considerations. On the other hand when the FASL is at the top of the MPLS label stack, packets can be bypassed (or tunneled) over the NRP incapable devices towards the next device that supports NRP as shown in Figure 5.

```
SR Node-SID:          FASL: 1001      @@@: NRP Policy enforced
1601: P1              ...: NRP Policy not enforced
1602: P2
1603: P3
1604: P4
1605: P5
```

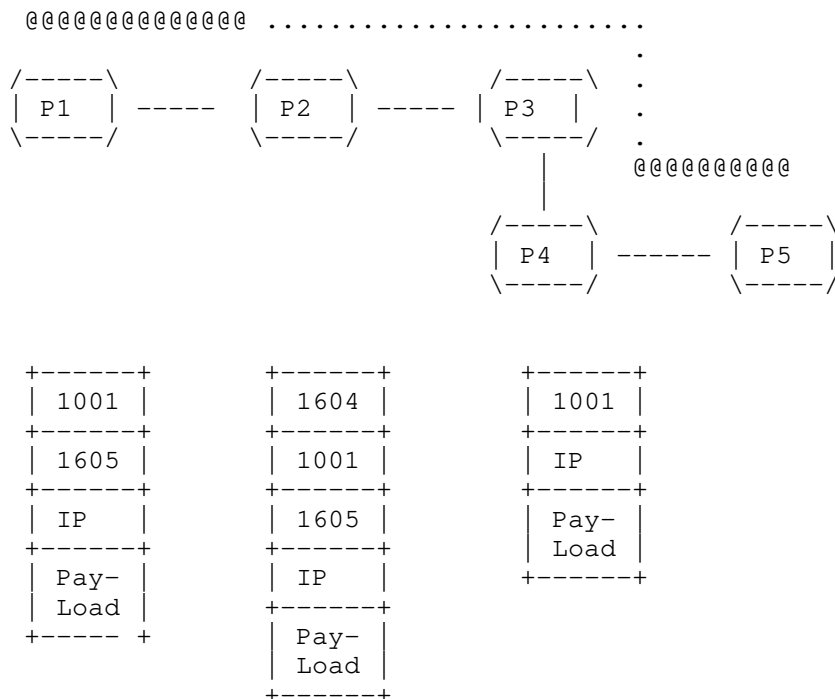


Figure 5: Extending network slice over NRP incapable device(s).

#### 5.2.4. Combining Network Resource Partition Modes

It is possible to employ a combination of the NRP modes that were discussed in Section 4 to realize a network slice. For example, data and control plane NRP modes can be employed in parts of a network, while control plane NRP mode can be employed in the other parts of the network. The path selection, in such case, can take into account the NRP available network resources. The FAS carried within packets allow transit nodes to enforce the corresponding NRP-PHB on the parts of the network that apply the data plane NRP mode. The FAS can be maintained while traffic traverses nodes that do not enforce data plane NRP mode, and so slice PHB enforcement can resume once traffic traverses capable nodes.

## 6. Mapping Traffic on Slice-Flow Aggregates

The usual techniques to steer traffic onto paths can be applicable when steering traffic over paths established for a specific Slice-Flow Aggregate.

For example, one or more (layer-2 or layer-3) VPN services can be directly mapped to paths established for a Slice-Flow Aggregate. In this case, the per Virtual Routing and Forwarding (VRF) instance traffic that arrives on the Provider Edge (PE) router over external interfaces can be directly mapped to a specific Slice-Flow Aggregate path. External interfaces can be further partitioned (e.g., using VLANs) to allow mapping one or more VLANs to specific Slice-Flow Aggregate paths.

Another option is steer traffic to specific destinations directly over multiple slice policies. This allows traffic arriving on any external interface and targeted to such destinations to be directly steered over the slice paths.

A third option that can also be used is to utilize a data plane firewall filter or classifier to enable matching of several fields in the incoming packets to decide whether the packet belongs to a specific Slice-Flow Aggregate. This option allows for applying a rich set of rules to identify specific packets to be mapped to a Slice-Flow Aggregate. However, it requires data plane network resources to be able to perform the additional checks in hardware.

### 6.1. Network Slice-Flow Aggregate Relationships

The following describes the generalization relationships between the IETF network slice and different parts of the solution as described in Figure 1.

- o A customer may request one or more IETF Network Slices.
- o Any given Attachment Circuit (AC) may support the traffic for one or more IETF Network Slices. If there is more than one IETF Network Slice using a single AC, the IETF Network Slice Service request must include enough information to allow the edge nodes to demultiplex the traffic for the different IETF Network Slices.
- o By definition, multiple IETF Network Slices may be mapped to a single Slice-Flow Aggregate. However, it is possible for an Slice-Flow Aggregate to contain just a single IETF Network Slice.

- o The physical network may be filtered to multiple Filter Topologies. Each such Filter Topology facilitates planning the placement of paths for the Slice-Flow Aggregate by presenting only the subset of links and nodes that meet specific criteria. Note, however, in absence of any Filter Topology, Slice-Flow Aggregate are free to operate over the full physical network.

- o It is anticipated that there may be very many IETF Network Slices supported by a network operator over a single physical network. A network may support a limited number of Slice-Flow Aggregates, with each of the Slice-Flow Aggregates grouping any number of the IETF Network Slices streams.

## 7. Path Selection and Instantiation

### 7.1. Applicability of Path Selection to Slice-Flow Aggregates

In State-dependent TE [I-D.ietf-teas-rfc3272bis], the path selection adapts based on the current state of the network. The state of the network can be based on parameters flooded by the routers as described in [RFC2702]. The link state is advertised with current reservations, thereby reflecting the available bandwidth on each link. Such link reservations may be maintained centrally on a network wide network resource manager, or distributed on devices (as usually done with RSVP-TE). TE extensions exist today to allow IGPs (e.g., [RFC3630] and [RFC5305]), and BGP-LS [RFC7752] to advertise such link state reservations.

When the network resource reservations are maintained for NRPs, the link state can carry per NRP state (e.g., reservable bandwidth). This allows path computation to take into account the specific network resources available for an NRP. In this case, we refer to the process of path placement and path provisioning as NRP aware TE (NRP-TE).

### 7.2. Applicability of Path Control Technologies to Slice-Flow Aggregates

The NRP modes described in this document are agnostic to the technology used to setup paths that carry Slice-Flow Aggregate traffic. One or more paths connecting the endpoints of the mapped IETF network slices may be selected to steer the corresponding traffic streams over the resources allocated for the NRP that supports a Slice-Flow Aggregate.

The feasible paths can be computed using the NRP topology and network state subject the optimization metrics and constraints.

### 7.2.1. RSVP-TE Based Slice-Flow Aggregate Paths

RSVP-TE [RFC3209] can be used to signal LSPs over the computed feasible paths in order to carry the Slice-Flow Aggregate traffic. The specific extensions to the RSVP-TE protocol required to enable signaling of NRP aware RSVP-TE LSPs are outside the scope of this document.

### 7.2.2. SR Based Slice-Flow Aggregate Paths

Segment Routing (SR) [RFC8402] can be used to setup and steer traffic over the computed Slice-Flow Aggregate feasible paths.

The SR architecture defines a number of building blocks that can be leveraged to support the realization of NRPs that support Slice-Flow Aggregates in an SR network.

Such building blocks include:

- \* SR Policy with or without Flexible Algorithm.
- \* Steering of services (e.g. VPN) traffic over SR paths
- \* SR Operation, Administration and Management (OAM) and Performance Management (PM)

SR allows a headend node to steer packets onto specific SR paths using a Segment Routing Policy (SR Policy). The SR policy supports various optimization objectives and constraints and can be used to steer Slice-Flow Aggregate traffic in the SR network.

The SR policy can be instantiated with or without the IGP Flexible Algorithm (Flex-Algorithm) feature. It may be possible to dedicate a single SR Flex-Algorithm to compute and instantiate SR paths for one Slice-Flow Aggregate traffic. In this case, the SR Flex-Algorithm computed paths and Flex-Algorithm SR SIDs are not shared by other Slice-Flow Aggregates traffic. However, to allow for better scale, it may be desirable for multiple Slice-Flow Aggregates traffic to share the same SR Flex-Algorithm computed paths and SIDs.

## 8. Network Resource Partition Protocol Extensions

Routing protocols may need to be extended to carry additional per NRP link state. For example, [RFC5305], [RFC3630], and [RFC7752] are ISIS, OSPF, and BGP protocol extensions to exchange network link state information to allow ingress TE routers and PCE(s) to do proper path placement in the network. The extensions required to support network slicing may be defined in other documents, and are outside

the scope of this document.

The instantiation of an NRP Policy may need to be automated. Multiple options are possible to facilitate automation of distribution of an NRP Policy to capable devices.

For example, a YANG data model for the NRP Policy may be supported on network devices and controllers. A suitable transport (e.g., NETCONF [RFC6241], RESTCONF [RFC8040], or gRPC) may be used to enable configuration and retrieval of state information for slice policies on network devices. The NRP Policy YANG data model is outside the scope of this document.

## 9. Outstanding Issues

Note to RFC Editor: Please remove this section prior to publication.

This section records non-blocking issues that were raised during the Working Group Adoption Poll for the document. The below list of issues needs to be fully addressed before progressing the document to publication in IESG.

1. Add new Appendix section with examples for the NRP modes described in Section 4.
2. Add text to clarify the relationship between Slice-Flow Aggregates, the NRP Policy, and the NRP.
3. Remove redundant references to Diffserv behaviors.
4. Elaborate on the SFA packet treatment when no rules to associate the packet to an NRP are defined in the NRP Policy.
5. Clarify the NRP instantiation through the NRP Policy enforcement.
6. Clarify how the solution caters to the different IETF Network Slice Service Demarcation Point locations described in Section 4.2 of [I-D.ietf-teas-ietf-network-slices].
7. Clarify the relationship the underlay physical network, the filter topology and the NRP resources.
8. Expand on how isolation between NRPs can be realized depending on the deployed NRP mode.
9. Revise Section 5.2.3 to describe how nodes can discover NRP incapable downstream neighbors.



10. Expand Section 11 on additional security threats introduced with the solution.
11. Expand Section 5.2 on NRP domain boundary and multi-domain aspects.

## 10. IANA Considerations

This document has no IANA actions.

## 11. Security Considerations

The main goal of network slicing is to allow for varying treatment of traffic from multiple different network slices that are utilizing a common network infrastructure and to allow for different levels of services to be provided for traffic traversing a given network resource.

A variety of techniques may be used to achieve this, but the end result will be that some packets may be mapped to specific resources and may receive different (e.g., better) service treatment than others. The mapping of network traffic to a specific NRP is indicated primarily by the FAS, and hence an adversary may be able to utilize resources allocated to a specific NRP by injecting packets carrying the same FAS field in their packets.

Such theft-of-service may become a denial-of-service attack when the modified or injected traffic depletes the resources available to forward legitimate traffic belonging to a specific NRP.

The defense against this type of theft and denial-of-service attacks consists of a combination of traffic conditioning at NRP domain boundaries with security and integrity of the network infrastructure within an NRP domain.

## 12. Acknowledgement

The authors would like to thank Krzysztof Szarkowicz, Swamy SRK, Navaneetha Krishnan, Prabhu Raj Villadathu Karunakaran, and Mohamed Boucadair for their review of this document and for providing valuable feedback on it. The authors would also like to thank Adrian Farrel for detailed discussions that resulted in Section 3.

## 13. Contributors

The following individuals contributed to this document:

Colby Barth  
Juniper Networks  
Email: cbarth@juniper.net

Srihari R. Sangli  
Juniper Networks  
Email: ssangli@juniper.net

Chandra Ramachandran  
Juniper Networks  
Email: csekar@juniper.net

Adrian Farrel  
Old Dog Consulting  
United Kingdom  
Email: adrian@olddog.co.uk

## 14. References

### 14.1. Normative References

- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.

### 14.2. Informative References

- [I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", Work in Progress, Internet-Draft, draft-ietf-lsr-flex-algo-19, 7 April 2022, <<https://www.ietf.org/archive/id/draft-ietf-lsr-flex-algo-19.txt>>.
- [I-D.ietf-teas-ietf-network-slices]  
Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-10, 27 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-10.txt>>.
- [I-D.ietf-teas-rfc3272bis]  
Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-16, 24 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-rfc3272bis-16.txt>>.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <<https://www.rfc-editor.org/info/rfc2475>>.
- [RFC2702] Awduche, D., Malcolm, J., Agoghua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <<https://www.rfc-editor.org/info/rfc2702>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <<https://www.rfc-editor.org/info/rfc4915>>.
- [RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <<https://www.rfc-editor.org/info/rfc5462>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

## Authors' Addresses

Tarek Saad  
Juniper Networks  
Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Vishnu Pavan Beeram  
Juniper Networks  
Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

Jie Dong  
Huawei Technologies  
Email: [jie.dong@huawei.com](mailto:jie.dong@huawei.com)

Bin Wen  
Comcast  
Email: [Bin\\_Wen@cable.comcast.com](mailto:Bin_Wen@cable.comcast.com)

Daniele Ceccarelli  
Ericsson  
Email: [daniele.ceccarelli@ericsson.com](mailto:daniele.ceccarelli@ericsson.com)

Joel Halpern  
Ericsson  
Email: [joel.halpern@ericsson.com](mailto:joel.halpern@ericsson.com)

Shaofu Peng  
ZTE Corporation

Email: peng.shaofu@zte.com.cn

Ran Chen  
ZTE Corporation  
Email: chen.ran@zte.com.cn

Xufeng Liu  
Volta Networks  
Email: xufeng.liu.ietf@gmail.com

Luis M. Contreras  
Telefonica  
Email: luismiguel.contrerasmurillo@telefonica.com

Reza Rokui  
Ciena  
Email: rrokui@ciena.com

Luay Jalil  
Verizon  
Email: luay.jalil@verizon.com

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 13 January 2022

I. Busi  
Huawei  
X. Liu  
Volta Networks  
I. Bryskin  
Individual  
V. Beeram  
T. Saad  
Juniper Networks  
O. Gonzalez de Dios  
Telefonica  
12 July 2021

Profiles for Traffic Engineering (TE) Topology Data Model  
draft-busi-teas-te-topology-profiles-02

Abstract

This document describes how profiles of the Traffic Engineering (TE) Topology Model, defined in RFC8795, can be used to address applications beyond "Traffic Engineering".

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 13 January 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                                       | 2  |
| 2. Examples of non-TE scenarios . . . . .                       | 3  |
| 2.1. UNI Topology Discovery . . . . .                           | 3  |
| 2.2. Administrative and Operational status management . . . . . | 5  |
| 2.3. Geolocation . . . . .                                      | 6  |
| 2.4. Overlay and Underlay non-TE Topologies . . . . .           | 7  |
| 2.5. Nodes with switching limitations . . . . .                 | 8  |
| 3. Technology-specific augmentations . . . . .                  | 9  |
| 3.1. Multi-inheritance . . . . .                                | 11 |
| 3.2. Example (Link augmentation) . . . . .                      | 12 |
| 4. Implemented profiles . . . . .                               | 13 |
| 5. Security Considerations . . . . .                            | 14 |
| 6. IANA Considerations . . . . .                                | 14 |
| 7. References . . . . .   | 14 |
| 7.1. Normative References . . . . .                             | 14 |
| 7.2. Informative References . . . . .                           | 14 |
| Contributors . . . . .  | 15 |
| Authors' Addresses . . . . .                                    | 16 |

## 1. Introduction

There are many network scenarios being discussed in various IETF Working Groups (WGs) that are not classified as "Traffic Engineering" but can be addressed by a sub-set (profile) of the Traffic Engineering (TE) Topology YANG data model, defined in [RFC8795].

Traffic Engineering (TE) is defined in [I-D.ietf-teas-rfc3272bis] as aspects of Internet network engineering that deal with the issues of performance evaluation and performance optimization of operational IP networks. TE encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic.

The TE Topology Model is augmenting the Network Topology Model defined in [RFC8345] with generic and technology-agnostic features that some are strictly applicable to TE networks, while others applicable to both TE and non-TE networks.

Examples of such features that are applicable to both TE and non-TE networks are: inter-domain link discovery (plug-id), geo-localization, and admin/operational status.

It is also worth noting that the TE Topology Model is quite an extensive and comprehensive model in which most features are optional. Therefore, even though the full model appears to be complex, at the first glance, a sub-set of the model (profile) can be used to address specific scenarios, e.g. suitable also to non-TE use cases.

The implementation of such TE Topology profiles can simplify and expedite adoption of the full TE topology YANG data model, and allow for its reuse even for non-TE use case. The key question being whether all or some of the attributes defined in the TE Topology Model are needed to address a given network scenario.

Section 2 provides examples where profiles of the TE Topology Model can be used to address some generic use cases applicable to both TE and non-TE technologies.

## 2. Examples of non-TE scenarios

### 2.1. UNI Topology Discovery

UNI Topology Discovery is independent from whether the network is TE or non-TE.

The TE Topology Model supports inter-domain link discovery (including but not being limited to UNI link discovery) using the plug-id attribute. This solution is quite generic and does not require the network to be a TE network.

The following profile of the TE Topology model can be used for the UNI Topology Discovery:

```
module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +---rw te-topology!
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +---rw te-tp-id?   te-types:te-tp-id
    +---rw te!
      +---rw admin-status?
      |       te-types:te-admin-status
    +---rw inter-domain-plug-id?          binary
    +---ro oper-status?                   te-types:te-oper-status
```

Figure 1: UNI Topology



The profile data model shown in Figure 1 can be used to discover TE and non TE UNIs as well as to discover UNIs for TE or non TE networks.

Such a UNI TE Topology profile model can also be used with technology-specific UNI augmentations, as described in section 3.

For example, in [I-D.ietf-ccamp-eth-client-te-topo-yang], the eth-svc container is defined to represent the capabilities of the Termination Point (TP) to be configured as an Ethernet client UNI, together with the Ethernet classification and VLAN operations supported by that TP.

The [I-D.ietf-ccamp-otn-topo-yang] provides another example, where:

- \* the client-svc container is defined to represent the capabilities of the TP to be configured as an transparent client UNI (e.g., STM-N, Fiber Channel or transparent Ethernet);
- \* the OTN technology-specific Link Termination Point (LTP) augmentations are defined to represent the capabilities of the TP to be configured as an OTN UNI, together with the information about OTN label and bandwidth availability at the OTN UNI.

For example, the UNI TE Topology profile can be used to model features defined in [I-D.ogondio-opsawg-uni-topology]:

- \* The inter-domain-plug-id attribute would provide the same information as the attachment-id attribute defined in [I-D.ogondio-opsawg-uni-topology];
- \* The admin-status and oper-status that exists in this TE topology profile can provide the same information as the admin-status and oper-status attributes defined in [I-D.ogondio-opsawg-uni-topology].

Following the same approach in [I-D.ietf-ccamp-eth-client-te-topo-yang] and [I-D.ietf-ccamp-otn-topo-yang], the type and encapsulation-type attributes can be defined by technology-specific UNI augmentations to represent the capability of a TP to be configured as a L2VPN/L3VPN UNI Service Attachment Point (SAP).

The advantages of using a TE Topology profile would be having common solutions for:

- \* discovering UNIs as well as inter-domain NNI links, which is applicable to any technology (TE or non TE) used at the UNI or within the network;

- \* modelling non TE UNIs such as Ethernet, and TE UNIs such as OTN, as well as UNIs which can be configured as TE or non-TE (e.g., being configured as either Ethernet or OTN UNI).

## 2.2. Administrative and Operational status management

The TE Topology Model supports the management of administrative and operational state, including also the possibility to associate some administrative names, for nodes, termination points and links. This solution is generic and also does not require the network to be a TE network.

The following profile of the TE Topology Model can be used for administrative and operational state management:

```

module: ietf-te-topology
augment /nw:networks/nw:network/nw:network-types:
  +--rw te-topology!
augment /nw:networks/nw:network:
  +--rw te-topology-identifier
  |   +--rw provider-id?    te-global-id
  |   +--rw client-id?     te-global-id
  |   +--rw topology-id?   te-topology-id
  +--rw te!
  |   +--rw name?          string
augment /nw:networks/nw:network/nw:node:
  +--rw te-node-id?    te-types:te-node-id
  +--rw te!
  |   +--rw te-node-attributes
  |   |   +--rw admin-status?    te-types:te-admin-status
  |   |   +--rw name?           string
  |   +--ro oper-status?        te-types:te-oper-status
augment /nw:networks/nw:network/nt:link:
  +--rw te!
  |   +--rw te-link-attributes
  |   |   +--rw name?           string
  |   |   +--rw admin-status?   te-types:te-admin-status
  |   +--ro oper-status?        te-types:te-oper-status
augment /nw:networks/nw:network/nw:node/nt:termination-point:
  +--rw te-tp-id?    te-types:te-tp-id
  +--rw te!
  |   +--rw admin-status?    te-types:te-admin-status
  |   +--rw name?           string
  |   +--ro oper-status?    te-types:te-oper-status

```

Figure 2: Generic Topology with admin and operational state

The TE topology data model profile shown in Figure 2 is applicable to any technology (TE or non-TE) that requires management of the administrative and operational state and administrative names for nodes, termination points and links.

### 2.3. Geolocation

The TE Topology model supports the management of geolocation coordinates for nodes and termination points. This solution is generic and does not necessarily require the network to be a TE network.

The TE topology data model profile shown in Figure 3 can be used to model geolocation data for networks.

```

module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw te-topology!
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +--rw te-tp-id?   te-types:te-tp-id
    +--rw te!
      +--ro geolocation
        +--ro altitude?   int64
        +--ro latitude?   geographic-coordinate-degree
        +--ro longitude?  geographic-coordinate-degree
  augment /nw:networks/nw:network/nw:node:
    +--rw te-node-id?   te-types:te-node-id
    +--rw te!
      +--ro geolocation
        +--ro altitude?   int64
        +--ro latitude?   geographic-coordinate-degree
        +--ro longitude?  geographic-coordinate-degree
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +--rw te-tp-id?   te-types:te-tp-id
    +--rw te!
      +--ro geolocation
        +--ro altitude?   int64
        +--ro latitude?   geographic-coordinate-degree
        +--ro longitude?  geographic-coordinate-degree

```

Figure 3: Generic Topology with geolocation information

This profile is applicable to any network technology (TE or non-TE) that requires management of the geolocation information for its nodes and termination points.

## 2.4. Overlay and Underlay non-TE Topologies

The TE Topology model supports the management of overlay/underlay relationship for nodes and links, as described in section 5.8 of [RFC8795]. This solution is generic and does not require the network to be a TE network.

The following TE topology data model profile can be used to manage overlay/underlay network data:

```

module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw te-topology!
  augment /nw:networks/nw:network/nw:node:
    +---rw te-node-id?    te-types:te-node-id
    +---rw te!
    +---rw te-node-attributes
    +---rw underlay-topology {te-topology-hierarchy}?
    +---rw network-ref?    -> /nw:networks/network/network-id
  augment /nw:networks/nw:network/nt:link:
    +---rw te!
    +---rw te-link-attributes
    +---rw underlay {te-topology-hierarchy}?
    +---rw enabled?                boolean
    +---rw primary-path
    +---rw network-ref?
    |       -> /nw:networks/network/network-id
    +---rw path-element* [path-element-id]
    +---rw path-element-id                uint32
    +---rw (type)?
    +---:(numbered-link-hop)
    |   +---rw numbered-link-hop
    |   |   +---rw link-tp-id    te-tp-id
    |   |   +---rw hop-type?    te-hop-type
    |   |   +---rw direction?   te-link-direction
    +---:(unnumbered-link-hop)
    +---rw unnumbered-link-hop
    +---rw link-tp-id    te-tp-id
    +---rw node-id      te-node-id
    +---rw hop-type?    te-hop-type
    +---rw direction?   te-link-direction

```

Figure 4: Generic Topology with overlay/underlay information

This profile is applicable to any technology (TE or non-TE) when it is needed to manage the overlay/underlay information. It is also allows a TE underlay network to support a non-TE overlay network and, vice versa, a non-TE underlay network to support a TE overlay network.

## 2.5. Nodes with switching limitations

A node can have some switching limitations where connectivity is not possible between all its TP pairs, for example when:

- \* the node represents a physical device with switching limitations;
- \* the node represents an abstraction of a network topology.

This scenario is generic and applies to both TE and non-TE technologies.

A connectivity TE Topology profile data model supports the management of the node connectivity matrix to represent feasible connections between termination points across the nodes. This solution is generic and does not necessarily require a TE enabled network.

The following profile of the TE Topology model can be used for nodes with connectivity constraints:

```

module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw te-topology!
  augment /nw:networks/nw:network/nw:node:
    +--rw te-node-id?    te-types:te-node-id
    +--rw te!
      +--rw te-node-attributes
        +--rw connectivity-matrices
          +--rw number-of-entries?    uint16
          +--rw is-allowed?           boolean
          +--rw connectivity-matrix* [id]
            +--rw id                  uint32
            +--rw from
              | +--rw tp-ref?         leafref
            +--rw to
              | +--rw tp-ref?         leafref
            +--rw is-allowed?         boolean

```

Figure 5: Generic Topology with connectivity constraints

The TE topology data model profile shown in Figure 5 is applicable to any technology (TE or non-TE) networks that requires managing nodes with certain connectivity constraints. When used with TE technologies, additional TE attributes, as defined in [RFC8795], can also be provided.

### 3. Technology-specific augmentations

There are two main options to define technology-specific Topology Models which can use the attributes defined in the TE Topology Model [RFC8795].

Both options are applicable to any possible profile of the TE Topology Model, such as those defined in Section 2.

The first option is to define a technology-specific TE Topology Model which augments the TE Topology Model, as shown in Figure 6:

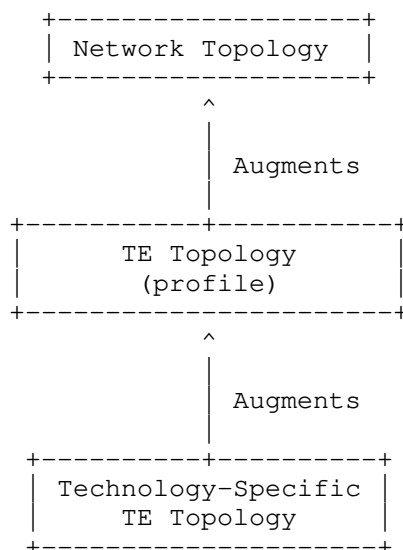


Figure 6: Augmenting the TE Topology Model

This approach is more suitable for cases when the technology-specific TE topology model provides augmentations to the TE Topology constructs, such as bandwidth information (e.g., link bandwidth), tunnel termination points (TTPs) or connectivity matrices. It also allows providing augmentations to the Network Topology constructs, such as nodes, links, and termination points (TPs).

This is the approach currently used in [I-D.ietf-ccamp-eth-client-te-topo-yang] and [I-D.ietf-ccamp-otn-topo-yang].

It is worth noting that a profile of the technology-specific TE Topology model not using any TE topology attribute or constructs can be used to address any use case that do not require these attributes. In this case, only the te-topology presence container of the TE Topology Model needs to be implemented.

The second option is to define a technology-specific Network Topology Model which augments the Network Topology Model and to rely on the multiple inheritance capability, which is implicit in the network-types definition of [RFC8345], to allow using also the generic attributes defined in the TE Topology model:

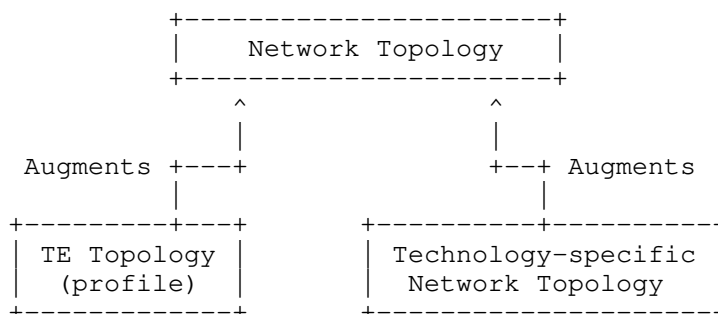


Figure 7: Augmenting the Network Topology Model with multi-inheritance

This approach is more suitable in cases where the technology-specific Network Topology Model provides augmentation only to the constructs defined in the Network Topology Model, such as nodes, links, and termination points (TPs). Therefore, with this approach, only the generic attributes defined in the TE Topology Model could be used.

It is also worth noting that in this case, technology-specific augmentations for the bandwidth information could not be defined.

In principle, it would be also possible to define both a technology specific TE Topology Model which augments the TE Topology Model, and a technology-specific Network Topology Model which augments the Network Topology Model and to rely on the multiple inheritance capability, as shown in Figure 8:

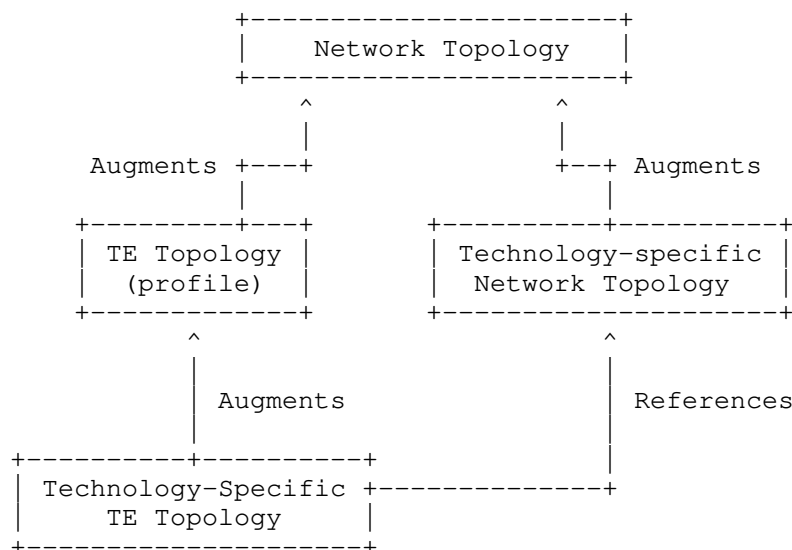


Figure 8: Augmenting both the Network and TE Topology Models

This option does not provide any technical advantage with respect to the first option, shown in Figure 6, but could be useful to add augmentations to the TE Topology constructs and to re-use an already existing technology-specific Network Topology Model.

It is worth noting that the technology-specific TE Topology model can reference constructs defined by the technology-specific Network Topology model but it could not augment constructs defined by the technology-specific Network Topology model.

### 3.1. Multi-inheritance

As described in section 4.1 of [RFC8345], the network types should be defined using presence containers to allow the representation of network subtypes.

The hierarchy of network subtypes can be single hierarchy, as shown in Figure 6. In this case, each presence container contains at most one child presence container, as shows in the JSON code below:



```
{
  "ietf-network:ietf-network": {
    "ietf-te-topology:te-topology": {
      "example-te-topology": {}
    }
  }
}
```

The hierarchy of network subtypes can also be multi-hierarchy, as shown in Figure 7 and Figure 8. In this case, one presence container can contain more than one child presence containers, as show in the JSON codes below:

```
{
  "ietf-network:ietf-network": {
    "ietf-te-topology:te-topology": {}
    "example-network-topology": {}
  }
}

{
  "ietf-network:ietf-network": {
    "ietf-te-topology:te-topology": {
      "example-te-topology": {}
    }
    "example-network-topology": {}
  }
}
```

Other examples of multi-hierarchy topologies are described in [I-D.ietf-teas-yang-sr-te-topo].

### 3.2. Example (Link augmentation)

This section provides an example on how technology-specific attributes can be added to the Link construct:

```

+--rw link* [link-id]
+--rw link-id          link-id
+--rw source
|   +--rw source-node?  -> ../../../../nw:node/node-id
|   +--rw source-tp?    leafref
+--rw destination
|   +--rw dest-node?    -> ../../../../nw:node/node-id
|   +--rw dest-tp?      leafref
+--rw supporting-link* [network-ref link-ref]
|   +--rw network-ref
|   |   -> ../../../../nw:supporting-network/network-ref
|   +--rw link-ref      leafref
+--rw example-link-attributes
|   <...>
+--rw te!
+--rw te-link-attributes
+--rw name?                                string
+--rw example-te-link-attributes
|   <...>
+--rw max-link-bandwidth
+--rw te-bandwidth
+--rw (technology)?
+--:(generic)
|   +--rw generic?    te-bandwidth
+--:(example)
|   +--rw example?    example-bandwidth

```

Figure 9: Augmenting the Link with technology-specific attributes

The technology-specific attributes within the example-link-attributes container can be defined either in the technology-specific TE Topology Model (Option 1) or in the technology-specific Network Topology Model (Option 2 or Option 3). These attributes can only be non-TE and do not require the implementation of the te container.

The technology-specific attributes within the example-te-link-attributes container as well as the example max-link-bandwidth can only be defined in the technology-specific TE Topology Model (Option 1 or Option 3). These attributes can be TE or non-TE and require the implementation of the te container.

#### 4. Implemented profiles

When a server implements a profile of the TE topology model, it is not clear how the server can report to the client the subset of the model being implemented.

It is also worth noting that the supported profile may also depend on other attributes (for example the network type).

In case the TE topology profile is reported by the server to the client, the server will report in the operational datastore only the leaves which have been implemented, as described in section 5.3 of [RFC8342].

More investigation is required in case the TE topology profile is configured by the client.

## 5. Security Considerations

This document provides only information about how the TE Topology Model, as defined in [RFC8795], can be profiled to address some scenarios which are not considered as TE.

As such, this document does not introduce any additional security considerations besides those already defined in [RFC8795].

## 6. IANA Considerations

This document requires no IANA actions.

## 7. References

### 7.1. Normative References

- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.

### 7.2. Informative References

[I-D.ietf-ccamp-eth-client-te-topo-yang]

Zheng, H., Guo, A., Busi, I., Xu, Y., Zhao, Y., and X. Liu, "A YANG Data Model for Ethernet TE Topology", Work in Progress, Internet-Draft, draft-ietf-ccamp-eth-client-te-topo-yang-00, 9 March 2021, <<https://www.ietf.org/archive/id/draft-ietf-ccamp-eth-client-te-topo-yang-00.txt>>.

[I-D.ietf-ccamp-otn-topo-yang]

Zheng, H., Busi, I., Liu, X., Belotti, S., and O. G. D. Dios, "A YANG Data Model for Optical Transport Network Topology", Work in Progress, Internet-Draft, draft-ietf-ccamp-otn-topo-yang-13, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-ccamp-otn-topo-yang-13.txt>>.

[I-D.ietf-teas-rfc3272bis]

Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-12, 15 May 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-rfc3272bis-12.txt>>.

[I-D.ietf-teas-yang-sr-te-topo]

Liu, X., Bryskin, I., Beeram, V. P., Saad, T., Shah, H., and S. Litkowski, "YANG Data Model for SR and SR TE Topologies on MPLS Data Plane", Work in Progress, Internet-Draft, draft-ietf-teas-yang-sr-te-topo-10, 6 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-yang-sr-te-topo-10.txt>>.

[I-D.ogondio-opsawg-uni-topology]

Dios, O. G. D., Barguil, S., Wu, Q., and M. Boucadair, "A YANG Model for User-Network Interface (UNI) Topologies", Work in Progress, Internet-Draft, draft-ogondio-opsawg-uni-topology-01, 2 April 2020, <<https://www.ietf.org/archive/id/draft-ogondio-opsawg-uni-topology-01.txt>>.

Contributors

Aihua Guo  
Futurewei Inc.

Email: [aihuaguo.ietf@gmail.com](mailto:aihuaguo.ietf@gmail.com)

Haomian Zheng  
Huawei

Email: zhenghaomian@huawei.com

Sergio Belotti  
Nokia

Email: sergio.belotti@nokia.com

#### Authors' Addresses

Italo Busi  
Huawei

Email: italo.busi@huawei.com

Xufeng Liu  
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Igor Bryskin  
Individual

Email: i\_bryskin@yahoo.com

Vishnu Pavan Beeram  
Juniper Networks

Email: vbeeram@juniper.net

Tarek Saad  
Juniper Networks

Email: tsaad@juniper.net

Oscar Gonzalez de Dios  
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 8 August 2022

I. Busi  
Huawei  
X. Liu  
Volta Networks  
I. Bryskin  
Individual  
T. Saad  
Juniper Networks  
O. Gonzalez de Dios  
Telefonica  
4 February 2022

Profiles for Traffic Engineering (TE) Topology Data Model and  
Applicability to non-TE Use Cases  
draft-busi-teas-te-topology-profiles-03

Abstract

This document describes how profiles of the Traffic Engineering (TE) Topology Model, defined in RFC8795, can be used to address applications beyond "Traffic Engineering".

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                                       | 2  |
| 2. Examples of non-TE scenarios . . . . .                       | 3  |
| 2.1. UNI Topology Discovery . . . . .                           | 3  |
| 2.2. Administrative and Operational status management . . . . . | 5  |
| 2.3. Geolocation . . . . .                                      | 6  |
| 2.4. Overlay and Underlay non-TE Topologies . . . . .           | 7  |
| 2.5. Nodes with switching limitations . . . . .                 | 8  |
| 3. Technology-specific augmentations . . . . .                  | 9  |
| 3.1. Multi-inheritance . . . . .                                | 11 |
| 3.2. Example (Link augmentation) . . . . .                      | 12 |
| 4. Implemented profiles . . . . .                               | 13 |
| 5. Security Considerations . . . . .                            | 14 |
| 6. IANA Considerations . . . . .                                | 14 |
| Acknowledgments . . . . .                                       | 14 |
| References . . . . .  | 14 |
| Normative References . . . . .                                  | 14 |
| Informative References . . . . .                                | 15 |
| Contributors . . . . .  | 16 |
| Authors' Addresses . . . . .                                    | 16 |

## 1. Introduction

There are many network scenarios being discussed in various IETF Working Groups (WGs) that are not classified as "Traffic Engineering" but can be addressed by a sub-set (profile) of the Traffic Engineering (TE) Topology YANG data model, defined in [RFC8795].

Traffic Engineering (TE) is defined in [I-D.ietf-teas-rfc3272bis] as aspects of Internet network engineering that deal with the issues of performance evaluation and performance optimization of operational IP networks. TE encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic.

The TE Topology Model is augmenting the Network Topology Model defined in [RFC8345] with generic and technology-agnostic features that some are strictly applicable to TE networks, while others applicable to both TE and non-TE networks.

Examples of such features that are applicable to both TE and non-TE networks are: inter-domain link discovery (plug-id), geo-localization, and admin/operational status.

It is also worth noting that the TE Topology Model is quite an extensive and comprehensive model in which most features are optional. Therefore, even though the full model appears to be complex, at the first glance, a sub-set of the model (profile) can be used to address specific scenarios, e.g. suitable also to non-TE use cases.

The implementation of such TE Topology profiles can simplify and expedite adoption of the full TE topology YANG data model, and allow for its reuse even for non-TE use case. The key question being whether all or some of the attributes defined in the TE Topology Model are needed to address a given network scenario.

Section 2 provides examples where profiles of the TE Topology Model can be used to address some generic use cases applicable to both TE and non-TE technologies.

## 2. Examples of non-TE scenarios

### 2.1. UNI Topology Discovery

UNI Topology Discovery is independent from whether the network is TE or non-TE.

The TE Topology Model supports inter-domain link discovery (including but not being limited to UNI link discovery) using the plug-id attribute. This solution is quite generic and does not require the network to be a TE network.

The following profile of the TE Topology model can be used for the UNI Topology Discovery:

```
module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +---rw te-topology!
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +---rw te-tp-id?   te-types:te-tp-id
    +---rw te!
      +---rw admin-status?
      |       te-types:te-admin-status
    +---rw inter-domain-plug-id?          binary
    +---ro oper-status?                   te-types:te-oper-status
```

Figure 1: UNI Topology



The profile data model shown in Figure 1 can be used to discover TE and non TE UNIs as well as to discover UNIs for TE or non TE networks.

Such a UNI TE Topology profile model can also be used with technology-specific UNI augmentations, as described in section 3.

For example, in [I-D.ietf-ccamp-eth-client-te-topo-yang], the eth-svc container is defined to represent the capabilities of the Termination Point (TP) to be configured as an Ethernet client UNI, together with the Ethernet classification and VLAN operations supported by that TP.

The [I-D.ietf-ccamp-otn-topo-yang] provides another example, where:

- \* the client-svc container is defined to represent the capabilities of the TP to be configured as an transparent client UNI (e.g., STM-N, Fiber Channel or transparent Ethernet);
- \* the OTN technology-specific Link Termination Point (LTP) augmentations are defined to represent the capabilities of the TP to be configured as an OTN UNI, together with the information about OTN label and bandwidth availability at the OTN UNI.

For example, the UNI TE Topology profile can be used to model features defined in [I-D.ogondio-opsawg-uni-topology]:

- \* The inter-domain-plug-id attribute would provide the same information as the attachment-id attribute defined in [I-D.ogondio-opsawg-uni-topology];
- \* The admin-status and oper-status that exists in this TE topology profile can provide the same information as the admin-status and oper-status attributes defined in [I-D.ogondio-opsawg-uni-topology].

Following the same approach in [I-D.ietf-ccamp-eth-client-te-topo-yang] and [I-D.ietf-ccamp-otn-topo-yang], the type and encapsulation-type attributes can be defined by technology-specific UNI augmentations to represent the capability of a TP to be configured as a L2VPN/L3VPN UNI Service Attachment Point (SAP).

The advantages of using a TE Topology profile would be having common solutions for:

- \* discovering UNIs as well as inter-domain NNI links, which is applicable to any technology (TE or non TE) used at the UNI or within the network;

- \* modelling non TE UNIs such as Ethernet, and TE UNIs such as OTN, as well as UNIs which can be configured as TE or non-TE (e.g., being configured as either Ethernet or OTN UNI).

## 2.2. Administrative and Operational status management

The TE Topology Model supports the management of administrative and operational state, including also the possibility to associate some administrative names, for nodes, termination points and links. This solution is generic and also does not require the network to be a TE network.

The following profile of the TE Topology Model can be used for administrative and operational state management:

```

module: ietf-te-topology
augment /nw:networks/nw:network/nw:network-types:
  +--rw te-topology!
augment /nw:networks/nw:network:
  +--rw te-topology-identifier
  |   +--rw provider-id?    te-global-id
  |   +--rw client-id?     te-global-id
  |   +--rw topology-id?   te-topology-id
  +--rw te!
  |   +--rw name?          string
augment /nw:networks/nw:network/nw:node:
  +--rw te-node-id?    te-types:te-node-id
  +--rw te!
  |   +--rw te-node-attributes
  |   |   +--rw admin-status?    te-types:te-admin-status
  |   |   +--rw name?           string
  |   +--ro oper-status?        te-types:te-oper-status
augment /nw:networks/nw:network/nt:link:
  +--rw te!
  |   +--rw te-link-attributes
  |   |   +--rw name?           string
  |   |   +--rw admin-status?    te-types:te-admin-status
  |   +--ro oper-status?        te-types:te-oper-status
augment /nw:networks/nw:network/nw:node/nt:termination-point:
  +--rw te-tp-id?    te-types:te-tp-id
  +--rw te!
  |   +--rw admin-status?    te-types:te-admin-status
  |   +--rw name?           string
  |   +--ro oper-status?    te-types:te-oper-status

```

Figure 2: Generic Topology with admin and operational state

The TE topology data model profile shown in Figure 2 is applicable to any technology (TE or non-TE) that requires management of the administrative and operational state and administrative names for nodes, termination points and links.

### 2.3. Geolocation

The TE Topology model supports the management of geolocation coordinates for nodes and termination points. This solution is generic and does not necessarily require the network to be a TE network.

The TE topology data model profile shown in Figure 3 can be used to model geolocation data for networks.

```
module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw te-topology!
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +--rw te-tp-id?   te-types:te-tp-id
    +--rw te!
      +--ro geolocation
        +--ro altitude?   int64
        +--ro latitude?   geographic-coordinate-degree
        +--ro longitude?  geographic-coordinate-degree
  augment /nw:networks/nw:network/nw:node:
    +--rw te-node-id?   te-types:te-node-id
    +--rw te!
      +--ro geolocation
        +--ro altitude?   int64
        +--ro latitude?   geographic-coordinate-degree
        +--ro longitude?  geographic-coordinate-degree
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +--rw te-tp-id?   te-types:te-tp-id
    +--rw te!
      +--ro geolocation
        +--ro altitude?   int64
        +--ro latitude?   geographic-coordinate-degree
        +--ro longitude?  geographic-coordinate-degree
```

Figure 3: Generic Topology with geolocation information

This profile is applicable to any network technology (TE or non-TE) that requires management of the geolocation information for its nodes and termination points.

## 2.4. Overlay and Underlay non-TE Topologies

The TE Topology model supports the management of overlay/underlay relationship for nodes and links, as described in section 5.8 of [RFC8795]. This solution is generic and does not require the network to be a TE network.

The following TE topology data model profile can be used to manage overlay/underlay network data:

```

module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw te-topology!
  augment /nw:networks/nw:network/nw:node:
    +---rw te-node-id?    te-types:te-node-id
    +---rw te!
    +---rw te-node-attributes
    +---rw underlay-topology {te-topology-hierarchy}?
    +---rw network-ref?    -> /nw:networks/network/network-id
  augment /nw:networks/nw:network/nt:link:
    +---rw te!
    +---rw te-link-attributes
    +---rw underlay {te-topology-hierarchy}?
    +---rw enabled?                boolean
    +---rw primary-path
    +---rw network-ref?
    |       -> /nw:networks/network/network-id
    +---rw path-element* [path-element-id]
    +---rw path-element-id                uint32
    +---rw (type)?
    +---:(numbered-link-hop)
    |   +---rw numbered-link-hop
    |   |   +---rw link-tp-id    te-tp-id
    |   |   +---rw hop-type?    te-hop-type
    |   |   +---rw direction?   te-link-direction
    +---:(unnumbered-link-hop)
    +---rw unnumbered-link-hop
    +---rw link-tp-id    te-tp-id
    +---rw node-id      te-node-id
    +---rw hop-type?    te-hop-type
    +---rw direction?   te-link-direction

```

Figure 4: Generic Topology with overlay/underlay information

This profile is applicable to any technology (TE or non-TE) when it is needed to manage the overlay/underlay information. It is also allows a TE underlay network to support a non-TE overlay network and, vice versa, a non-TE underlay network to support a TE overlay network.

## 2.5. Nodes with switching limitations

A node can have some switching limitations where connectivity is not possible between all its TP pairs, for example when:

- \* the node represents a physical device with switching limitations;
- \* the node represents an abstraction of a network topology.

This scenario is generic and applies to both TE and non-TE technologies.

A connectivity TE Topology profile data model supports the management of the node connectivity matrix to represent feasible connections between termination points across the nodes. This solution is generic and does not necessarily require a TE enabled network.

The following profile of the TE Topology model can be used for nodes with connectivity constraints:

```

module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw te-topology!
  augment /nw:networks/nw:network/nw:node:
    +--rw te-node-id?    te-types:te-node-id
    +--rw te!
      +--rw te-node-attributes
        +--rw connectivity-matrices
          +--rw number-of-entries?    uint16
          +--rw is-allowed?           boolean
          +--rw connectivity-matrix* [id]
            +--rw id                  uint32
            +--rw from
              | +--rw tp-ref?         leafref
            +--rw to
              | +--rw tp-ref?         leafref
            +--rw is-allowed?         boolean

```

Figure 5: Generic Topology with connectivity constraints

The TE topology data model profile shown in Figure 5 is applicable to any technology (TE or non-TE) networks that requires managing nodes with certain connectivity constraints. When used with TE technologies, additional TE attributes, as defined in [RFC8795], can also be provided.

### 3. Technology-specific augmentations

There are two main options to define technology-specific Topology Models which can use the attributes defined in the TE Topology Model [RFC8795].

Both options are applicable to any possible profile of the TE Topology Model, such as those defined in Section 2.

The first option is to define a technology-specific TE Topology Model which augments the TE Topology Model, as shown in Figure 6:

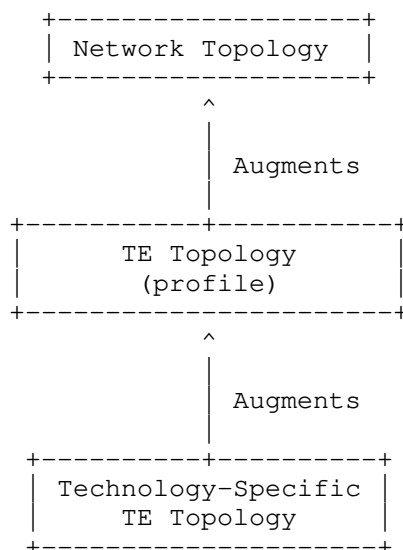


Figure 6: Augmenting the TE Topology Model

This approach is more suitable for cases when the technology-specific TE topology model provides augmentations to the TE Topology constructs, such as bandwidth information (e.g., link bandwidth), tunnel termination points (TTPs) or connectivity matrices. It also allows providing augmentations to the Network Topology constructs, such as nodes, links, and termination points (TPs).

This is the approach currently used in [I-D.ietf-ccamp-eth-client-te-topo-yang] and [I-D.ietf-ccamp-otn-topo-yang].

It is worth noting that a profile of the technology-specific TE Topology model not using any TE topology attribute or constructs can be used to address any use case that do not require these attributes. In this case, only the te-topology presence container of the TE Topology Model needs to be implemented.

The second option is to define a technology-specific Network Topology Model which augments the Network Topology Model and to rely on the multiple inheritance capability, which is implicit in the network-types definition of [RFC8345], to allow using also the generic attributes defined in the TE Topology model:

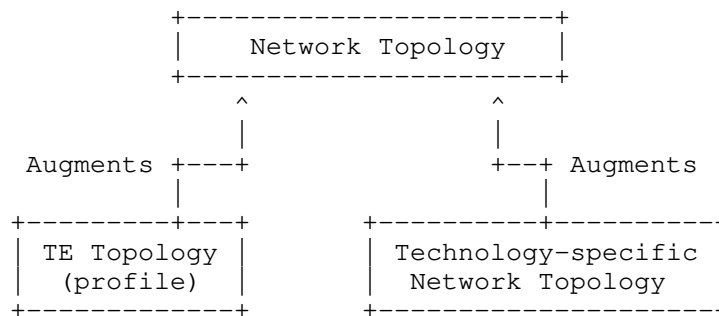


Figure 7: Augmenting the Network Topology Model with multi-inheritance

This approach is more suitable in cases where the technology-specific Network Topology Model provides augmentation only to the constructs defined in the Network Topology Model, such as nodes, links, and termination points (TPs). Therefore, with this approach, only the generic attributes defined in the TE Topology Model could be used.

It is also worth noting that in this case, technology-specific augmentations for the bandwidth information could not be defined.

In principle, it would be also possible to define both a technology specific TE Topology Model which augments the TE Topology Model, and a technology-specific Network Topology Model which augments the Network Topology Model and to rely on the multiple inheritance capability, as shown in Figure 8:

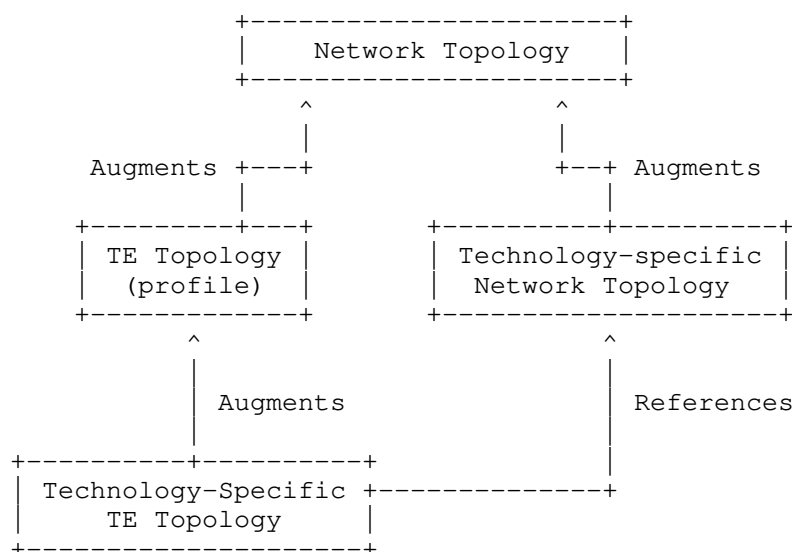


Figure 8: Augmenting both the Network and TE Topology Models

This option does not provide any technical advantage with respect to the first option, shown in Figure 6, but could be useful to add augmentations to the TE Topology constructs and to re-use an already existing technology-specific Network Topology Model.

It is worth noting that the technology-specific TE Topology model can reference constructs defined by the technology-specific Network Topology model but it could not augment constructs defined by the technology-specific Network Topology model.

### 3.1. Multi-inheritance

As described in section 4.1 of [RFC8345], the network types should be defined using presence containers to allow the representation of network subtypes.

The hierarchy of network subtypes can be single hierarchy, as shown in Figure 6. In this case, each presence container contains at most one child presence container, as shows in the JSON code below:



```
{
  "ietf-network:ietf-network": {
    "ietf-te-topology:te-topology": {
      "example-te-topology": {}
    }
  }
}
```

The hierarchy of network subtypes can also be multi-hierarchy, as shown in Figure 7 and Figure 8. In this case, one presence container can contain more than one child presence containers, as show in the JSON codes below:

```
{
  "ietf-network:ietf-network": {
    "ietf-te-topology:te-topology": {}
    "example-network-topology": {}
  }
}

{
  "ietf-network:ietf-network": {
    "ietf-te-topology:te-topology": {
      "example-te-topology": {}
    }
    "example-network-topology": {}
  }
}
```

Other examples of multi-hierarchy topologies are described in [I-D.ietf-teas-yang-sr-te-topo].

### 3.2. Example (Link augmentation)

This section provides an example on how technology-specific attributes can be added to the Link construct:

```

+--rw link* [link-id]
  +--rw link-id          link-id
  +--rw source
    | +--rw source-node?  -> ../../../../nw:node/node-id
    | +--rw source-tp?    leafref
  +--rw destination
    | +--rw dest-node?    -> ../../../../nw:node/node-id
    | +--rw dest-tp?      leafref
  +--rw supporting-link* [network-ref link-ref]
    | +--rw network-ref
    | | -> ../../../../nw:supporting-network/network-ref
    | +--rw link-ref      leafref
  +--rw example-link-attributes
    | <...>
  +--rw te!
    +--rw te-link-attributes
      +--rw name?                                string
      +--rw example-te-link-attributes
        | <...>
      +--rw max-link-bandwidth
        +--rw te-bandwidth
          +--rw (technology)?
            +--:(generic)
            | +--rw generic?    te-bandwidth
            +--:(example)
            | +--rw example?    example-bandwidth

```

Figure 9: Augmenting the Link with technology-specific attributes

The technology-specific attributes within the example-link-attributes container can be defined either in the technology-specific TE Topology Model (Option 1) or in the technology-specific Network Topology Model (Option 2 or Option 3). These attributes can only be non-TE and do not require the implementation of the te container.

The technology-specific attributes within the example-te-link-attributes container as well as the example max-link-bandwidth can only be defined in the technology-specific TE Topology Model (Option 1 or Option 3). These attributes can be TE or non-TE and require the implementation of the te container.

#### 4. Implemented profiles

When a server implements a profile of the TE topology model, it is not clear how the server can report to the client the subset of the model being implemented.

It is also worth noting that the supported profile may also depend on other attributes (for example the network type).

In case the TE topology profile is reported by the server to the client, the server will report in the operational datastore only the leaves which have been implemented, as described in section 5.3 of [RFC8342].

More investigation is required in case the TE topology profile is configured by the client.

## 5. Security Considerations

This document provides only information about how the TE Topology Model, as defined in [RFC8795], can be profiled to address some scenarios which are not considered as TE.

As such, this document does not introduce any additional security considerations besides those already defined in [RFC8795].

## 6. IANA Considerations

This document requires no IANA actions.

## Acknowledgments

The authors would like to thank Daniele Ceccarelli, Jonas Ahlberg and Scott Mansfield for providing useful suggestions for this draft.

This document was prepared using kramdown.

Previous versions of this document was prepared using 2-Word-v2.0.template.dot.

## References

### Normative References

- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.

- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.

#### Informative References

- [I-D.ietf-ccamp-eth-client-te-topo-yang]  
Zheng, H., Guo, A., Busi, I., Xu, Y., Zhao, Y., and X. Liu, "A YANG Data Model for Ethernet TE Topology", Work in Progress, Internet-Draft, draft-ietf-ccamp-eth-client-te-topo-yang-01, 7 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-ccamp-eth-client-te-topo-yang-01.txt>>.
- [I-D.ietf-ccamp-otn-topo-yang]  
Zheng, H., Busi, I., Liu, X., Belotti, S., and O. G. D. Dios, "A YANG Data Model for Optical Transport Network Topology", Work in Progress, Internet-Draft, draft-ietf-ccamp-otn-topo-yang-13, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-ccamp-otn-topo-yang-13.txt>>.
- [I-D.ietf-teas-rfc3272bis]  
Farrel, A., "Overview and Principles of Internet Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc3272bis-13, 8 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-rfc3272bis-13.txt>>.
- [I-D.ietf-teas-yang-sr-te-topo]  
Liu, X., Bryskin, I., Beeram, V. P., Saad, T., Shah, H., and S. Litkowski, "YANG Data Model for SR and SR TE Topologies on MPLS Data Plane", Work in Progress, Internet-Draft, draft-ietf-teas-yang-sr-te-topo-11, 24 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-yang-sr-te-topo-11.txt>>.
- [I-D.ogondio-opsawg-uni-topology]  
Dios, O. G. D., Barguil, S., Wu, Q., and M. Boucadair, "A YANG Model for User-Network Interface (UNI) Topologies", Work in Progress, Internet-Draft, draft-ogondio-opsawg-uni-topology-01, 2 April 2020, <<https://www.ietf.org/archive/id/draft-ogondio-opsawg-uni-topology-01.txt>>.

Contributors

Aihua Guo  
Futurewei Inc.

Email: aihuaguo.ietf@gmail.com

Haomian Zheng  
Huawei

Email: zhenghaomian@huawei.com

Vishnu Pavan Beeram  
Juniper Networks

Email: vbeeram@juniper.net

Sergio Belotti  
Nokia

Email: sergio.belotti@nokia.com

Authors' Addresses

Italo Busi  
Huawei

Email: italo.busi@huawei.com

Xufeng Liu  
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Igor Bryskin  
Individual

Email: i\_bryskin@yahoo.com

Tarek Saad  
Juniper Networks

Email: tsaad@juniper.net

Oscar Gonzalez de Dios  
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 13, 2022

I. Busi  
H. Zheng  
Huawei Technologies  
A. Guo  
Futurewei Inc.  
X. Liu  
Volta Networks  
July 12, 2021

A YANG Data Model for MPLS-TE Topology  
draft-busizheng-teas-yang-te-mpls-topology-01

Abstract

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) with Traffic Engineering (MPLS-TE) networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 13, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .                        | 2  |
| 1.1. Tree Diagram . . . . .                      | 2  |
| 1.2. Prefixes in Data Node Names . . . . .       | 3  |
| 2. MPLS-TE Types Overview . . . . .              | 3  |
| 3. MPLS-TE Topology Model Overview . . . . .     | 3  |
| 3.1. TE Label Augmentations . . . . .            | 4  |
| 3.2. MPLS-TP Topology . . . . .                  | 5  |
| 4. YANG model for common MPLS-TE Types . . . . . | 6  |
| 5. YANG model for MPLS-TE Topology . . . . .     | 8  |
| 5.1. YANG Tree . . . . .                         | 8  |
| 5.2. YANG Code . . . . .                         | 8  |
| 6. Security Considerations . . . . .             | 11 |
| 7. IANA Considerations . . . . .                 | 11 |
| 8. References . . . . .                          | 11 |
| 8.1. Normative References . . . . .              | 11 |
| 8.2. Informative References . . . . .            | 12 |
| Acknowledgments . . . . .                        | 12 |
| Authors' Addresses . . . . .                     | 12 |

## 1. Introduction

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) with Traffic Engineering (MPLS-TE) networks.

This document also defines a collection of common data types and groupings in YANG data modeling language for MPLS-TE networks. These derived common types and groupings are intended to be imported by the MPLS-TE topology model, defined in this document, as well as by the MPLS-TE tunnel model, defined in [I-D.ietf-teas-yang-te-mpls].

Multi-Protocol Label Switching - Transport Profile (MPLS-TP) is a profile of the MPLS protocol that is used in packet switched transport networks and operated in a similar manner to other existing transport technologies (e.g., OTN), as described in [RFC5921]. The YANG model defined in this document can also be for MPLS-TP networks.

### 1.1. Tree Diagram

A simplified graphical representation of the data model is used in Section 5.1 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].



## 1.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

| Prefix          | YANG module             | Reference                       |
|-----------------|-------------------------|---------------------------------|
| rt-types        | ietf-routing-types      | [RFC8294]                       |
| tet             | ietf-te-topology        | [RFC8795]                       |
| tet-pkt         | ietf-te-topology-packet | [I-D.ietf-teas-yang-l3-te-topo] |
| te-packet-types | ietf-te-packet-types    | [I-D.ietf-teas-yang-l3-te-topo] |
| mte-types       | ietf-mpls-te-types      | This document                   |
| tet-mpls        | ietf-te-mpls-topology   | This document                   |

Table 1: Prefixes and corresponding YANG modules

## 2. MPLS-TE Types Overview

The module `ietf-mpls-te-types` contains the following YANG types and groupings which can be reused by MPLS-TE YANG models:

`load-balancing-type` This identify defines the types of load-balancing algorithms used on bundled MPLS-TE link.

`te-mpls-label-hop` This grouping is used for the augmentation of TE label for MPLS-TE path.

## 3. MPLS-TE Topology Model Overview

The MPLS-TE technology specific topology model augments the `ietf-te-topology-packet` YANG module, defined in [I-D.ietf-teas-yang-l3-te-topo], which in turns augment the generic `ietf-te-topology` YANG module, defined in [RFC8795], as shown in Figure 1.

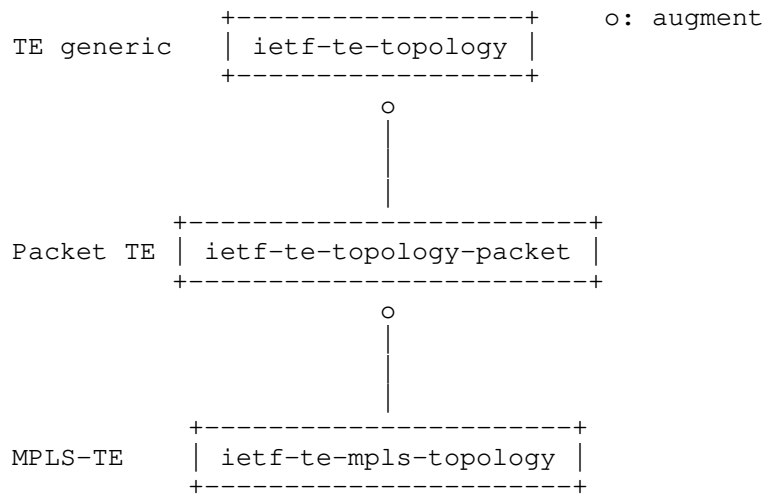


Figure 1: Relationship between MPLS-TE, Packet-TE and TE topology models

Given the guidance for augmentation in [RFC8795], the following technology-specific augmentations need to be provided:

- o A network-type to indicate that the TE topology is an MPLS-TE Topology, as follow:

```

augment /nw:networks/nw:network/nw:network-types/tet:te-topology
  /tet-pkt:packet:
    +---rw mpls-topology!
  
```

- o TE Label Augmentations as described in Section 3.1.

Note: TE Bandwidth Augmentations for paths, LSPs and links are provided by the ietf-te-topology-packet module, defined in [I-D.ietf-teas-yang-l3-te-topo].

### 3.1. TE Label Augmentations

In MPLS-TE, the label allocation is done by NE, information about label values availability is not necessary to be provided to the controller. Moreover, MPLS-TE tunnels are currently established within a single domain.

Therefore this document does not define any MPLS-TE technology-specific augmentations, of the TE Topology model, for the TE label since no TE label related attributes should be instantiated for MPLS-TE Topologies.

Open issue: shall this module allows the setup of MPLS-TE multi-domain tunnels?

### 3.2. MPLS-TP Topology

Multi-Protocol Label Switching - Transport Profile (MPLS-TP) is a profile of the MPLS protocol that is used in packet switched transport networks and operated in a similar manner to other existing transport technologies (e.g., OTN), as described in [RFC5921].

Therefore YANG model defined in this document can also be applicable for MPLS-TP networks.

However, as described in [RFC5921], MPLS-TP networks support bidirectional LSPs and require no ECMP and no PHP. When reporting the topology for an MPLS-TP network, additional information is required to indicate whether the network support these MPLS-TP characteristics.

It is worth noting that [RFC8795] is already capable to model TE topologies supporting either unidirectional or bidirectional LSPs: all bidirectional TE links can support bidirectional LSPs and all the links can support unidirectional LSPs and it is always possible to associated unidirectional LSPs as long as they belong to the same tunnel.

When setting up bidirectional LSPs (e.g., MPLS-TP LSPs) only bidirectional TE Links are selected by path computation.

In order to allow reporting that ECMP is not affecting forwarding the packets of a given LSP, the load-balancing-type attribute reports whether a LAG or TE Bundled Link performs load-balancing on a per-flow or per-top-label:

```
augment /nw:networks/nw:network/nt:link/tet:te:
  +--rw load-balancing-type?  mte-types:load-balancing-type
```

When setting up LSPs which do not requires ECMP (e.g., MPLS-TP LSPs) only Links that are not part of a LAG or TE Bundle or that performs per-top-label load balancing are selected by path computation.

It is assumed that almost all the MPLS-TE nodes are capable to support Ultimate Hop Popping (UHP). However, if some interfaces are not able to support UHP, they can report it in the MPLS-TE topology:

```
augment /nw:networks/nw:network/nw:node/nt:termination-point
  /tet:te:
    +--ro uhp-incapable?  empty
```

When setting up LSPs which do not requires PHP (e.g., MPLS-TP LSPs) only the interfaces (LTPs) which are capable to support UHP in the destination node are selected by path computation.

#### 4. YANG model for common MPLS-TE Types

```
<CODE BEGINS>file "ietf-mpls-te-types@2021-07-12.yang"
module ietf-mpls-te-types {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-te-types";

  prefix "mte-types";

  import ietf-routing-types {
    prefix "rt-types";
  }

  organization
    "Internet Engineering Task Force (IETF) TEAS WG";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/teas/>
    WG List:  <mailto:teas@ietf.org>

    Editor: Italo Busi
            <mailto:italo.busi@huawei.com>

    Editor: Haomian Zheng
            <mailto:zhenghaomian@huawei.com>

    Editor: Aihua Guo
            <mailto:aihuaguo.ietf@gmail.com>

    Editor: Xufeng Liu
            <mailto:xufeng.liu.ietf@gmail.com>";

  description
    "This module defines technology-specific MPLS-TE types
    data model.

    Copyright (c) 2021 IETF Trust and the persons identified
    as authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with
    or without modification, is permitted pursuant to, and
    subject to the license terms contained in, the Simplified
    BSD License set forth in Section 4.c of the IETF Trust's
    Legal Provisions Relating to IETF Documents
    (http://trustee.ietf.org/license-info).
```

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2021-07-12 {
  description
    "Initial Version";
  reference
    "draft-busizheng-teas-yang-te-mpls-topology";
}

/*
 * Typedefs
 */

typedef load-balancing-type {
  type enumeration {
    enum per-flow {
      description
        "The load-balancing algorithm ensures that packets
        characterized as the same flow (e.g. based on IP 5-tuple)
        that egress on a LAG or a bundled TE link are forwarded
        on the same component link.

        Packets for different flows within the same LSP can be
        forwarded on different component links.";
    }
    enum per-top-label {
      description
        "The load-balancing algorithm ensures incoming MPLS
        packets with the same top MPLS label and that egress on
        on a LAG or bundled TE link are forwarded on the same
        component link.

        Packets for different flows within the same LSP are
        forwarded on the same component link.";
    }
  }
  description
    "The type of load balancing used on bundled links.";
} // typedef load-balancing-type

/*
 * Groupings
 */

grouping te-mpls-label-hop {
  description
    "MPLS-TE Label Hop.";
```

```

    leaf mpls-label {
        type rt-types:mpls-label;
        description
            "MPLS Label.";
    }
} // grouping te-mpls-label-hop
}
<CODE ENDS>

```

Figure 2: MPLS-TE Types YANG model

## 5. YANG model for MPLS-TE Topology

### 5.1. YANG Tree

Figure 3 below shows the tree diagram of the YANG model defined in module `ietf-te-mpls-topology.yang`.

```

module: ietf-te-mpls-topology

  augment /nw:networks/nw:network/nw:network-types/tet:te-topology
    /tet-pkt:packet:
      +--rw mpls-topology!
  augment /nw:networks/nw:network/nt:link/tet:te:
    +--rw load-balancing-type? mte-types:load-balancing-type
  augment /nw:networks/nw:network/nw:node/nt:termination-point
    /tet:te:
      +--ro uhp-incapable? empty

```

Figure 3: MPLS-TE topology YANG tree

### 5.2. YANG Code

```

<CODE BEGINS>file "ietf-te-mpls-topology@2021-07-12.yang"
module ietf-te-mpls-topology {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-te-mpls-topology";

    prefix "tet-mpls";

    import ietf-network {
        prefix "nw";
    }

    import ietf-network-topology {
        prefix "nt";
    }
}

```

```
import ietf-te-topology {
  prefix "tet";
}

import ietf-te-topology-packet {
  prefix "tet-pkt";
}

import ietf-mpls-te-types {
  prefix "mte-types";
}

organization
  "Internet Engineering Task Force (IETF) TEAS WG";
contact
  "WG Web:  <https://datatracker.ietf.org/wg/teas/>
  WG List:  <mailto:teas@ietf.org>

  Editor: Italo Busi
         <mailto:italo.busi@huawei.com>

  Editor: Haomian Zheng
         <mailto:zhenghaomian@huawei.com>

  Editor: Aihua Guo
         <mailto:aihuaguo.ietf@gmail.com>

  Editor: Xufeng Liu
         <mailto:xufeng.liu.ietf@gmail.com>";

description
  "This module defines technology-specific MPLS-TE topology
  data model.

  Copyright (c) 2021 IETF Trust and the persons identified
  as authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with
  or without modification, is permitted pursuant to, and
  subject to the license terms contained in, the Simplified
  BSD License set forth in Section 4.c of the IETF Trust's
  Legal Provisions Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices.";

revision 2021-07-12 {
```

```
    description
      "Initial Version";
    reference
      "draft-busizheng-teas-yang-te-mpls-topology";
  }

/*
 * Augmentations
 */

augment "/nw:networks/nw:network/nw:network-types/"
  + "tet:te-topology/tet-pkt:packet" {
  description
    "Augment network types to include MPLS-TE Topology Type";
  container mpls-topology {
    presence
      "Indicates an MPLS-TE Topology Type.";
    description
      "Its presence indicates an MPLS-TE Topology";
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te" {
  when "../nw:network-types/tet:te-topology/"
    + "tet-pkt:packet/tet-mpls:mpls-topology" {
    description
      "Augment MPLS-TE Topology.";
  }
  description
    "Augment TE Link.";

  leaf load-balancing-type {
    type mte-types:load-balancing-type;
    default 'per-flow';
    description
      "Indicates the type of load-balancing (per-flow or per-LSP)
       performed by the bundled TE Link.

       This leaf is not present when the TE Link is not bundled.";
  } // leaf load-balancing-type
}

augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
  + "tet:te" {
  when "../nw:network-types/tet:te-topology/"
    + "tet-pkt:packet/tet-mpls:mpls-topology" {
    description "Augment MPLS-TE Topology.";
  }
}
```



```
description "Augment LTP.";

leaf uhp-incapable {
  type empty;
  config false;
  description
    "When present, indicates that the LTP is not capable to
    support Ultimate Hop Popping (UHP).";
} // leaf uhp-incapable
}
}
<CODE ENDS>
```

Figure 4: MPLS-TE topology YANG module

## 6. Security Considerations

To be added.

## 7. IANA Considerations

To be added.

## 8. References

### 8.1. Normative References

- [I-D.ietf-teas-yang-l3-te-topo] Liu, X., Bryskin, I., Beeram, V. P., Saad, T., Shah, H., and O. G. D. Dios, "YANG Data Model for Layer 3 TE Topologies", draft-ietf-teas-yang-l3-te-topo-11 (work in progress), July 2021.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.

## 8.2. Informative References

- [I-D.ietf-teas-yang-te-mpls]  
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., and I. Bryskin, "A YANG Data Model for MPLS Traffic Engineering Tunnels", draft-ietf-teas-yang-te-mpls-03 (work in progress), March 2020.
- [RFC5921] Bocci, M., Ed., Bryant, S., Ed., Frost, D., Ed., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, DOI 10.17487/RFC5921, July 2010, <<https://www.rfc-editor.org/info/rfc5921>>.

## Acknowledgments

The authors would also like to thank Tarek Saad, Vishnu Pavan Beeram, Rakesh Gandhi, Xufeng Liu, Igor Bryskin for their input on how to support MPLS-TP features (bidirectional LSPs, no ECMP, no PHP) using a common MPLS-TE topology model.

We thank Loa Andersson and Igor Bryskin for providing useful suggestions for this draft.

This document was prepared using kramdown.

Previous versions of this document was prepared using 2-Word-v2.0.template.dot.

## Authors' Addresses

Italo Busi  
Huawei Technologies

Email: [italo.busi@huawei.com](mailto:italo.busi@huawei.com)

Haomian Zheng  
Huawei Technologies

Email: [zhenghaomian@huawei.com](mailto:zhenghaomian@huawei.com)

Aihua Guo  
Futurewei Inc.

Email: [aihuaguo.ietf@gmail.com](mailto:aihuaguo.ietf@gmail.com)

Xufeng Liu  
Volta Networks

Email: xufeng.liu.ietf@gmail.com

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 30 October 2022

I. Busi  
Huawei Technologies  
A. Guo  
Futurewei Inc.  
X. Liu  
Volta Networks  
T. Saad  
Juniper Networks  
R. Gandhi  
Cisco Systems, Inc.  
28 April 2022

A YANG Data Model for MPLS-TE Topology  
draft-busizheng-teas-yang-te-mpls-topology-03

Abstract

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) with Traffic Engineering (MPLS-TE) networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 30 October 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .                        | 2  |
| 1.1. Tree Diagram . . . . .                      | 2  |
| 1.2. Prefixes in Data Node Names . . . . .       | 3  |
| 2. MPLS-TE Types Overview . . . . .              | 3  |
| 3. MPLS-TE Topology Model Overview . . . . .     | 4  |
| 3.1. TE Label Augmentations . . . . .            | 5  |
| 3.2. MPLS-TP Topology . . . . .                  | 5  |
| 4. YANG model for common MPLS-TE Types . . . . . | 6  |
| 5. YANG model for MPLS-TE Topology . . . . .     | 8  |
| 5.1. YANG Tree . . . . .                         | 8  |
| 5.2. YANG Code . . . . .                         | 9  |
| 6. Security Considerations . . . . .             | 12 |
| 7. IANA Considerations . . . . .                 | 12 |
| 8. References . . . . .                          | 12 |
| 8.1. Normative References . . . . .              | 12 |
| 8.2. Informative References . . . . .            | 12 |
| Acknowledgments . . . . .                        | 13 |
| Contributors . . . . .                           | 13 |
| Authors' Addresses . . . . .                     | 13 |

## 1. Introduction

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) with Traffic Engineering (MPLS-TE) networks.

This document also defines a collection of common data types and groupings in YANG data modeling language for MPLS-TE networks. These derived common types and groupings are intended to be imported by the MPLS-TE topology model, defined in this document, as well as by the MPLS-TE tunnel model, defined in [I-D.ietf-teas-yang-te-mpls].

Multi-Protocol Label Switching - Transport Profile (MPLS-TP) is a profile of the MPLS protocol that is used in packet switched transport networks and operated in a similar manner to other existing transport technologies (e.g., OTN), as described in [RFC5921]. The YANG model defined in this document can also be for MPLS-TP networks.

### 1.1. Tree Diagram

A simplified graphical representation of the data model is used in Section 5.1 of this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

## 1.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

| Prefix          | YANG module             | Reference                       |
|-----------------|-------------------------|---------------------------------|
| rt-types        | ietf-routing-types      | [RFC8294]                       |
| tet             | ietf-te-topology        | [RFC8795]                       |
| tet-pkt         | ietf-te-topology-packet | [I-D.ietf-teas-yang-l3-te-topo] |
| te-packet-types | ietf-te-packet-types    | [I-D.ietf-teas-yang-l3-te-topo] |
| mte-types       | ietf-mpls-te-types      | This document                   |
| tet-mpls        | ietf-te-mpls-topology   | This document                   |

Table 1: Prefixes and corresponding YANG modules

## 2. MPLS-TE Types Overview

The module `ietf-mpls-te-types` contains the following YANG types and groupings which can be reused by MPLS-TE YANG models:

`load-balancing-type`:

This identify defines the types of load-balancing algorithms used on bundled MPLS-TE link.

`te-mpls-label-hop`:

This grouping is used for the augmentation of TE label for MPLS-TE path.

### 3. MPLS-TE Topology Model Overview

The MPLS-TE technology specific topology model augments the `ietf-te-topology-packet` YANG module, defined in [I-D.ietf-teas-yang-l3-te-topo], which in turns augment the generic `ietf-te-topology` YANG module, defined in [RFC8795], as shown in Figure 1.

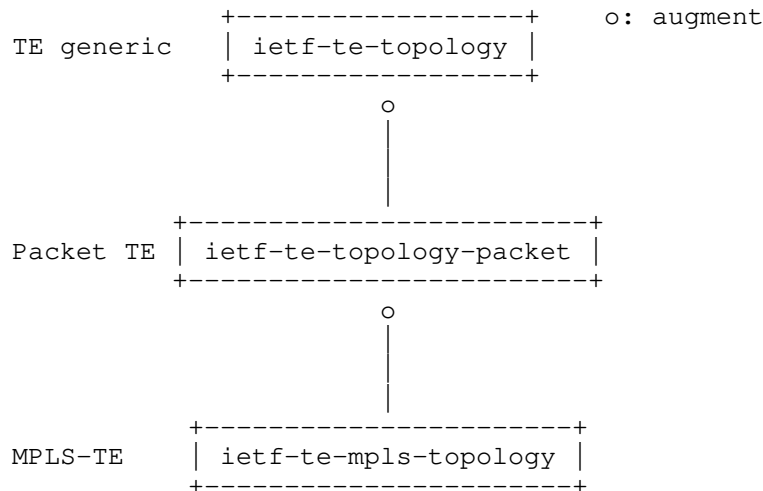


Figure 1: Relationship between MPLS-TE, Packet-TE and TE topology models

Given the guidance for augmentation in [RFC8795], the following technology-specific augmentations need to be provided:

- \* A network-type to indicate that the TE topology is an MPLS-TE Topology, as follow:

```

augment /nw:networks/nw:network/nw:network-types/tet:te-topology
  /tet-pkt:packet:
    +---rw mpls-topology!
  
```

- \* TE Label Augmentations as described in Section 3.1.

Note: TE Bandwidth Augmentations for paths, LSPs and links are provided by the `ietf-te-topology-packet` module, defined in [I-D.ietf-teas-yang-l3-te-topo].

### 3.1. TE Label Augmentations

In MPLS-TE, the label allocation is done by NE, information about label values availability is not necessary to be provided to the controller. Moreover, MPLS-TE tunnels are currently established within a single domain.

Therefore this document does not define any MPLS-TE technology-specific augmentations, of the TE Topology model, for the TE label since no TE label related attributes should be instantiated for MPLS-TE Topologies.

Open issue: shall this module allows the setup of MPLS-TE multi-domain tunnels?

### 3.2. MPLS-TP Topology

Multi-Protocol Label Switching - Transport Profile (MPLS-TP) is a profile of the MPLS protocol that is used in packet switched transport networks and operated in a similar manner to other existing transport technologies (e.g., OTN), as described in [RFC5921].

Therefore YANG model defined in this document can also be applicable for MPLS-TP networks.

However, as described in [RFC5921], MPLS-TP networks support bidirectional LSPs and require no ECMP and no PHP. When reporting the topology for an MPLS-TP network, additional information is required to indicate whether the network support these MPLS-TP characteristics.

It is worth noting that [RFC8795] is already capable to model TE topologies supporting either unidirectional or bidirectional LSPs: all bidirectional TE links can support bidirectional LSPs and all the links can support unidirectional LSPs and it is always possible to associated unidirectional LSPs as long as they belong to the same tunnel.

When setting up bidirectional LSPs (e.g., MPLS-TP LSPs) only bidirectional TE Links are selected by path computation.

In order to allow reporting that ECMP is not affecting forwarding the packets of a given LSP, the load-balancing-type attribute reports whether a LAG or TE Bundled Link performs load-balancing on a per-flow or per-top-label:

```
augment /nw:networks/nw:network/nt:link/tet:te:
  +---rw load-balancing-type?  mte-types:load-balancing-type
```



When setting up LSPs which do not requires ECMP (e.g., MPLS-TP LSPs) only Links that are not part of a LAG or TE Bundle or that performs per-top-label load balancing are selected by path computation.

It is assumed that almost all the MPLS-TE nodes are capable to support Ultimate Hop Popping (UHP). However, if some interfaces are not able to support UHP, they can report it in the MPLS-TE topology:

```
augment /nw:networks/nw:network/nw:node/nt:termination-point
  /tet:te:
    +--ro uhp-incapable?    empty
```

When setting up LSPs which do not requires PHP (e.g., MPLS-TP LSPs) only the interfaces (LTPs) which are capable to support UHP in the destination node are selected by path computation.

#### 4. YANG model for common MPLS-TE Types

```
<CODE BEGINS> file "ietf-mpls-te-types@2021-10-12.yang"
module ietf-mpls-te-types {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-te-types";

  prefix "mte-types";

  import ietf-routing-types {
    prefix "rt-types";
  }

  organization
    "Internet Engineering Task Force (IETF) TEAS WG";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/teas/>
    WG List:  <mailto:teas@ietf.org>

    Editor:   Italo Busi
              <mailto:italo.busi@huawei.com>

    Editor:   Haomian Zheng
              <mailto:zhenghaomian@huawei.com>

    Editor:   Aihua Guo
              <mailto:aihuaguo.ietf@gmail.com>

    Editor:   Xufeng Liu
              <mailto:xufeng.liu.ietf@gmail.com>

    Editor:   Vishnu Pavan Beeram
```

<mailto:vbeeram@juniper.net>

Editor: Tarek Saad  
<mailto:tasaad@juniper.net>

Editor: Rakesh Gandhi  
<mailto:rgandhi@cisco.com>

Editor: Igor Bryskin  
<mailto:i\_bryskin@yahoo.com>

Editor: Yanlei Zheng  
<mailto:zhengyanlei@chinaunicom.cn>;

#### description

"This module defines technology-specific MPLS-TE types data model.

Copyright (c) 2021 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2021-10-12 {  
  description  
    "Initial Version";  
  reference  
    "draft-busizheng-teas-yang-te-mpls-topology-02";  
}
```

```
/*  
 * Typedefs  
 */
```

```
typedef load-balancing-type {  
  type enumeration {  
    enum per-flow {  
      description  
        "The load-balancing algorithm ensures that packets  
        characterized as the same flow (e.g. based on IP 5-tuple)
```

```

        that egress on a LAG or a bundled TE link are forwarded
        on the same component link.

        Packets for different flows within the same LSP can be
        forwarded on different component links.";
    }
    enum per-top-label {
        description
            "The load-balancing algorithm ensures incoming MPLS
            packets with the same top MPLS label and that egress on
            on a LAG or bundled TE link are forwarded on the same
            component link.

            Packets for different flows within the same LSP are
            forwarded on the same component link.";
    }
}
description
    "The type of load balancing used on bundled links.";
} // typedef load-balancing-type

/*
 * Groupings
 */

grouping te-mpls-label-hop {
    description
        "MPLS-TE Label Hop.";

    leaf mpls-label {
        type rt-types:mpls-label;
        description
            "MPLS Label.";
    }
} // grouping te-mpls-label-hop
}
<CODE ENDS>

```

Figure 2: MPLS-TE Types YANG model

## 5. YANG model for MPLS-TE Topology

### 5.1. YANG Tree

Figure 3 below shows the tree diagram of the YANG model defined in module `ietf-te-mpls-topology.yang`.

```

module: ietf-te-mpls-topology

  augment /nw:networks/nw:network/nw:network-types/tet:te-topology
    /tet-pkt:packet:
      +--rw mpls-topology!
  augment /nw:networks/nw:network/nt:link/tet:te:
    +--rw load-balancing-type? mte-types:load-balancing-type
  augment /nw:networks/nw:network/nw:node/nt:termination-point
    /tet:te:
      +--ro uhp-incapable? empty

```

Figure 3: MPLS-TE topology YANG tree

## 5.2. YANG Code

```

<CODE BEGINS> file "ietf-te-mpls-topology@2021-07-12.yang"
module ietf-te-mpls-topology {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-mpls-topology";

  prefix "tet-mpls";

  import ietf-network {
    prefix "nw";
  }

  import ietf-network-topology {
    prefix "nt";
  }

  import ietf-te-topology {
    prefix "tet";
  }

  import ietf-te-topology-packet {
    prefix "tet-pkt";
  }

  import ietf-mpls-te-types {
    prefix "mte-types";
  }

  organization
    "Internet Engineering Task Force (IETF) TEAS WG";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/teas/>
    WG List:  <mailto:teas@ietf.org>

```

Editor: Italo Busi  
<mailto:italo.busi@huawei.com>

Editor: Haomian Zheng  
<mailto:zhenghaomian@huawei.com>

Editor: Aihua Guo  
<mailto:aihuaguo.ietf@gmail.com>

Editor: Xufeng Liu  
<mailto:xufeng.liu.ietf@gmail.com>

Editor: Vishnu Pavan Beeram  
<mailto:vbeeram@juniper.net>

Editor: Tarek Saad  
<mailto:tsaad@juniper.net>

Editor: Rakesh Gandhi  
<mailto:rgandhi@cisco.com>

Editor: Igor Bryskin  
<mailto:i\_bryskin@yahoo.com>

Editor: Yanlei Zheng  
<mailto:zhengyanlei@chinaunicom.cn>";

description

"This module defines technology-specific MPLS-TE topology data model.

Copyright (c) 2021 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

revision 2021-10-12 {  
 description  
 "Initial Version";  
 reference

```
    "draft-busizheng-teas-yang-te-mpls-topology-02";
}

/*
 * Augmentations
 */

augment "/nw:networks/nw:network/nw:network-types/"
  + "tet:te-topology/tet-pkt:packet" {
  description
    "Augment network types to include MPLS-TE Topology Type";
  container mpls-topology {
    presence
      "Indicates an MPLS-TE Topology Type.";
    description
      "Its presence indicates an MPLS-TE Topology";
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te" {
  when "../nw:network-types/tet:te-topology/"
    + "tet-pkt:packet/tet-mpls:mpls-topology" {
    description
      "Augment MPLS-TE Topology.";
  }
  description
    "Augment TE Link.";

  leaf load-balancing-type {
    type mte-types:load-balancing-type;
    default 'per-flow';
    description
      "Indicates the type of load-balancing (per-flow or per-LSP)
       performed by the bundled TE Link.

       This leaf is not present when the TE Link is not bundled.";
  } // leaf load-balancing-type
}

augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
  + "tet:te" {
  when "../nw:network-types/tet:te-topology/"
    + "tet-pkt:packet/tet-mpls:mpls-topology" {
    description "Augment MPLS-TE Topology.";
  }
  description "Augment LTP.";

  leaf uhp-incapable {
```

```
    type empty;
    config false;
    description
        "When present, indicates that the LTP is not capable to
        support Ultimate Hop Popping (UHP).";
    } // leaf uhp-incapable
}
}
<CODE ENDS>
```

Figure 4: MPLS-TE topology YANG module

## 6. Security Considerations

To be added.

## 7. IANA Considerations

To be added.

## 8. References

### 8.1. Normative References

- [I-D.ietf-teas-yang-l3-te-topo]  
Liu, X., Bryskin, I., Beeram, V. P., Saad, T., Shah, H.,  
and O. G. D. Dios, "YANG Data Model for Layer 3 TE  
Topologies", Work in Progress, Internet-Draft, draft-ietf-  
teas-yang-l3-te-topo-12, 24 October 2021,  
<[https://www.ietf.org/archive/id/draft-ietf-teas-yang-l3-  
te-topo-12.txt](https://www.ietf.org/archive/id/draft-ietf-teas-yang-l3-te-topo-12.txt)>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger,  
"Common YANG Data Types for the Routing Area", RFC 8294,  
DOI 10.17487/RFC8294, December 2017,  
<<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams",  
BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018,  
<<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and  
O. Gonzalez de Dios, "YANG Data Model for Traffic  
Engineering (TE) Topologies", RFC 8795,  
DOI 10.17487/RFC8795, August 2020,  
<<https://www.rfc-editor.org/info/rfc8795>>.

### 8.2. Informative References

[I-D.ietf-teas-yang-te-mpls]

Saad, T., Gandhi, R., Liu, X., Beeram, V. P., and I. Bryskin, "A YANG Data Model for MPLS Traffic Engineering Tunnels", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-mpls-03, 9 March 2020, <<https://www.ietf.org/archive/id/draft-ietf-teas-yang-te-mpls-03.txt>>.

[RFC5921] Bocci, M., Ed., Bryant, S., Ed., Frost, D., Ed., Levrau, L., and L. Berger, "A Framework for MPLS in Transport Networks", RFC 5921, DOI 10.17487/RFC5921, July 2010, <<https://www.rfc-editor.org/info/rfc5921>>.

#### Acknowledgments

We thank Loa Andersson for providing useful suggestions for this draft.

This document was prepared using kramdown.

Previous versions of this document was prepared using 2-Word-v2.0.template.dot.

#### Contributors

Haomian Zheng  
Huawei Technologies  
Email: [zhenghaomian@huawei.com](mailto:zhenghaomian@huawei.com)

Vishnu Pavan Beeram  
Juniper Networks  
Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

Igor Bryskin  
Individual  
Email: [i\\_bryskin@yahoo.com](mailto:i_bryskin@yahoo.com)

Yanlei Zheng  
China Unicom  
Email: [zhengyanlei@chinaunicom.cn](mailto:zhengyanlei@chinaunicom.cn)

#### Authors' Addresses



Italo Busi  
Huawei Technologies  
Email: italo.busi@huawei.com

Aihua Guo  
Futurewei Inc.  
Email: aihuaguo.ietf@gmail.com

Xufeng Liu  
Volta Networks  
Email: xufeng.liu.ietf@gmail.com

Tarek Saad  
Juniper Networks  
Email: tsaad@juniper.net

Rakesh Gandhi  
Cisco Systems, Inc.  
Email: rgandhi@cisco.com

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 13, 2022

LM. Contreras  
Telefonica  
S. Homma  
NTT  
J. Ordonez-Lucena  
Telefonica  
J. Tantsura  
Microsoft  
K. Szarkowicz  
Juniper Networks  
July 12, 2021

IETF Network Slice Use Cases and Attributes for Northbound Interface of  
IETF Network Slice Controllers  
draft-contreras-teas-slice-nbi-05

Abstract

This document analyses the needs of potential customers of network slices realized with IETF techniques in several use cases, identifies the functionalities for the North Bound Interface (NBI) of an IETF Network Slice Controller to satisfy such requests.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 13, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 3  |
| 2. Conventions used in this document and terminology . . . . .                             | 3  |
| 3. Northbound Interface for IETF Network Slices . . . . .                                  | 4  |
| 4. IETF Network Slice Use Cases . . . . .  | 5  |
| 4.1. 5G Services . . . . .   | 5  |
| 4.1.1. 3GPP network slice . . . . .  | 6  |
| 4.1.1.1. Topology of the TN-NSS . . . . .  | 6  |
| 4.1.1.2. Traffic segregation and mapping to S-NSSAI list .                                 | 7  |
| 4.1.1.3. Reachability information . . . . .  | 10 |
| 4.1.1.4. QoS profiling . . . . .   | 10 |
| 4.1.2. Generic network Slice Template . . . . .  | 10 |
| 4.1.3. Categorization of GST attributes . . . . .  | 11 |
| 4.1.3.1. Attributes with direct impact on the IETF network<br>slice definition . . . . .   | 12 |
| 4.1.3.2. Attributes with indirect impact on the IETF<br>network slice definition . . . . . | 12 |
| 4.1.3.3. Attributes with no impact on the IETF network<br>slice definition . . . . .       | 13 |
| 4.1.4. Provisioning procedures . . . . .   | 14 |
| 4.2. NFV-based services . . . . .  | 14 |
| 4.2.1. Connectivity attributes . . . . .   | 15 |
| 4.2.2. Provisioning procedures . . . . .   | 15 |
| 4.3. Network sharing . . . . .   | 16 |
| 4.3.1. Connectivity attributes . . . . .   | 17 |
| 4.3.2. Provisioning procedures . . . . .   | 17 |
| 4.4. SD-WAN . . . . .  | 17 |
| 4.4.1. SD-WAN Structure . . . . .  | 18 |
| 4.4.2. Connectivity Attributes . . . . .   | 19 |
| 4.4.3. SD-WAN Endpoint Attributes . . . . .  | 21 |
| 4.4.4. SD-WAN UNI Attributes . . . . .   | 21 |
| 4.5. Radio functional splits . . . . .   | 22 |
| 4.5.1. Attributes and procedures . . . . .   | 23 |
| 4.6. Additional use cases . . . . .  | 23 |
| 5. Security Considerations . . . . .   | 23 |
| 6. IANA Considerations . . . . .   | 23 |
| 7. References . . . . .  | 23 |
| 7.1. Normative References . . . . .  | 23 |
| 7.2. Informative References . . . . .  | 23 |

|                    |    |
|--------------------|----|
| Authors' Addresses | 24 |
|--------------------|----|

## 1. Introduction

A number of new technologies, such as 5G, NFV and SDN are not only evolving the network from a pure technological perspective but also are changing the concept in which new services are offered to the customers [I-D.homma-slice-provision-models] by introducing the concept of network slicing.

The transport network is an essential component in the end-to-end delivery of services and, consequently, it is necessary to understand what could be the way in which the transport network is consumed as a slice. For a definition of IETF network slice refer to [I-D.ietf-teas-ietf-network-slice-definition].

In this document it is assumed that there exists a (logically) centralized component in the transport network, namely IETF Network Slice Controller (NSC) with the responsibilities on the control and management of the IETF network slices invoked for a given service, as requested by IETF network slice customers.

This document analyses different use cases deriving the needs of potential IETF network slice customers in order to identify the functionality required on the North Bound Interface (NBI) of the NSC to be exposed towards such IETF network slice customers. Solutions to construct the requested IETF network slices are out of scope of this document.

This document addresses some of the discussions of the TEAS Slice Design Team. However, it is not at this stage an official outcome of the Design Team.

## 2. Conventions used in this document and terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [RFC2119].

The terminology in this draft will be aligned in forthcoming versions with the final terminology selected for describing the notion of IETF network slice when applied to IETF technologies, which is currently under discussion. By now same terminology as used in [I-D.ietf-teas-ietf-network-slice-definition] and [I-D.nsdtd-teas-ns-framework] is primarily used here.

The term "transport network" in the context of this draft refers in broad sense to WAN, MBH, IP backbone and other network segments implemented by IETF technologies.

### 3. Northbound Interface for IETF Network Slices

In a general manner, the transport network supports different kinds of services. These services consume capabilities provided by the transport network for deploying end-to-end services, interconnecting network functions or applications spread across the network and providing connectivity toward the final users of these services.

Under the slicing approach, a IETF network slice customer requests to a IETF network slice controller a slice with certain characteristics and parametrization. Such request it is assumed here to be done through a NBI exposed by the NSC to the customer, as reflected in Figure 1.

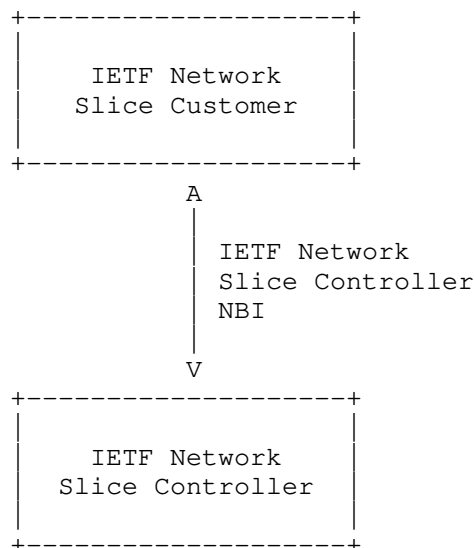


Figure 1: IETF network slice NBI concept

The functionality supported by the NBI depends on the requirements that the slice customer has to satisfy. It is then important to understand the needs of the slice customers as well as the way of expressing them.

#### 4. IETF Network Slice Use Cases

Different use cases for slice customers can be identified, as described in the following sections.

##### 4.1. 5G Services

5G services natively rely on the concept of network slicing. 5G is expected to allow vertical customers to request slices in such a manner that the allocated resources and capabilities in the network appear as dedicated for them.

In network slicing scenarios, a vertical customer requests a network operator to allocate a network slice instance (NSI) satisfying a particular set of service requirements. The content/format of these requirements are highly dependent on the networking expertise and use cases of the customer under consideration. To deal with this heterogeneity, it is fundamental for the network operator to define a unified ability to interpret service requirements from different vertical customers, and to represent them in a common language, with the purposes of facilitating their translation/mapping into specific slicing-aware network configuration actions. In this regard, model-based network slice descriptors built on the principles of reproducibility, reusability and customizability can be defined for this end.

As a starting point for such a definition, GSMA developed the idea of having a universal blueprint that, being offered by network operators, can be used by any vertical customer to order the deployment of an NSI based on a specific set of service requirements. The result of this work has been the definition of a baseline network slice descriptor called Generic network Slice Template (GST). The GST contains multiple attributes that can be used to characterize a network slice. A Network Slice Type (NEST) describes the characteristics of a network slice by means of filling GST attributes with values based on specific service requirements. Basically, a NEST is a filled-in version of a GST. Different NESTs allow describing different types of network slices. For slices based on standardized service types, e.g. eMBB, uRLLC and mMTC, the network operator may have a set of readymade, standardized NESTs (S-NESTs). For slices based on specific industry use cases, the network operator can define additional NESTs.

Service requirements from a given vertical customer are mapped to a NEST, which provides a self-contained description of the network slice to be provisioned for that vertical customer. According to this reasoning, the NEST can be used by the network operator as input to the NSI preparation phase, which is defined in [TS28.530]. 3GPP is

working on the translation of the GST/NEST attributes into NSI related requirements, which are defined in the "ServiceProfile" data type from the Network Slice Information Object Class (IOC) in [TS28.541]. These requirements are used by the 3GPP Management System to allocate the NSI across all network domains, including transport network. The IETF network slice defines the part of that NSI that is deployed across the transport network.

Despite the translation is an on-going work in 3GPP it seems convenient to start looking at the GST attributes to understand what kind of parameters could be required for the IETF network slice NBI.

#### 4.1.1.1. 3GPP network slice

A 3GPP network slice represents a logical network that provides specific capabilities and network characteristics, supporting the service requirements of one or more network slice customers. The service requirements of each network slice customer are captured into a separate "ServiceProfile" artifact within the network slice class (see Network Slicing NRM fragment in TS 28.541).

A 3GPP network slice spans from 5G NR access nodes to the UPF that terminates the PDU session, i.e. PSA UPF. In this in-slice data path, there are TN segments (e.g. backhaul) that are out of scope of 3GPP management domain. For the provisioning and operation of these TN segments, usually referred to as transport Network Slice Subnets (TN-NSS), the 3GPP management system relies on an external TN management system, which hosts (among other components) the IETF NSC. To proceed with this delegation, the 3GPP management system needs to make available to the TN management system the information described in the following sub-sections.

##### 4.1.1.1.1. Topology of the TN-NSS

The TN management system needs to know the transport termination/end points to determine the transport resources, either physical or virtual nodes. 3GPP management system systems need to provide the transport endpoints of 3GPP managed functions that are part of the RAN-NSS (e.g., gNB-CU-UP, gNB-CU-CP) and CN-NSS (e.g., UPF, AMF), and if applicable further information such as the next-hop router IP address configured in a RAN-NSS or CN-NSS. The TN management system should be able to correlate this with the transport network topology and derive the site or border routers connecting to 3GPP managed functions.

#### 4.1.1.2. Traffic segregation and mapping to S-NSSAI list

As network functions can be shared by many network slices, it will be necessary to segregate the traffic belonging to specific slices on transport interfaces.

One option for traffic segregation is to assign application endpoints to specific sets of S-NSSAI values. The transport network can map packets to connectivity services based on local remote or remote endpoints, provided that the allocation of S-NSSAI to endpoints is known and exposed, and provided that the application endpoints are visible on the transport layer. The application endpoints visible in a RAN-NSS and CN-NSS are already mapped to a specific set of S-NSSAI. Figure 2 illustrates an example of this solution, whereby a 3GPP network slice with S-NSSAI=1 is mapped to specific application endpoints (e.g., N3 tunnel endpoint 1) by the access network node. In this example, the TN management system decides to map application endpoints 1 and 2 to the same transport connectivity service A. This mapping is implemented by the site router connecting to the access network node. On the core network slice, a similar mapping is done by the border router. Demultiplexing the packet streams belonging to different transport interfaces is based on regular routing and reachability of endpoint IP addresses.



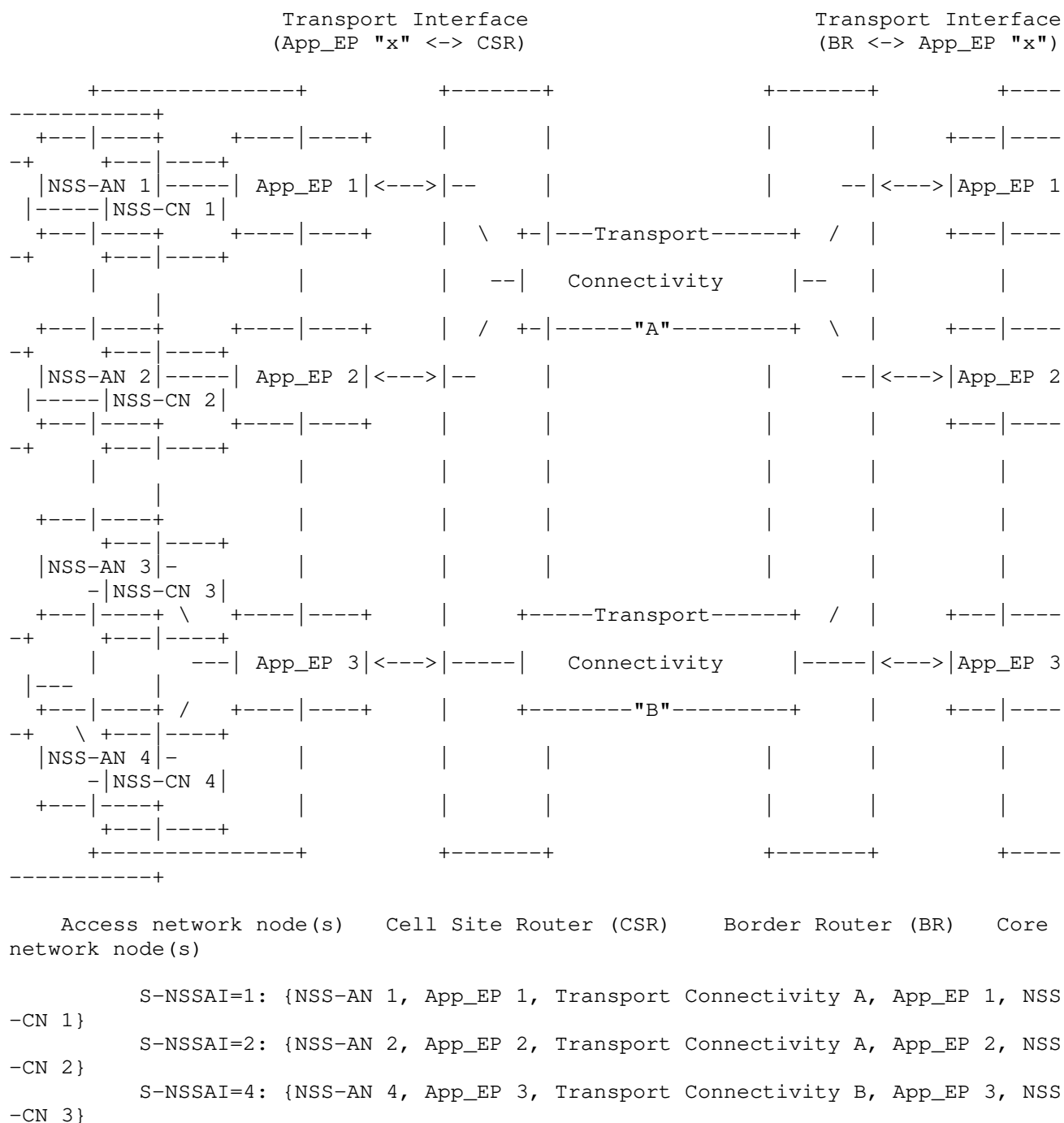
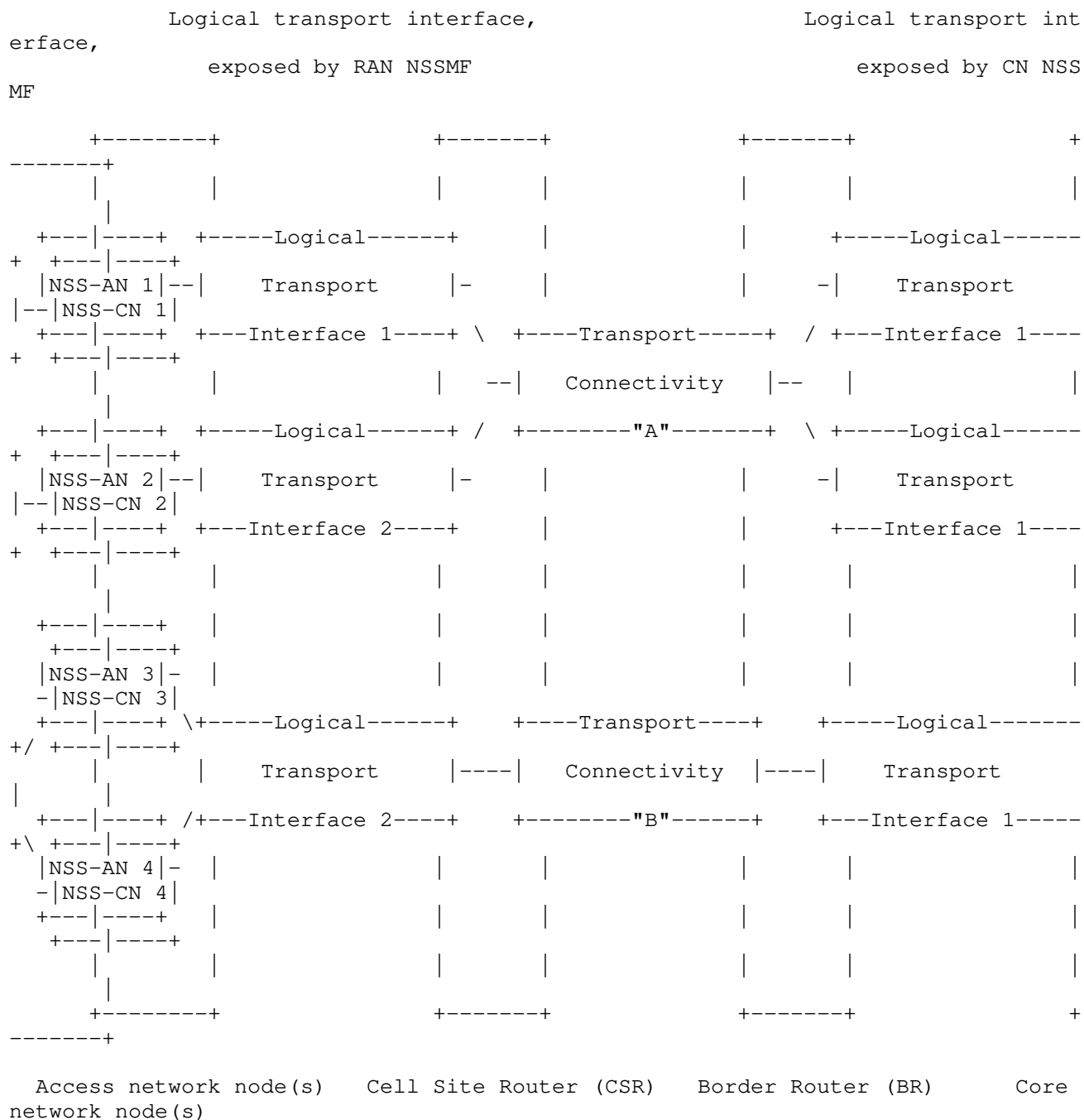


Figure 2: Mapping of S-NSSAI to specific application endpoints

Despite the simplicity of the above-referred approach, notice that it is not a universal solution as the application endpoint addresses are not always visible to the TN, for example when they are encrypted by IPSec tunnels. In such a case, the application endpoints are not visible to the site router, and thus cannot be used for transport connectivity mappings. To deal with these situations, an alternative solution is to use the concept of logical transport interfaces. A logical transport interface is a virtual interface separate from

application endpoints; it can be for example a specific IP address / VLAN combination that corresponds to an IPSec termination point, an identifier (e.g., MPLS label, segment ID) that the TN recognizes, or it can be just a logical interface defined on top of top a physical transport interface. As long as the interface identity can be derived from packet headers, the TN nodes can perform the mapping to transport connectivity services. In this regard, it is useful to indicate to the TN which traffic types are carried over an interface (e.g., N3 user plane packets, N2 control plane packets, etc.).

Figure 3 illustrates an example on the use of this solution. As seen, logical transport need to be exposed from 3GPP management system to TN management system, so that the latter can create transport network topology and determine the TN resources to support the 3GPP slice.



S-NSSAI=1: {NSS-AN 1, Logical Transport Interface 1, Transport Connectivity A, Logical Transport Interface 1, NSS-CN 1}

S-NSSAI=2: {NSS-AN 2, Logical Transport Interface 2, Transport Connectivity A, Logical Transport Interface 2, NSS-CN 2}

S-NSSAI=4: {NSS-AN 4, Logical Transport Interface 3, Transport Connectivity B, Logical Transport Interface 3, NSS-CN 4}

Figure 3: Logical Transport Interfaces

For traffic segregation, though solutions might be valid, 3GPP prefers the second solution: on the use of concept of transport logical interface. The reason is that it does not impose 1:1 mapping between application endpoint and transport interface (allowing for better redundancy) and that it always works, no matter if encryption. To support this solution, the 3GPP has recently extended the Network Slice NRM fragment, including a new Information Object Class called EP\_Transport. This class provides a complete characterization of the logical transport interface, including transport level information (i.e., IP address, reachability information, QoS profile) and the set

of application endpoints aggregated to this interface. For further information on reachability information and QoS profile, see next subsections. For further details on fields of EP\_Transport, see Network Slice NRM fragment in TS 28.541.

#### 4.1.1.3. Reachability information

Each physical or logical transport interface will carry the traffic associated with some 3GPP application endpoints that may be using IP addresses separate from the transport interface. These IP addresses must be reachable within the TN-NSS, and hence they need to be advertised to populate forwarding tables. A 3GPP network function can advertise such reachability information by running a dynamic routing protocol towards the next hop router. If that is not possible, it can create association between the reachability data with the logical transport interface and expose it towards the 3GPP and TN management system. This information can be derived from the IP addresses available for application and transport endpoints.

#### 4.1.1.4. QoS profiling

Each TN-NSS may be associated a "TNSliceSubnetProfile", which hosts the SLO requirements (e.g., guaranteed throughput, bounded latency, maximum jitter) that the TN-NSS must support. "TNSliceSubnetProfile" is a 3GPP artifact that result from the decomposition of e2e service requirements ("ServiceProfile" artifact ) into domain-specific service requirements ("RANSliceSubnetProfile", "CNSliceSubnetProfile" and "TNSliceSubnetProfile") applicable to RAN-NSS, CN-NSS and TN-NSS respectively. Unlike "RANSliceSubnetProfile" and "CNSliceSubnetProfile", there is not agreement yet on the specific parameters to be captured by the "TNSliceSubnetProfile". Further work in this regard in the upcoming 3GPP SA5 meetings.

Upon receiving the "TNSliceSubnetProfile" from the 3GPP management system, the TN management system translates the SLO requirements therein into a QoS profile, which includes applicability and use of DSCPs and other QoS related properties onto the TN-NSS realization. To enable this, each logical interface may have an associated QoS profile. The QoS profile is just a reference to the detailed profile parameters which are logically provisioned on both sides of a logical transport interface.

#### 4.1.2. Generic network Slice Template

The structure of the GST is defined in [GSMA]. The template defines a total of 35 attributes. For each of them, the following information is provided:

- o Attribute definition, which provides a formal definition of what the attribute represents.
- o Attribute parameters, including:
  - \* Value, e.g. integer, float.
  - \* Measurement unit, e.g. milliseconds, Gbps
  - \* Example, which provides examples of values the parameter can take in different use cases.
  - \* Tag, which allow describing the type of parameter, according to its semantics. An attribute can be tagged as a characterization attribute or a scalability attribute. If it is characterization attribute, it can be further tagged as a performance-related attribute, a functionality-related attribute or an operation-related attribute.
  - \* Exposure, which allow describing how this attribute interact with the slice customer, either as an API or a KPI.
- o Attribute presence, either mandatory, conditional or optional.

Attributes from GST can be used by the network operator (slice controller) and a vertical customer (slice customer) to agree SLA.

GST attributes are generic in the sense that they can be used to characterize different types of network slices. Once those attributes become filled with specific values, it becomes a NEST which can be ordered by slice customers.

#### 4.1.3. Categorization of GST attributes

Not all the GST attributes as defined in [GSMA] have impact in the transport network since some of them are specific to either the radio or the mobile core part.

In the analysis performed in this document, the attributes have been categorized as:

- o Directly impactful attributes, which are those that have direct impact on the definition of the IETF network slice, i.e., attributes that can be directly translated into requirements required to be satisfied by a IETF network slice.
- o Indirectly impactful attributes, which are those that impact in an indirect manner on the definition of the IETF network slice, i.e.,

attributes that indirectly impose some requirements to a IETF network slice.

- o Non-impactive attributes, that are those which do not have impact on the IETF network slice at all.

The following sections describe the attributes falling into the three categories.

#### 4.1.3.1. Attributes with direct impact on the IETF network slice definition

The following attributes impose requirements in the IETF network slice

- o Availability
- o Deterministic communication
- o Downlink throughput per network slice
- o Energy efficiency
- o Group communication support
- o Isolation level
- o Maximum supported packet size
- o Mission critical support
- o Performance monitoring
- o Slice quality of service parameters
- o Support for non-IP traffic
- o Uplink throughput per network slice
- o User data access (i.e., tunneling mechanisms)

#### 4.1.3.2. Attributes with indirect impact on the IETF network slice definition

The following attributes indirectly impose requirements in the IETF network slice to support the end-to-end service.

- o Area of service (i.e., the area where terminals can access a particular network slice)
- o Delay tolerance (i.e., if the service can be delivered when the system has sufficient resources)
- o Downlink (maximum) throughput per UE
- o Network functions owned by Network Slice Customer
- o Maximum number of (concurrent) PDU sessions
- o Performance prediction (i.e., capability to predict the network and service status)
- o Root cause investigation
- o Session and Service Continuity support
- o Simultaneous use of the network slice
- o Supported device velocity
- o UE density
- o Uplink (maximum) throughput per UE
- o User management openness (i.e., capability to manage users' network services and corresponding requirements)
- o Latency from (last) UPF to Application Server

#### 4.1.3.3. Attributes with no impact on the IETF network slice definition

The following attributes do not impact the IETF network slice.

- o Location based message delivery (not related to the geographical spread of the network slice itself but with the localized distribution of information)
- o MMTel support, i.e. support of and Multimedia Telephony Service (MMTel) as well as IP Multimedia Subsystem (IMS) support.
- o NB-IoT Support, i.e., support of NB-IoT in the RAN in the network slice.
- o Maximum number of (simultaneous) UEs



- o Positioning support
- o Radio spectrum
- o Synchronicity (among devices)
- o V2X communication mode
- o Network Slice Specific Authentication and Authorization (NSSAA)

#### 4.1.4. Provisioning procedures

3GPP identifies in [TS28.541] a number of procedures for the provisioning of a network slice in general. It can be assumed that similar procedures may also apply to a transport slice, facilitating a consistent management and control of end-to-end slices.

The envisioned procedures are the following:

- o Slice instance allocation: this procedure permits to create a new slice instance (or reuse an existing one).
- o Slice instance de-allocation: this procedure decommissions a previously instantiated slice.
- o Slice instance modification: this procedure permits the change in the characteristics of an existing slice instance.
- o Get slice instance status: this procedure helps to retrieve run-time information on the status of a deployed slice instance.
- o Retrieval of slice capabilities: this procedure assists on getting information about the capabilities (e.g. maximum latency supported).

All these procedures fit in the operation of transport network slices.

#### 4.2. NFV-based services

NFV technology allows the flexible and dynamic instantiation of virtualized network functions (and their composition into network services) on top of a distributed, cloud-enabled compute infrastructure. This infrastructure can span across different points of presence in a carrier network. By leveraging on transport network slicing, connectivity services established across geographically remote points of presence can be enriched by providing additional QoS

guarantees with respect present state-of-the-art mechanisms, as conventional L2/L3 VPNs.

#### 4.2.1. Connectivity attributes

The connectivity services are expressed through a number of attributes as listed:

- o Incoming and outgoing bandwidth: bandwidth required for the connectivity services (in Mbps).
- o Qos metrics: set of metrics (e.g., cost, latency and delay variation) applicable to a specific connectivity service
- o Directionality: indication if the traffic is unidirectional or bidirectional.
- o MTU: value of the largest PDU to be transmitted in the connectivity service.
- o Protection scheme: indication of the kind of protection to be performed (e.g., 1;1, 1+1, etc.)
- o Connectivity mode: indication of the service is point-to-point or point-to-multipoint

All those attributes will assist on the characterization of the connectivity slice to be deployed, and thus, are relevant for the definition of a IETF network slice supporting such connectivity.

#### 4.2.2. Provisioning procedures

ETSI NFV defines the role of WAN Infrastructure Manager (WIM) as the component in charge of managing and controlling the connectivity external to the PoPs. In [IFA032] a number of interfaces are identified to be exposed by the WIM for supporting the multi-site connectivity, thus representing the capabilities expected for a transport network slice, as well, in case of satisfying such connectivity needs by means of the slice concept.

The interfaces considered are the following:

- o Multi-Site Connectivity Service (MSCS) Management: this interface permits the creation, termination, update and query of MSCSs, including reservation. It also enables subscription for notifications and information retrieval associated to the connectivity service.

- o Capacity Management: this interface allows querying about the capacity (e.g. bandwidth), topology, and network edge points of the connectivity service, as well as about information of consumed and available capacity on the underlying network resources.
- o Fault Management: this interface serves for the provision of alarms related to the MSCSs.
- o Performance Management: this interface assists on the retrieval of performance information (measurement results collection and notifications) related to MSCSs.

#### 4.3. Network sharing

Network sharing is one of the means network operators exploit for increasing efficiencies. There are different scenarios of network sharing, being especially popular in the deployment of mobile networks, typically referred to as Radio Access Network (RAN) sharing. From an operational perspective, in RAN sharing we have two roles: master operator, being the actor (e.g. infrastructure provider, network operator) to which the deployment and daily operation of shared RAN elements are entrusted to; and the participant operators, who are the mobile operators who share the RAN facilities provided by the master operator. Note that in this context the master and participant operator can be seen as provider and customer, respectively.

While there exist different modes of RAN sharing [TS23.251], including passive RAN sharing (infrastructure site sharing) and active RAN sharing (e.g. Multi-Operator Core Networks or MOCN), most of the cases require the establishment of separated connections in order to separate the traffic per participant operator. Such connections typically extend from the cell site to some pre-defined and agreed interconnection points, from which the traffic is routed and delivered to individual participant operators.

The above-referred connections can have specific attributes. Aspects like guaranteed bandwidth (in line with the expected load from the aggregated cells), redundancy, bounded latency (per kind of traffic), or secure delivery of the information should be considered.

The master operator is the one in charge of provisioning the connections and collecting management data (e.g. performance measurements, telemetry, fault alarms, trace data) for individual participant operators. The use of network slicing could make the network sharing approach more flexible by allowing the other operators control and manage the established connections [MEF].

The implications of the RAN sharing scenario here described can be extended to either fixed networks or even to mobile networks leveraging on radio functional split (i.e., including fronthaul and midhaul network segments).

#### 4.3.1. Connectivity attributes

The connections for RAN sharing typically consider attributes like:

- o Maximum and Guaranteed Bit Rate (MBR and GBR respectively).
- o Bounded latency (e.g., for user plane, control plane, etc)
- o Packet loss rate.
- o IP addressing (consistent among the operators sharing the infrastructure).
- o L2/L3 reachability.
- o Recovery time (on the event of failures).
- o Secure connection (e.g., encryption support).

#### 4.3.2. Provisioning procedures

The expected provisioning procedures are:

- o Connection provisioning between site and interconnection point. Those connections could evolve in time in terms of capacity depending on the capacity growth of each particular site.
- o Collection of management data, including performance measurements, fault alarms and trace data.

#### 4.4. SD-WAN

SD-WAN is a solution to provide a virtual overlay network for connecting between customer's sites, (virtual) private cloud, or public cloud/Internet. SD-WAN operates over one or more underlay networks, and enables to offer more differentiated service delivery capabilities. SD-WAN can be esteemed as a type of network slices or can be established over underlay networks provided as network slices. The definitions, specification, service attributes, and framework of SD-WAN is defined in Metro Ethernet Forum ([MEF-70]).

SD-WAN forwards traffic based on application flows, and the policies include rules and constraints on the forwarding of the application

flows. In SD-WAN, it may be required from the customer to adjust the behaviors based on its needs in near real time. The service provider is required to monitor the performance of the service and modify the forwarding policies based on the real-time telemetry from the underlying network components.

#### 4.4.1. SD-WAN Structure

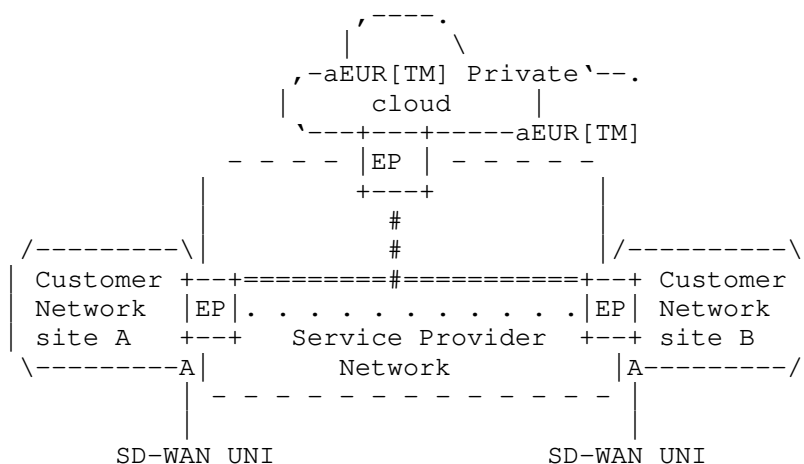
SD-WAN has three logical constructs:

- o SD-WAN virtual connection
- o SD-WAN virtual connection endpoint
- o SD-WAN UNI

Several additional components may be visible to the customer. These include:

- o Customer network
- o Service provider network
- o Underlay connectivity
- o Tunnel virtual connection

The following figure shows the overview of SD-WAN structure. In this case, the customer sites are connected with underlay connectivity#1 and they are also connected to remote private cloud with underlay connectivity#2. An SD-WAN endpoint is usually located in each customer network site as a CPE or a customer edge, and it allocates application flow to appropriate underlay connectivity.



\* Legend

```

. . . : Underlay connectivity#1

```

```
===== : Underlay Connectivity#2
```

EP : SD-WAN Endpoint

Figure 4: Overview of SD-WAN Structure

SD-WAN may be provided as a network slice, or it is realized on several network slices provided as underlay connectivities. In the former case, a network slice PE will be mapped to CE in SD-WAN. In the later case, PEs of the provider of underlay connectivities will behave as network slice PEs.

#### 4.4.2. Connectivity Attributes

SD-WAN defined in MEF-70 has several attributes on its connectivity as below:

- o SD-WAN Identifier: the value is a string that is used by the customer and service provider to uniquely identify an SD-WAN connectivity.
- o Endpoint list: the value is a list contains endpoint identifiers and their connected endpoints.
- o Service Uptime Objective: the value is the proportion of time that the connectivity service is working during a given time period.

- o **Reserved Prefixes:** the values are IP prefixes reserved by the service provider for use for SD-WAN within its own network or for distribution to the customer via DHCP or SLAAC.
- o **List for Policies:** the value is a list of policies applied to application flows and application flow groups at endpoints. An SD-WAN policy list contains policy name and list of policy criteria. Support of the criteria listed below would be required:
  - \* **Encryption:** indicates whether or not the application flow requires encryption
  - \* **Public-Private:** indicates whether the application flow can traverse public or private underlay connectivity services (or both).
  - \* **Internet-Breakout:** indicates whether the application flow should be forwarded to an Internet destination.
  - \* **Billing-Method:** indicate the application flow can be sent over an underlay connectivity service that has usage-based or flat-rate billing.
  - \* **Backup:** indicates whether this application flow can use a TVC designated as aEUR&#157;backupaEUR&#157;.
  - \* **Bandwidth:** specifies a rate limit on the application flow.
- o **List of Application Flow Groups:** the value is a list of application flow groups that application flows can be members of. An application flow group list contains application flow group name and application flow group policy.
- o **List of Application Flows:** the value is a list of the application flows that are recognized by the SD-WAN. An application flow list contains application flow name, list of application flow criteria, and application flow group name. The criteria is listed below:
  - \* **Ethertype**
  - \* **C-VLAN ID list**
  - \* **IPv4 source address**
  - \* **IPv4 destination address**
  - \* **IPv4 source or destination address**

- \* IPv4 protocol list
- \* IPv6 source address
- \* IPv6 destination address
- \* IPv6 source or destination address
- \* IPv6 next header list
- \* TCP/UDP source port list
- \* TCP/UDP destination port list
- \* Application identifier
- \* any

#### 4.4.3. SD-WAN Endpoint Attributes

SD-WAN contains some endpoints as boundary nodes between underlay connections and customers sites. [MEF-70] defines some attributes for SD-WAN endpoints as below:

- o Endpoint Identifier: the value is for identification of SD-WAN endpoint for management purposes.
- o Endpoint UNI: the value is for identification of the UNI that the endpoint is associated with.
- o Endpoint policy map: the value is for mapping policies to application flows and application flow groups.

#### 4.4.4. SD-WAN UNI Attributes

SD-WAN UNI is a reference point that represents the demarcation between the responsibility of the customer and the responsibility of the provider. Some attributes for UNI is defined in [MEF-70] as below:

- o SD-WAN UNI Identifier: the value is for identification of the UNI for management purposes.
- o SD-WAN UNI L2 Interface: the value describes the underlay L2 interface for the UNI.
- o SD-WAN UNI Maximum L2 Frame Size: the value specifies the maximum length L2 frame that is accepted by the provider.



- o SD-WAN UNI IPv4 connection addressing: the value describes IPv4 connection address mechanisms (e.g., Static or DHCP).
- o SD-WAN UNI IPv6 connection addressing: the value describes IPv6 connection address mechanisms (e.g., DHCP, SLAAC, Static or Link-Local-only).

#### 4.5. Radio functional splits

The disaggregation of the software stack in radio base stations allows the centralization of some of the radio processing functions. O-RAN is promoting the interoperability of implementations of radio functional splits, defining an architecture where three main entities can be considered: the Radio Unit (RU), with some basic processing, the Distributed Unit (DU) with the rest of real-time processing capabilities, and the Centralized Unit (CU) with the non-real-time processing of the software stack. The network segment between RU and DU is known as fronthaul (FH), while the segment between DU and CU is referred as midhaul (MH). Figure 5 shows this situation.

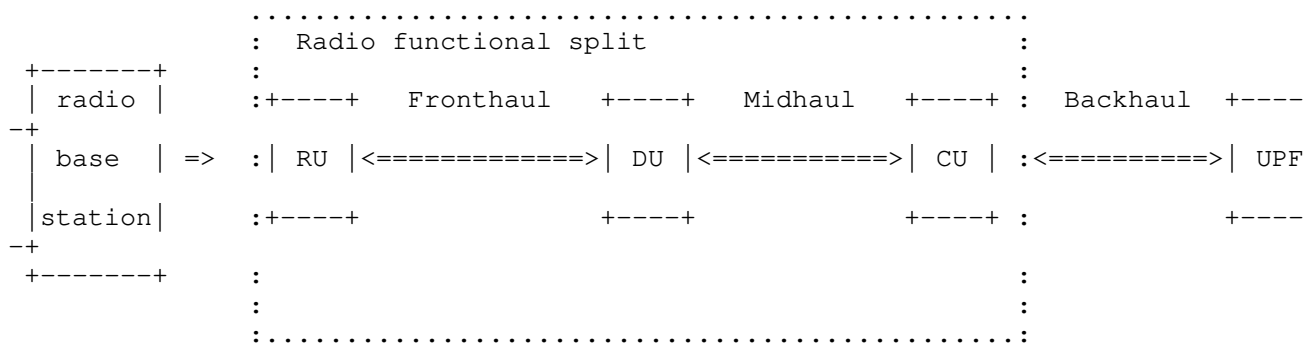


Figure 5: Logical Transport Interfaces

The fronthaul leverages on eCPRI protocol which can be transported directly on Ethernet frames or encapsulated in IP/UDP (for the user plane). The midhaul can be transported in a similar way as the backhaul.

With current specifications, individual service flows being carried by FH cannot be distinguished, so no possibility of differentiating connectivity slices at that point. Similar thing happens for MH. The only possible differentiation per flow can happen in downstream direction from CU to DU, but this basically can only help for policing traffic at that point (i.e., slice is yet the same).

Advanced scenarios such as RU sharing could allow traffic differentiation per mobile operator based on e.g. vlans, being each of those vlans mapped to a different slice.

#### 4.5.1. Attributes and procedures

The attributes of IETF network slices for the conveniently supported the radio functional split are based on main characteristics of FH/MH: Latency, BW, and packet loss, as specified in [O-RAN]. Geographical location could have an impact due to latency restrictions for FH.

Regarding slice management procedures, it can be assumed a similar lifecycle as in 3GPP slices.

#### 4.6. Additional use cases

This is a placeholder for describing additional use cases (e.g., data center interconnection, etc). To be completed.

#### 5. Security Considerations

This draft does not include any security considerations.

#### 6. IANA Considerations

This draft does not include any IANA considerations

#### 7. References

##### 7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

##### 7.2. Informative References

[GSMA] "Generic Network Slice Template, version 3.0", NG.116 , May 2020.

[I-D.homma-slice-provision-models]  
Homma, S., Nishihara, H., Miyasaka, T., Galis, A., OV, V. R., Lopez, D. R., Contreras, L. M., Ordonez-Lucena, J. A., Martinez-Julia, P., Qiang, L., Rokui, R., Ciavaglia, L., and X. D. Foy, "Network Slice Provision Models", draft-homma-slice-provision-models-02 (work in progress), November 2019.

- [I-D.ietf-teas-ietf-network-slice-definition]  
Rokui, R., Homma, S., Makhiyani, K., Contreras, L. M., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-01 (work in progress), February 2021.
- [I-D.nsd-t-teas-ns-framework]  
Gray, E. and J. Drake, "Framework for IETF Network Slices", draft-nsdt-teas-ns-framework-05 (work in progress), February 2021.
- [IFA032] "IFA032 Interface and Information Model Specification for Multi-Site Connectivity Services V3.2.1.", ETSI GS NFV-IFA 032 V3.2.1 , April 2019.
- [MEF] "Slicing for Shared 5G Fronthaul and Backhaul", MEF White paper , April 2020.
- [MEF-70] "SD-WAN Service Attributes and Services", MEF-70 , July 2019.
- [O-RAN] "O-RAN Xhaul Transport Requirements 1.0", O-RAN.WG9.XTRP-REQ-v01.00 , November 2020.
- [TS23.251]  
"TS 23.251 Network Sharing; Architecture and functional description (Release 16) V16.0.0.", 3GPP TS 23.251 V16.0.0 , July 2020.
- [TS28.530]  
"TS 28.530 Management and orchestration; Concepts, use cases and requirements (Release 16) V16.0.0.", 3GPP TS 28.530 V16.0.0 , September 2019.
- [TS28.541]  
"TS 28.541 Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 16) V16.2.0.", 3GPP TS 28.541 V16.2.0 , September 2019.

Authors' Addresses

Luis M. Contreras  
Telefonica  
Ronda de la Comunicacion, s/n  
Sur-3 building, 3rd floor  
Madrid 28050  
Spain

Email: [luismiguel.contrerasmurillo@telefonica.com](mailto:luismiguel.contrerasmurillo@telefonica.com)  
URI: <http://lmcontreras.com/>

Shunsuke Homma  
NTT  
Japan

Email: [shunsuke.homma.ietf@gmail.com](mailto:shunsuke.homma.ietf@gmail.com)

Jose A. Ordonez-Lucena  
Telefonica  
Ronda de la Comunicacion, s/n  
Sur-3 building, 3rd floor  
Madrid 28050  
Spain

Email: [joseantonio.ordonezlucena@telefonica.com](mailto:joseantonio.ordonezlucena@telefonica.com)

Jeff Tantsura  
Microsoft

Email: [jefftant.ietf@gmail.com](mailto:jefftant.ietf@gmail.com)

Krzysztof Szarkowicz  
Juniper Networks

Email: [kszarkowicz@juniper.net](mailto:kszarkowicz@juniper.net)

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 8, 2022

LM. Contreras  
Telefonica  
S. Homma  
NTT  
J. Ordonez-Lucena  
Telefonica  
J. Tantsura  
Microsoft  
H. Nishihara  
NTT  
March 7, 2022

IETF Network Slice Use Cases and Attributes for Northbound Interface of  
IETF Network Slice Controllers  
draft-contreras-teas-slice-nbi-06

Abstract

This document analyses the needs of potential customers of network slices realized with IETF techniques in several use cases, identifies the functionalities for the North Bound Interface (NBI) of an IETF Network Slice Controller to satisfy such requests.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 8, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .   | 3  |
| 2. Conventions used in this document and terminology . . . . .                          | 3  |
| 3. Northbound Interface for IETF Network Slices . . . . .                               | 4  |
| 4. IETF Network Slice Use Cases . . . . .   | 5  |
| 4.1. 5G Services . . . . .  | 5  |
| 4.1.1. 3GPP network slice . . . . .   | 6  |
| 4.1.1.1. Topology of the TN-NSS . . . . .   | 6  |
| 4.1.1.2. Traffic segregation and mapping to S-NSSAI list . . . . .                      | 7  |
| 4.1.1.3. Reachability information . . . . .   | 10 |
| 4.1.1.4. QoS profiling . . . . .  | 10 |
| 4.1.2. Private 5G networks . . . . .  | 11 |
| 4.1.2.1. Structure Patterns of Private 5G system . . . . .                              | 11 |
| 4.1.2.2. Use Cases Assumed in Private 5G . . . . .                                      | 11 |
| 4.1.2.3. Attributes Required in Private 5G . . . . .                                    | 12 |
| 4.1.3. Generic network Slice Template . . . . .   | 12 |
| 4.1.4. Categorization of GST attributes . . . . .                                       | 13 |
| 4.1.4.1. Attributes with direct impact on the IETF network slice definition . . . . .   | 14 |
| 4.1.4.2. Attributes with indirect impact on the IETF network slice definition . . . . . | 14 |
| 4.1.4.3. Attributes with no impact on the IETF network slice definition . . . . .       | 15 |
| 4.1.5. Provisioning procedures . . . . .  | 16 |
| 4.2. NFV-based services . . . . .   | 16 |
| 4.2.1. Connectivity attributes . . . . .  | 16 |
| 4.2.2. Provisioning procedures . . . . .  | 17 |
| 4.3. Network sharing . . . . .  | 18 |
| 4.3.1. Connectivity attributes . . . . .  | 18 |
| 4.3.2. Provisioning procedures . . . . .  | 19 |
| 4.4. SD-WAN . . . . .   | 19 |
| 4.4.1. SD-WAN Structure . . . . .   | 19 |
| 4.4.2. Connectivity Attributes . . . . .  | 21 |
| 4.4.3. SD-WAN Endpoint Attributes . . . . .   | 22 |
| 4.4.4. SD-WAN UNI Attributes . . . . .  | 23 |
| 4.5. Radio functional splits . . . . .  | 23 |
| 4.5.1. Attributes and procedures . . . . .  | 24 |
| 4.6. Additional use cases . . . . .   | 24 |
| 5. Summary of attributes and procedures . . . . .                                       | 25 |

|                                       |    |
|---------------------------------------|----|
| 5.1. Summary of SLOs . . . . .        | 25 |
| 5.2. Summary of SLEs . . . . .        | 25 |
| 5.3. Summary of procedures . . . . .  | 25 |
| 6. Security Considerations . . . . .  | 26 |
| 7. IANA Considerations . . . . .      | 26 |
| 8. References . . . . .               | 26 |
| 8.1. Normative References . . . . .   | 26 |
| 8.2. Informative References . . . . . | 26 |
| Authors' Addresses . . . . .          | 27 |

## 1. Introduction

A number of new technologies, such as 5G, NFV and SDN are not only evolving the network from a pure technological perspective but also are changing the concept in which new services are offered to the customers [I-D.homma-slice-provision-models] by introducing the concept of network slicing.

The transport network is an essential component in the end-to-end delivery of services and, consequently, it is necessary to understand what could be the way in which the transport network is consumed as a slice. For a definition of IETF network slice refer to [I-D.ietf-teas-ietf-network-slices].

In this document it is assumed that there exists a (logically) centralized component in the transport network, namely IETF Network Slice Controller (NSC) with the responsibilities on the control and management of the IETF network slices invoked for a given service, as requested by IETF network slice customers.

This document analyses different use cases deriving the needs of potential IETF network slice customers in order to identify the functionality required on the North Bound Interface (NBI) of the NSC to be exposed towards such IETF network slice customers. Solutions to construct the requested IETF network slices are out of scope of this document.

## 2. Conventions used in this document and terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [RFC2119].

The terminology in this draft will be aligned in forthcoming versions with the final terminology selected for describing the notion of IETF network slice when applied to IETF technologies, as defined in [I-D.ietf-teas-ietf-network-slices] .

The term "transport network" in the context of this draft refers in broad sense to WAN, MBH, IP backbone and other network segments implemented by IETF technologies.

### 3. Northbound Interface for IETF Network Slices

In a general manner, the transport network supports different kinds of services. These services consume capabilities provided by the transport network for deploying end-to-end services, interconnecting network functions or applications spread across the network and providing connectivity toward the final users of these services.

Under the slicing approach, a IETF network slice customer requests to a IETF network slice controller a slice with certain characteristics and parametrization. Such request it is assumed here to be done through a NBI exposed by the NSC to the customer, as reflected in Figure 1.

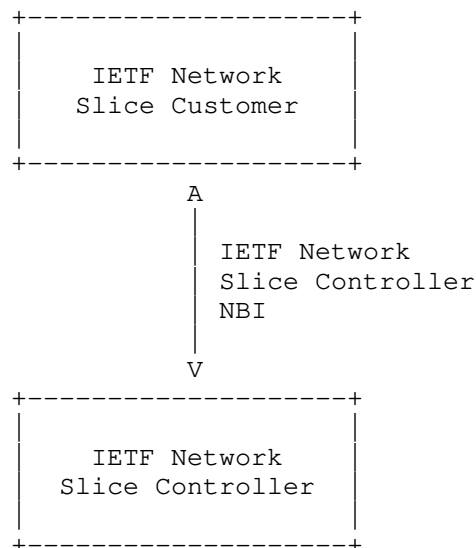


Figure 1: IETF network slice NBI concept

The functionality supported by the NBI depends on the requirements that the slice customer has to satisfy. It is then important to understand the needs of the slice customers as well as the way of expressing them.



#### 4. IETF Network Slice Use Cases

Different use cases for slice customers can be identified, as described in the following sections.

##### 4.1. 5G Services

5G services natively rely on the concept of network slicing. 5G is expected to allow vertical customers to request slices in such a manner that the allocated resources and capabilities in the network appear as dedicated for them.

In network slicing scenarios, a vertical customer requests a network operator to allocate a network slice instance (NSI) satisfying a particular set of service requirements. The content/format of these requirements are highly dependent on the networking expertise and use cases of the customer under consideration. To deal with this heterogeneity, it is fundamental for the network operator to define a unified ability to interpret service requirements from different vertical customers, and to represent them in a common language, with the purposes of facilitating their translation/mapping into specific slicing-aware network configuration actions. In this regard, model-based network slice descriptors built on the principles of reproducibility, reusability and customizability can be defined for this end.

As a starting point for such a definition, GSMA developed the idea of having a universal blueprint that, being offered by network operators, can be used by any vertical customer to order the deployment of an NSI based on a specific set of service requirements. The result of this work has been the definition of a baseline network slice descriptor called Generic network Slice Template (GST). The GST contains multiple attributes that can be used to characterize a network slice. A Network Slice Type (NEST) describes the characteristics of a network slice by means of filling GST attributes with values based on specific service requirements. Basically, a NEST is a filled-in version of a GST. Different NESTs allow describing different types of network slices. For slices based on standardized service types, e.g. eMBB, uRLLC and mMTC, the network operator may have a set of readymade, standardized NESTs (S-NESTs). For slices based on specific industry use cases, the network operator can define additional NESTs.

Service requirements from a given vertical customer are mapped to a NEST, which provides a self-contained description of the network slice to be provisioned for that vertical customer. According to this reasoning, the NEST can be used by the network operator as input to the NSI preparation phase, which is defined in [TS28.530]. 3GPP is

working on the translation of the GST/NEST attributes into NSI related requirements, which are defined in the "ServiceProfile" data type from the Network Slice Information Object Class (IOC) in [TS28.541]. These requirements are used by the 3GPP Management System to allocate the NSI across all network domains, including transport network. The IETF network slice defines the part of that NSI that is deployed across the transport network.

Despite the translation is an on-going work in 3GPP it seems convenient to start looking at the GST attributes to understand what kind of parameters could be required for the IETF network slice NBI.

#### 4.1.1.1. 3GPP network slice

A 3GPP network slice represents a logical network that provides specific capabilities and network characteristics, supporting the service requirements of one or more network slice customers. The service requirements of each network slice customer are captured into a separate "ServiceProfile" artifact within the network slice class (see Network Slicing NRM fragment in TS 28.541).

A 3GPP network slice spans from 5G NR access nodes to the UPF that terminates the PDU session, i.e. PSA UPF. In this in-slice data path, there are TN segments (e.g. backhaul) that are out of scope of 3GPP management domain. For the provisioning and operation of these TN segments, usually referred to as transport Network Slice Subnets (TN-NSS), the 3GPP management system relies on an external TN management system, which hosts (among other components) the IETF NSC. To proceed with this delegation, the 3GPP management system needs to make available to the TN management system the information described in the following sub-sections.

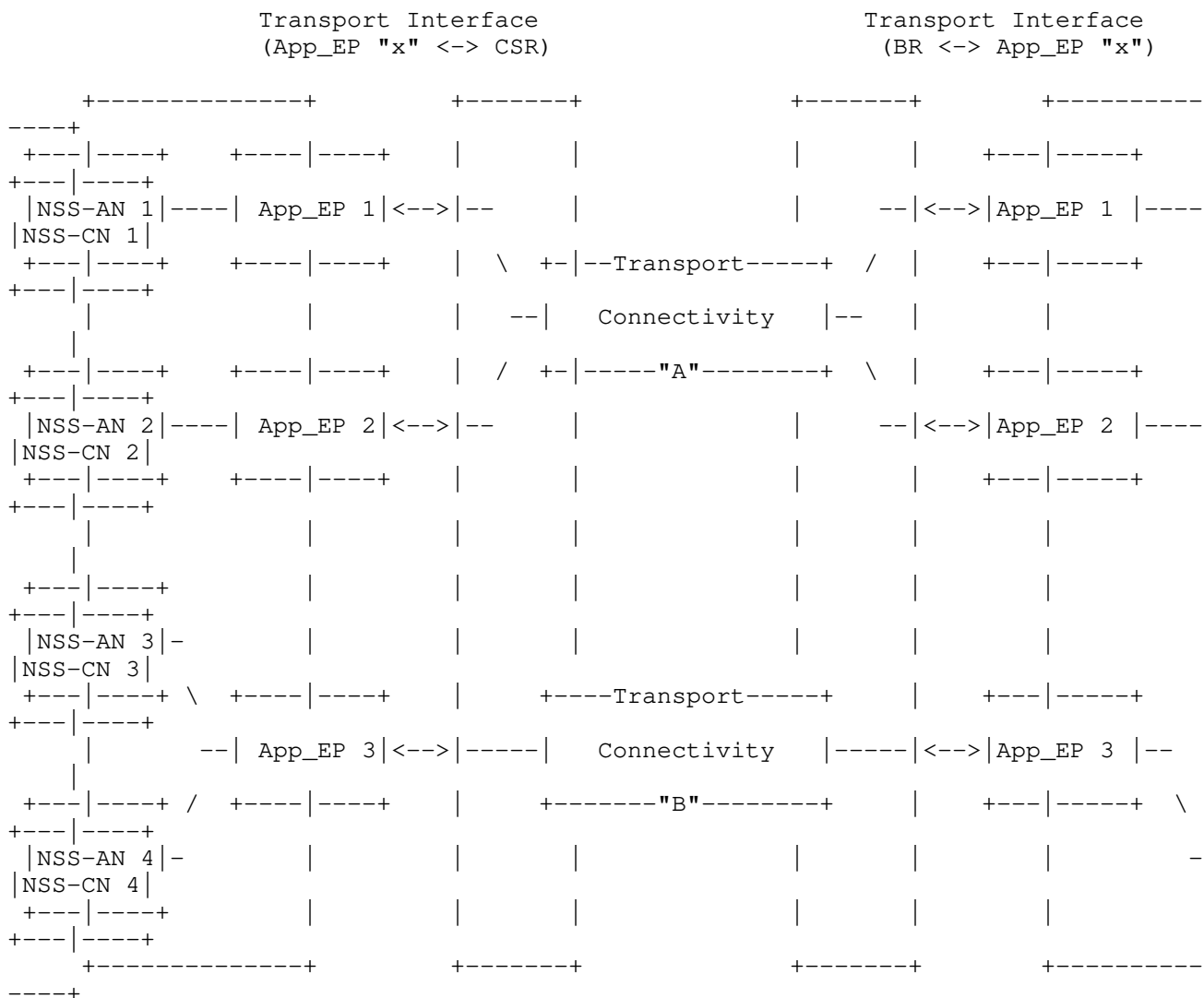
##### 4.1.1.1.1. Topology of the TN-NSS

The TN management system needs to know the transport termination/end points to determine the transport resources, either physical or virtual nodes. 3GPP management system systems need to provide the transport endpoints of 3GPP managed functions that are part of the RAN-NSS (e.g., gNB-CU-UP, gNB-CU-CP) and CN-NSS (e.g., UPF, AMF), and if applicable further information such as the next-hop router IP address configured in a RAN-NSS or CN-NSS. The TN management system should be able to correlate this with the transport network topology and derive the site or border routers connecting to 3GPP managed functions.

#### 4.1.1.2. Traffic segregation and mapping to S-NSSAI list

As network functions can be shared by many network slices, it will be necessary to segregate the traffic belonging to specific slices on transport interfaces.

One option for traffic segregation is to assign application endpoints to specific sets of S-NSSAI values. The transport network can map packets to connectivity services based on local remote or remote endpoints, provided that the allocation of S-NSSAI to endpoints is known and exposed, and provided that the application endpoints are visible on the transport layer. The application endpoints visible in a RAN-NSS and CN-NSS are already mapped to a specific set of S-NSSAI. Figure 2 illustrates an example of this solution, whereby a 3GPP network slice with S-NSSAI=1 is mapped to specific application endpoints (e.g., N3 tunnel endpoint 1) by the access network node. In this example, the TN management system decides to map application endpoints 1 and 2 to the same transport connectivity service A. This mapping is implemented by the site router connecting to the access network node. On the core network slice, a similar mapping is done by the border router. Demultiplexing the packet streams belonging to different transport interfaces is based on regular routing and reachability of endpoint IP addresses.



Access network node(s) Cell Site Router (CSR) Border Router (BR) Core network node(s)

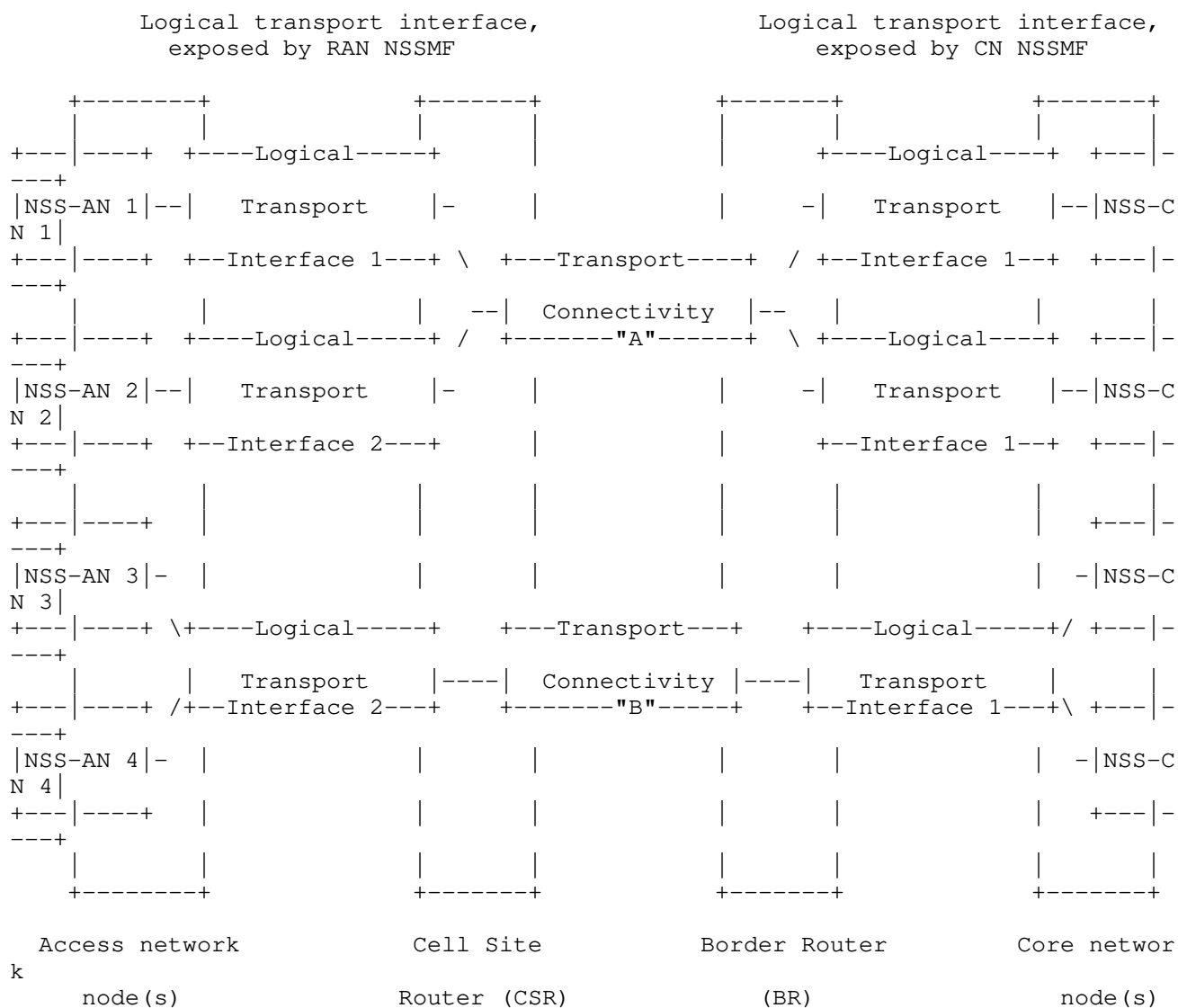
- 1} S-NSSAI=1: {NSS-AN 1, App\_EP 1, Transport Connectivity A, App\_EP 1, NSS-CN
- 2} S-NSSAI=2: {NSS-AN 2, App\_EP 2, Transport Connectivity A, App\_EP 2, NSS-CN
- 3} S-NSSAI=4: {NSS-AN 4, App\_EP 3, Transport Connectivity B, App\_EP 3, NSS-CN

Figure 2: Mapping of S-NSSAI to specific application endpoints

Despite the simplicity of the above-referred approach, notice that it is not a universal solution as the application endpoint addresses are not always visible to the TN, for example when they are encrypted by IPSec tunnels. In such a case, the application endpoints are not visible to the site router, and thus cannot be used for transport connectivity mappings. To deal with these situations, an alternative solution is to use the concept of logical transport interfaces. A logical transport interface is a virtual interface separate from

application endpoints; it can be for example a specific IP address / VLAN combination that corresponds to an IPSec termination point, an identifier (e.g., MPLS label, segment ID) that the TN recognizes, or it can be just a logical interface defined on top of top a physical transport interface. As long as the interface identity can derived from packet headers, the TN nodes can perform the mapping to transport connectivity services. In this regard, it is useful to indicate to the TN which traffic types are carried over an interface (e.g., N3 user plane packets, N2 control plane packets, etc.).

Figure 3 illustrates an example on the use of this solution. As seen, logical transport need to be exposed from 3GPP management system to TN management system, so that the latter can create transport network topology and determine the TN resources to support the 3GPP slice.



For traffic segregation, though solutions might be valid, 3GPP prefers the second solution: on the use of concept of transport logical interface. The reason is that it does not impose 1:1 mapping between application endpoint and transport interface (allowing for

better redundancy) and that it always works, no matter if encryption. To support this solution, the 3GPP has recently extended the Network Slice NRM fragment, including a new Information Object Class called

EP\_Transport. This class provides a complete characterization of the logical transport interface, including transport level information (i.e., IP address, reachability information, QoS profile) and the set of application endpoints aggregated to this interface. For further information on reachability information and QoS profile, see next subsections. For further details on fields of EP\_Transport, see Network Slice NRM fragment in TS 28.541.

#### 4.1.1.3. Reachability information

Each physical or logical transport interface will carry the traffic associated with some 3GPP application endpoints that may be using IP addresses separate from the transport interface. These IP addresses must be reachable within the TN-NSS, and hence they need to be advertised to populate forwarding tables. A 3GPP network function can advertise such reachability information by running a dynamic routing protocol towards the next hop router. If that is not possible, it can create association between the reachability data with the logical transport interface and expose it towards the 3GPP and TN management system. This information can be derived from the IP addresses available for application and transport endpoints.

#### 4.1.1.4. QoS profiling

Each TN-NSS may be associated a "TNSliceSubnetProfile", which hosts the SLO requirements (e.g., guaranteed throughput, bounded latency, maximum jitter) that the TN-NSS must support. "TNSliceSubnetProfile" is a 3GPP artifact that result from the decomposition of e2e service requirements ("ServiceProfile" artifact ) into domain-specific service requirements ("RANSliceSubnetProfile", "CNSliceSubnetProfile" and "TNSliceSubnetProfile") applicable to RAN-NSS, CN-NSS and TN-NSS respectively. Unlike "RANSliceSubnetProfile" and "CNSliceSubnetProfile", there is not agreement yet on the specific parameters to be captured by the "TNSliceSubnetProfile". Further work in this regard in the upcoming 3GPP SA5 meetings.

Upon receiving the "TNSliceSubnetProfile" from the 3GPP management system, the TN management system translates the SLO requirements therein into a QoS profile, which includes applicability and use of DSCPs and other QoS related properties onto the TN-NSS realization. To enable this, each logical interface may have an associated QoS profile. The QoS profile is just a reference to the detailed profile parameters which are logically provisioned on both sides of a logical transport interface.



#### 4.1.2. Private 5G networks

Private 5G is one of variations of 5G service provision. Private 5G allows unlicensed as well as licensed companies to establish and operate 5G networks, with frequency band assigned for private 5G, in their own companies.

Private 5G can be customized flexibly rather than public 5G, and thus it enables us to provide networks specialized for their use cases. Private 5G is also called non-public 5G, and its deployment scenarios and service attributes are described in (ref. [TS23.501]).

##### 4.1.2.1. Structure Patterns of Private 5G system

In Private 5G, a Service Provider does not necessarily have its own resources (e.g., radio bases, transit network and server resources for 5G CP functions) and can flexibly customize and deploy by selecting and combining various resources.

Private 5G has several structure patterns:

- o Pattern 1: a service provider has all resources including radio bases, transit networks, and server resources for 5G CP functions.
- o Pattern 2: a service provider has radio bases and server resources for 5G CP functions, and lends transit networks from other network operators.
- o Pattern 3: a service provider has only radio bases and lends transit networks and server resources for 5G CP functions from other network operators and data center companies.

In pattern 2 and 3, it is assumed that a service provider uses network slices provided by other companies.

##### 4.1.2.2. Use Cases Assumed in Private 5G

Private 5G provides a wireless communication environment which has specific features depending on applications or usage, within limited areas. From such aspects, within 5G use cases (ref. [TS22.261]), the following communication types and use cases could be especially expected to be provided with private 5G.

- o High-bandwidth and reliable communication:
  - \* VR streaming
- o Low latency and jitter:

- \* Smart factory
- \* Remote automated robot operation (e.g., robot concierge/assistant, robot waiter, drone)
- o High-bandwidth on up-link and low latency and jitter
  - \* Remote surgery
  - \* Uploading of high-definition video

#### 4.1.2.3. Attributes Required in Private 5G

Private 5G has some distinguished requirements to network slice as below.

- o QoS customization:
  - \* assured bandwidth
  - \* assured latency and jitter
  - \* customization of UL/DL rate on throughput (e.g., for video upstreaming consumes much UL bandwidth)
- o Multi-homing (for high reliability, preparing multiple paths traverse different physical routes)
- o Performance monitoring (e.g., for connectivity status and service availability of devices)
- o Traffic flow separation/segregation (e.g., segregation of user plane and other communications physically and/or logically)

#### 4.1.3. Generic network Slice Template

The structure of the GST is defined in [GSMA]. The template defines a total of 35 attributes. For each of them, the following information is provided:

- o Attribute definition, which provides a formal definition of what the attribute represents.
- o Attribute parameters, including:
  - \* Value, e.g. integer, float.
  - \* Measurement unit, e.g. milliseconds, Gbps

- \* Example, which provides examples of values the parameter can take in different use cases.
  - \* Tag, which allow describing the type of parameter, according to its semantics. An attribute can be tagged as a characterization attribute or a scalability attribute. If it is characterization attribute, it can be further tagged as a performance-related attribute, a functionality-related attribute or an operation-related attribute.
  - \* Exposure, which allow describing how this attribute interact with the slice customer, either as an API or a KPI.
- o Attribute presence, either mandatory, conditional or optional.

Attributes from GST can be used by the network operator (slice controller) and a vertical customer (slice customer) to agree SLA.

GST attributes are generic in the sense that they can be used to characterize different types of network slices. Once those attributes become filled with specific values, it becomes a NEST which can be ordered by slice customers.

#### 4.1.4. Categorization of GST attributes

Not all the GST attributes as defined in [GSMA] have impact in the transport network since some of them are specific to either the radio or the mobile core part.

In the analysis performed in this document, the attributes have been categorized as:

- o Directly impactful attributes, which are those that have direct impact on the definition of the IETF network slice, i.e., attributes that can be directly translated into requirements required to be satisfied by a IETF network slice.
- o Indirectly impactful attributes, which are those that impact in an indirect manner on the definition of the IETF network slice, i.e., attributes that indirectly impose some requirements to a IETF network slice.
- o Non-impactful attributes, that are those which do not have impact on the IETF network slice at all.

The following sections describe the attributes falling into the three categories.

#### 4.1.4.1. Attributes with direct impact on the IETF network slice definition

The following attributes impose requirements in the IETF network slice

- o Availability
- o Deterministic communication
- o Downlink throughput per network slice
- o Energy efficiency
- o Group communication support
- o Isolation level
- o Maximum supported packet size
- o Mission critical support
- o Performance monitoring
- o Slice quality of service parameters
- o Support for non-IP traffic
- o Uplink throughput per network slice
- o User data access (i.e., tunneling mechanisms)

#### 4.1.4.2. Attributes with indirect impact on the IETF network slice definition

The following attributes indirectly impose requirements in the IETF network slice to support the end-to-end service.

- o Area of service (i.e., the area where terminals can access a particular network slice)
- o Delay tolerance (i.e., if the service can be delivered when the system has sufficient resources)
- o Downlink (maximum) throughput per UE
- o Network functions owned by Network Slice Customer

- o Maximum number of (concurrent) PDU sessions
- o Performance prediction (i.e., capability to predict the network and service status)
- o Root cause investigation
- o Session and Service Continuity support
- o Simultaneous use of the network slice
- o Supported device velocity
- o UE density
- o Uplink (maximum) throughput per UE
- o User management openness (i.e., capability to manage users' network services and corresponding requirements)
- o Latency from (last) UPF to Application Server

#### 4.1.4.3. Attributes with no impact on the IETF network slice definition

The following attributes do not impact the IETF network slice.

- o Location based message delivery (not related to the geographical spread of the network slice itself but with the localized distribution of information)
- o MMTel support, i.e. support of and Multimedia Telephony Service (MMTel) as well as IP Multimedia Subsystem (IMS) support.
- o NB-IoT Support, i.e., support of NB-IoT in the RAN in the network slice.
- o Maximum number of (simultaneous) UEs
- o Positioning support
- o Radio spectrum
- o Synchronicity (among devices)
- o V2X communication mode
- o Network Slice Specific Authentication and Authorization (NSSAA)

#### 4.1.5. Provisioning procedures

3GPP identifies in [TS28.541] a number of procedures for the provisioning of a network slice in general. It can be assumed that similar procedures may also apply to a transport slice, facilitating a consistent management and control of end-to-end slices.

The envisioned procedures are the following:

- o Slice instance allocation: this procedure permits to create a new slice instance (or reuse an existing one).
- o Slice instance de-allocation: this procedure decommissions a previously instantiated slice.
- o Slice instance modification: this procedure permits the change in the characteristics of an existing slice instance.
- o Get slice instance status: this procedure helps to retrieve run-time information on the status of a deployed slice instance.
- o Retrieval of slice capabilities: this procedure assists on getting information about the capabilities (e.g. maximum latency supported).

All these procedures fit in the operation of transport network slices.

#### 4.2. NFW-based services

NFW technology allows the flexible and dynamic instantiation of virtualized network functions (and their composition into network services) on top of a distributed, cloud-enabled compute infrastructure. This infrastructure can span across different points of presence in a carrier network. By leveraging on transport network slicing, connectivity services established across geographically remote points of presence can be enriched by providing additional QoS guarantees with respect present state-of-the-art mechanisms, as conventional L2/L3 VPNs.

##### 4.2.1. Connectivity attributes

The connectivity services are expressed through a number of attributes as listed:

- o Incoming and outgoing bandwidth: bandwidth required for the connectivity services (in Mbps).

- o Qos metrics: set of metrics (e.g., cost, latency and delay variation) applicable to a specific connectivity service
- o Directionality: indication if the traffic is unidirectional or bidirectional.
- o MTU: value of the largest PDU to be transmitted in the connectivity service.
- o Protection scheme: indication of the kind of protection to be performed (e.g., 1;1, 1+1, etc.)
- o Connectivity mode: indication of the service is point-to-point of point-to-multipoint

All those attributes will assist on the characterization of the connectivity slice to be deployed, and thus, are relevant for the definition of a IETF network slice supporting such connectivity.

#### 4.2.2. Provisioning procedures

ETSI NFV defines the role of WAN Infrastructure Manager (WIM) as the component in charge of managing and controlling the connectivity external to the PoPs. In [IFA032] a number of interfaces are identified to be exposed by the WIM for supporting the multi-site connectivity, thus representing the capabilities expected for a transport network slice, as well, in case of satisfying such connectivity needs by means of the slice concept.

The interfaces considered are the following:

- o Multi-Site Connectivity Service (MSCS) Management: this interface permits the creation, termination, update and query of MSCSs, including reservation. It also enables subscription for notifications and information retrieval associated to the connectivity service.
- o Capacity Management: this interface allows querying about the capacity (e.g. bandwidth), topology, and network edge points of the connectivity service, as well as about information of consumed and available capacity on the underlying network resources.
- o Fault Management: this interface serves for the provision of alarms related to the MSCSs.
- o Performance Management: this interface assists on the retrieval of performance information (measurement results collection and notifications) related to MSCSs.

#### 4.3. Network sharing

Network sharing is one of the means network operators exploit for increasing efficiencies. There are different scenarios of network sharing, being especially popular in the deployment of mobile networks, typically referred to as Radio Access Network (RAN) sharing. From an operational perspective, in RAN sharing we have two roles: master operator, being the actor (e.g. infrastructure provider, network operator) to which the deployment and daily operation of shared RAN elements are entrusted to; and the participant operators, who are the mobile operators who share the RAN facilities provided by the master operator. Note that in this context the master and participant operator can be seen as provider and customer, respectively.

While there exist different modes of RAN sharing [TS23.251], including passive RAN sharing (infrastructure site sharing) and active RAN sharing (e.g. Multi-Operator Core Networks or MOCN), most of the cases require the establishment of separated connections in order to separate the traffic per participant operator. Such connections typically extend from the cell site to some pre-defined and agreed interconnection points, from which the traffic is routed and delivered to individual participant operators.

The above-referred connections can have specific attributes. Aspects like guaranteed bandwidth (in line with the expected load from the aggregated cells), redundancy, bounded latency (per kind of traffic), or secure delivery of the information should be considered.

The master operator is the one in charge of provisioning the connections and collecting management data (e.g. performance measurements, telemetry, fault alarms, trace data) for individual participant operators. The use of network slicing could make the network sharing approach more flexible by allowing the other operators control and manage the established connections [MEF].

The implications of the RAN sharing scenario here described can be extended to either fixed networks or even to mobile networks leveraging on radio functional split (i.e., including fronthaul and midhaul network segments).

##### 4.3.1. Connectivity attributes

The connections for RAN sharing typically consider attributes like:

- o Maximum and Guaranteed Bit Rate (MBR and GBR respectively).
- o Bounded latency (e.g., for user plane, control plane, etc)



- o Packet loss rate.
- o IP addressing (consistent among the operators sharing the infrastructure).
- o L2/L3 reachability.
- o Recovery time (on the event of failures).
- o Secure connection (e.g., encryption support).

#### 4.3.2. Provisioning procedures

The expected provisioning procedures are:

- o Connection provisioning between site and interconnection point. Those connections could evolve in time in terms of capacity depending on the capacity growth of each particular site.
- o Collection of management data, including performance measurements, fault alarms and trace data.

#### 4.4. SD-WAN

SD-WAN is a solution to provide a virtual overlay network for connecting between customer's sites, (virtual) private cloud, or public cloud/Internet. SD-WAN operates over one or more underlay networks, and enables to offer more differentiated service delivery capabilities. SD-WAN can be esteemed as a type of network slices or can be established over underlay networks provided as network slices. The definitions, specification, service attributes, and framework of SD-WAN is defined in Metro Ethernet Forum ([MEF-70]).

SD-WAN forwards traffic based on application flows, and the policies include rules and constraints on the forwarding of the application flows. In SD-WAN, it may be required from the customer to adjust the behaviors based on its needs in near real time. The service provider is required to monitor the performance of the service and modify the forwarding policies based on the real-time telemetry from the underlying network components.

##### 4.4.1. SD-WAN Structure

SD-WAN has three logical constructs:

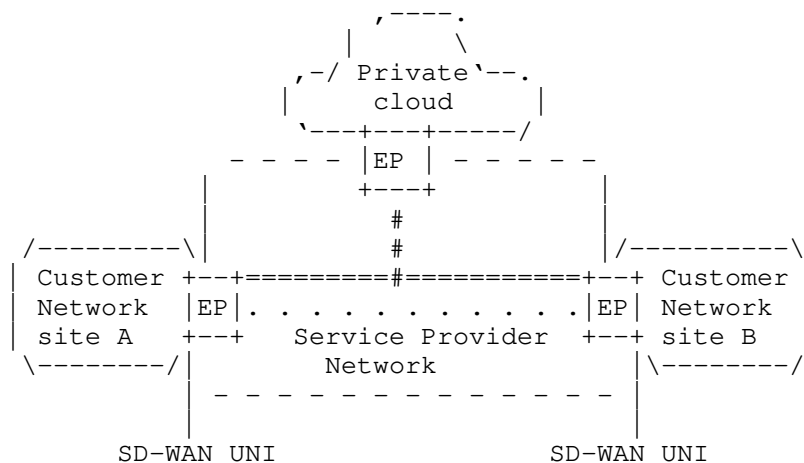
- o SD-WAN virtual connection
- o SD-WAN virtual connection endpoint

- o SD-WAN UNI

Several additional components may be visible to the customer. These include:

- o Customer network
- o Service provider network
- o Underlay connectivity
- o Tunnel virtual connection

The following figure shows the overview of SD-WAN structure. In this case, the customer sites are connected with underlay connectivity#1 and they are also connected to remote private cloud with underlay connectivity#2. An SD-WAN endpoint is usually located in each customer network site as a CPE or a customer edge, and it allocates application flow to appropriate underlay connectivity.



\* Legend

. . . : Underlay connectivity#1  
 ===== : Underlay Connectivity#2  
 EP : SD-WAN Endpoint

Figure 4: Overview of SD-WAN Structure

SD-WAN may be provided as a network slice, or it is realized on several network slices provided as underlay connectivities. In the former case, a network slice PE will be mapped to CE in SD-WAN. In

the later case, PEs of the provider of underlay connectivities will behave as network slice PEs.

#### 4.4.2. Connectivity Attributes

SD-WAN defined in MEF-70 has several attributes on its connectivity as below:

- o SD-WAN Identifier: the value is a string that is used by the customer and service provider to uniquely identify an SD-WAN connectivity.
- o Endpoint list: the value is a list contains endpoint identifiers and their connected endpoints.
- o Service Uptime Objective: the value is the proportion of time that the connectivity service is working during a given time period.
- o Reserved Prefixes: the values are IP prefixes reserved by the service provider for use for SD-WAN within its own network or for distribution to the customer via DHCP or SLAAC.
- o List for Policies: the value is a list of policies applied to application flows and application flow groups at endpoints. An SD-WAN policy list contains policy name and list of policy criteria. Support of the criteria listed below would be required:
  - \* Encryption: indicates whether or not the application flow requires encryption
  - \* Public-Private: indicates whether the application flow can traverse public or private underlay connectivity services (or both).
  - \* Internet-Breakout: indicates whether the application flow should be forwarded to an Internet destination.
  - \* Billing-Method: indicate the application flow can be sent over an underlay connectivity service that has usage-based or flat-rate billing.
  - \* Backup: indicates whether this application flow can use a TVC designated as aEUR&#157;backupaEUR&#157;.
  - \* Bandwidth: specifies a rate limit on the application flow.

- o List of Application Flow Groups: the value is a list of application flow groups that application flows can be members of. An application flow group list contains application flow group name and application flow group policy.
- o List of Application Flows: the value is a list of the application flows that are recognized by the SD-WAN. An application flow list contains application flow name, list of application flow criteria, and application flow group name. The criteria is listed below:
  - \* Ethertype
  - \* C-VLAN ID list
  - \* IPv4 source address
  - \* IPv4 destination address
  - \* IPv4 source or destination address
  - \* IPv4 protocol list
  - \* IPv6 source address
  - \* IPv6 destination address
  - \* IPv6 source or destination address
  - \* IPv6 next header list
  - \* TCP/UDP source port list
  - \* TCP/UDP destination port list
  - \* Application identifier
  - \* any

#### 4.4.3. SD-WAN Endpoint Attributes

SD-WAN contains some endpoints as boundary nodes between underlay connections and customers sites. [MEF-70] defines some attributes for SD-WAN endpoints as below:

- o Endpoint Identifier: the value is for identification of SD-WAN endpoint for management purposes.

- o Endpoint UNI: the value is for identification of the UNI that the endpoint is associated with.
- o Endpoint policy map: the value is for mapping policies to application flows and application flow groups.

#### 4.4.4. SD-WAN UNI Attributes

SD-WAN UNI is a reference point that represents the demarcation between the responsibility of the customer and the responsibility of the provider. Some attributes for UNI is defined in [MEF-70] as below:

- o SD-WAN UNI Identifier: the value is for identification of the UNI for management purposes.
- o SD-WAN UNI L2 Interface: the value describes the underlay L2 interface for the UNI.
- o SD-WAN UNI Maximum L2 Frame Size: the value specifies the maximum length L2 frame that is accepted by the provider.
- o SD-WAN UNI IPv4 connection addressing: the value describes IPv4 connection address mechanisms (e.g., Static or DHCP).
- o SD-WAN UNI IPv6 connection addressing: the value describes IPv6 connection address mechanisms (e.g., DHCP, SLAAC, Static or Link-Local-only).

#### 4.5. Radio functional splits

The disaggregation of the software stack in radio base stations allows the centralization of some of the radio processing functions. O-RAN is promoting the interoperability of implementations of radio functional splits, defining an architecture where three main entities can be considered: the Radio Unit (RU), with some basic processing, the Distributed Unit (DU) with the rest of real-time processing capabilities, and the Centralized Unit (CU) with the non-real-time processing of the software stack. The network segment between RU and DU is known as fronthaul (FH), while the segment between DU and CU is referred as midhaul (MH). Figure 5 shows this situation.

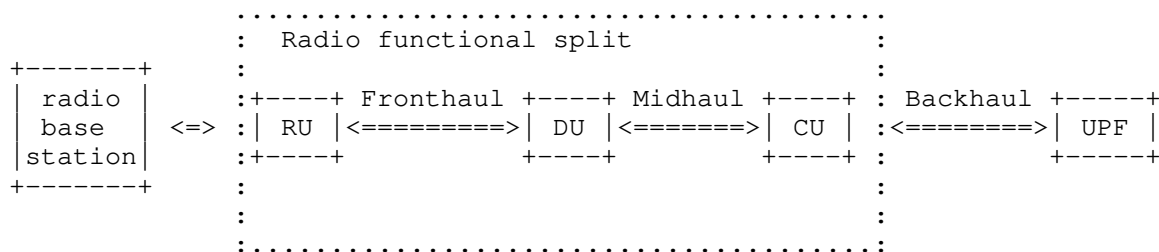


Figure 5: Logical Transport Interfaces

The fronthaul leverages on eCPRI protocol which can be transported directly on Ethernet frames or encapsulated in IP/UDP (for the user plane). The midhaul can be transported in a similar way as the backhaul.

With current specifications, individual service flows being carried by FH cannot be distinguished, so no possibility of differentiating connectivity slices at that point. Similar thing happens for MH. The only possible differentiation per flow can happen in downstream direction from CU to DU, but this basically can only help for policing traffic at that point (i.e., slice is yet the same).

Advanced scenarios such as RU sharing could allow traffic differentiation per mobile operator based on e.g. vlans, being each of those vlans mapped to a different slice.

#### 4.5.1. Attributes and procedures

The attributes of IETF network slices for the conveniently supported the radio functional split are based on main characteristics of FH/MH: Latency, BW, and packet loss, as specified in [O-RAN]. Geographical location could have an impact due to latency restrictions for FH.

Regarding slice management procedures, it can be assumed a similar lifecycle as in 3GPP slices.

#### 4.6. Additional use cases

This is a placeholder for describing additional use cases (e.g., data center interconnection, etc). To be completed.

## 5. Summary of attributes and procedures

After analysing the different use cases, a number of attributes and procedures can be identified to provide IETF Network Slice services. Following sections summarize the findings per SLO, SLE and procedures.

Editor Note: this summary is yet under review.

### 5.1. Summary of SLOs

The following SLOs can be considered common to the majority of use cases.

- o Bandwidth (or throughput), as an indication of the amount of traffic allowed to be delivered. It can be expressed unidirectional or bidirectional.
- o Latency, as an indication of the maximum delay expected in a connection.
- o Jitter (or delay variation), as an indication of the maximum variation on the delay expected in a connection.
- o Packet loss, as an indication of the bounded limit of packet losses allowed in a connection
- o To be completed

### 5.2. Summary of SLEs

To be completed.

### 5.3. Summary of procedures

The following procedures allow to cover the analysed use cases.

- o IETF Network Slice provision, including allocation and de-allocation of the slice.
- o IETF Network Slice modification (or update) of an existing allocated slice.
- o Retrieval (or query) of IETF Network Slice status and capabilities of an existing allocated slice.
- o IETF Network Slice reservation, allowing a late instantiation of the slice.

- o IETF Network Slice fault management, permitting the collection of alarms associated to the IETF Network Slice.
- o IETF Network Slice performance management, permitting the retrieval of performance measurements associated to the IETF Network Slice.

## 6. Security Considerations

This draft does not include any security considerations.

## 7. IANA Considerations

This draft does not include any IANA considerations

## 8. References

### 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

### 8.2. Informative References

[GSMA] "Generic Network Slice Template, version 3.0", NG.116 , May 2020.

[I-D.homma-slice-provision-models]  
Homma, S., Nishihara, H., Miyasaka, T., Galis, A., OV, V. R., Lopez, D. R., Contreras, L. M., Ordonez-Lucena, J. A., Martinez-Julia, P., Qiang, L., Rokui, R., Ciavaglia, L., and X. D. Foy, "Network Slice Provision Models", draft-homma-slice-provision-models-02 (work in progress), November 2019.

[I-D.ietf-teas-ietf-network-slice-definition]  
Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-01 (work in progress), February 2021.

[I-D.ietf-teas-ietf-network-slices]  
Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", draft-ietf-teas-ietf-network-slices-08 (work in progress), March 2022.



- [I-D.nsd-t-teas-ns-framework]  
Gray, E. and J. Drake, "Framework for IETF Network Slices", draft-nsdt-teas-ns-framework-05 (work in progress), February 2021.
- [IFA032] "IFA032 Interface and Information Model Specification for Multi-Site Connectivity Services V3.2.1.", ETSI GS NFV-IFA 032 V3.2.1 , April 2019.
- [MEF] "Slicing for Shared 5G Fronthaul and Backhaul", MEF White paper , April 2020.
- [MEF-70] "SD-WAN Service Attributes and Services", MEF-70 , July 2019.
- [O-RAN] "O-RAN Xhaul Transport Requirements 1.0", O-RAN.WG9.XTRP-REQ-v01.00 , November 2020.
- [TS23.251]  
"TS 23.251 Network Sharing; Architecture and functional description (Release 16) V16.0.0.", 3GPP TS 23.251 V16.0.0 , July 2020.
- [TS28.530]  
"TS 28.530 Management and orchestration; Concepts, use cases and requirements (Release 16) V16.0.0.", 3GPP TS 28.530 V16.0.0 , September 2019.
- [TS28.541]  
"TS 28.541 Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 16) V16.2.0.", 3GPP TS 28.541 V16.2.0 , September 2019.

#### Authors' Addresses

Luis M. Contreras  
Telefonica  
Ronda de la Comunicacion, s/n  
Sur-3 building, 3rd floor  
Madrid 28050  
Spain

Email: [luismiguel.contrerasmurillo@telefonica.com](mailto:luismiguel.contrerasmurillo@telefonica.com)  
URI: <http://lmcontreras.com/>

Shunsuke Homma  
NTT  
Japan

Email: shunsuke.homma.ietf@gmail.com

Jose A. Ordonez-Lucena  
Telefonica  
Ronda de la Comunicacion, s/n  
Sur-3 building, 3rd floor  
Madrid 28050  
Spain

Email: joseantonio.ordonezlucena@telefonica.com

Jeff Tantsura  
Microsoft

Email: jefftant.ietf@gmail.com

Hidetaka Nishihara  
NTT

Email: hidetaka.nishihara1104@gmail.com

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 12, 2022

J. Dong  
Z. Li  
Huawei Technologies  
L. Gong  
China Mobile  
G. Yang  
China Telecom  
J. Guichard  
Futurewei Technologies  
G. Mishra  
Verizon Inc.  
F. Qin  
China Mobile  
July 11, 2021

Scalability Considerations for Enhanced VPN (VPN+)  
draft-dong-teas-enhanced-vpn-vtn-scalability-03

Abstract

Enhanced VPN (VPN+) aims to provide enhancements to existing VPN services to support the needs of new applications, particularly including the applications that are associated with 5G services. VPN+ could be used to provide network slicing, and may also be of use in more generic scenarios, such as enterprise services which have demanding requirement. With the requirement for VPN+ services increase, scalability would become an important factor for the deployment of VPN+. This document describes the scalability considerations in the control plane and data plane to enable VPN+ services, some optimization mechanisms are also described.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 12, 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .                                | 2  |
| 2. VPN+ Scalability Requirements . . . . .               | 3  |
| 3. VPN+ Scalability Considerations . . . . .             | 5  |
| 3.1. Control Plane Scalability . . . . .                 | 5  |
| 3.1.1. Distributed Control Plane . . . . .               | 5  |
| 3.1.2. Centralized Control Plane . . . . .               | 6  |
| 3.2. Data Plane Scalability . . . . .                    | 6  |
| 3.3. Gap Analysis of Existing Mechanisms . . . . .       | 7  |
| 4. Possible Scalability Optimizations . . . . .          | 8  |
| 4.1. Control Plane Optimizations . . . . .               | 8  |
| 4.2. Data Plane Optimizations . . . . .                  | 10 |
| 5. Solution Evolution for Improved Scalability . . . . . | 11 |
| 6. Security Considerations . . . . .                     | 12 |
| 7. IANA Considerations . . . . .                         | 12 |
| 8. Contributors . . . . .                                | 12 |
| 9. Acknowledgments . . . . .                             | 12 |
| 10. References . . . . .                                 | 12 |
| 10.1. Normative References . . . . .                     | 12 |
| 10.2. Informative References . . . . .                   | 13 |
| Authors' Addresses . . . . .                             | 14 |

## 1. Introduction

Virtual Private Networks (VPNs) have served the industry well as a means of providing different customers with logically isolated connectivity over a common network infrastructure. The common or base network that is used to provide the VPNs is often referred to as the underlay, and the VPN is often called an overlay. The underlay is responsible for establishing the network connectivity and managing network resources to meet the service requirement. The overlay is used to distribute the membership and reachability information of the

customer, and provide logical separation of service delivery between different customers.

Enhanced VPN service (VPN+) [I-D.ietf-teas-enhanced-vpn] is targeted at new applications which require better isolation between customers and/or services, and have more stringent performance requirements than can be provided with existing VPNs. To meet the requirement of VPN+ services, a number of Virtual Transport Networks (VTNs) need to be created, each has a subset of the underlay network topology and a set of network resources allocated from the physical network to meet the requirements of one or a group of VPN+ services. The overlay VPNs together with the corresponding underlay VTN provide the VPN+ service.

Section 6 of [I-D.ietf-teas-enhanced-vpn] provides some general analysis of the scalability of VPN+. This document gives detailed analysis of the scalability considerations when a large number of VPN+ services are provided. Since the scalability of the overlay is not the major bottleneck, this document mainly focuses on the scalability of the underlay VTN.

## 2. VPN+ Scalability Requirements

As described in [I-D.ietf-teas-enhanced-vpn], VPN+ services may require additional state to be introduced into the network to take advantage of the enhanced functionality. This introduces some scalability considerations to the network. This section gives some analysis of the number of VPN+ services that might be needed in a network.

There are several use cases where VPN+ may be needed, and these determine how many VPN+ will be required in a network. One typical use case of VPN+ is to deliver IETF network slice [I-D.ietf-teas-ietf-network-slices] for applications or services in 5G and other scenarios, thus the number of IETF network slices needed could reflect the number of VPN+ services. With the development and evolution of 5G, it is expected that an increasing number of network slices will be deployed. The number of network slices required depends on how IETF network slices will be used, and the progress of 5G for the vertical industrial services. The potential number of network slices is analyzed by classifying the network slicing deployment into three typical scenarios:

1. Network slices can be used by a network operator internally for different types of services. For example, in a converged multi-service network, different network slices can be created to carry mobile transport service, fixed broadband service and enterprise services respectively, each type of service could be managed by a

separate department or management team. Some service types, such as multicast service may also be deployed in a dedicated network slice. It is also possible that an infrastructure network operator provides network slices to other network operators as a wholesale service. In this scenario, the number of network slices in a network would be relatively small, such as on the order of 10 or so. This could be the typical case in the beginning of the network slice deployment.

2. Network slices can be used to provide isolated and customized virtual networks for customers in different vertical industries. At the early stage of the vertical industrial service deployment, a few top customers in some industries will begin to use network slices to ensure the performance of their business, such as smart grid, manufacturing, public safety, on-line gaming, etc. Considering the number of the vertical industries, and the number of top customers in each industry, the number of network slices may increase to the order of 100.
3. With the evolution of 5G, network slices could be widely used by both vertical industrial customers and enterprise customers which require guaranteed or predictable service performance. The total amount of network slices may increase to the order of 1000 or more. However, it is expected that the number of network slices would still be less than the number of traditional VPN services in the network.

In 3GPP [TS23501], a 5G network slice is identified using Single Network Slice Selection Assistance Information (S-NSSAI), which is a 32-bit identifier comprised of 8-bit Slice/Service Type (SST) and 24-bit Slice Differentiator (SD). This allows the mobile networks (RAN and CN) to provide a large number of network slices. Although it is possible that multiple 5G network slices in RAN and CN are mapped to the same IETF network slice, the number of IETF network slices may still be comparable with the number of 5G network slices. Thus the scalability of IETF network slices needs to be taken into consideration.

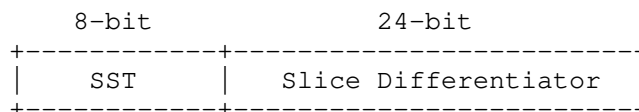


Figure 1. Format of S-NSSAI in 3GPP

VPN+ needs to meet the scalability requirement of network slicing in different scenarios. The increased number of VPN+ services will introduce additional complexity and overhead to both the control

plane and data plane, especially in the aspects related to the underlying VTNs. Although multiple VPN+ services can be mapped to the same VTN as the underlay, there still can be scalability challenges with the increased number of VTNs.

### 3. VPN+ Scalability Considerations

In this section, the scalability of VTN in the control plane and data plane is analyzed to understand the possible gaps in meeting the scalability requirement of VPN+.

#### 3.1. Control Plane Scalability

As described in [I-D.ietf-teas-enhanced-vpn], the control plane of VPN+ could be based on the hybrid of a centralized controller and the distributed control plane.

##### 3.1.1. Distributed Control Plane

At part of the construction of VPN+ services, it is necessary to create multiple VTNs which provide customized topology and resource attributes. The attributes and state information of each VTN needs to be exchanged in the control plane. The scalability of the distributed control plane for the establishment and maintenance of VTNs needs to be considered in the following aspects:

- o The number of control protocol instances maintained on each node
- o The number of protocol sessions maintained on each link
- o The number of routes advertised by each node
- o The amount of attributes associated with each route
- o The number of route computation (i.e. SPF computation) executed on each node

As the number of VTNs increases, it is expected that in some of the above aspects, the overhead in the control plane may increase dramatically. For example, the overhead of maintaining separated control protocol instances (e.g. IGP instances) for different VTNs is considered higher than maintaining the information of separated VTNs in the same control protocol instance with appropriate separation, and the overhead of maintaining separate protocol sessions for different VTNs is considered higher than using a shared protocol session for the information exchange of multiple VTNs. To meet the requirement of the increasing number of VTNs, It is

suggested to choose the control plane mechanisms which could improve the scalability while still provide the required functionality.

### 3.1.2. Centralized Control Plane

Although the SDN approach can reduce the amount of control plane overhead in the distributed control plane, it may transfer some of the scalability concerns from network nodes to the centralized controller, thus the scalability of the controller also needs to be considered.

To provide global optimization for the Traffic Engineered (TE) paths in different VTNs, the controller needs to keep the topology and resource information of all the VTNs up to date. To achieve this, the controller may need to maintain a communication channel with each network node in the network. When there is significant change in the network, or multiple VTNs requires global optimization concurrently, there may be a heavy processing burden at the controller, and a heavy load in the network surrounding the controller for the distribution of the updated network state and the TE paths.

### 3.2. Data Plane Scalability

To provide different VPN+ services with the required isolation and performance characteristics, it is necessary to allocate different sets of network resources to different VTNs. As the number of VPN+ increases, the number of VTNs will increase accordingly. This requires the underlying network to provide fine-granular network resource partitioning, which means the amount of state about the reserved network resources to be maintained on network nodes will also increase.

In data plane, traffic of different VPN+ services need to be processed separately according to the topology and resource constraints of the associated VTN, thus the information used for VTN identification needs to be carried either directly or implicitly in the data packet. Different approaches of encapsulating the VTN information in data packet can have different scalability implications.

One approach is to reuse some existing fields in the data packet to additionally identify the VTN the packet belongs to. This avoids the cost of introducing new fields in the data packet, while since it introduces additional semantics to an existing field, it requires to change the processing of the existing field in packet forwarding. And when the identifiers which were used to identify a node or link are reused to further identify a VTN, the number of the identifiers



may be increased in proportion to the number of the VTNs, which may cause scalability problem in some networks.

Another alternative approach is to introduce a dedicated field in the packet for VTN identification. This could avoid the impact to the existing fields in the packet. And if this new field carries a global-significant VTN identifier, it could be used together with the existing fields to determine the VTN-specific packet forwarding. The potential issue with this approach is the difficulty in introducing a new field in some types of the data plane.

In addition, the introduction of per VTN packet forwarding has impact on the scalability of the forwarding entries on network nodes, as a network node may need to maintain separate forwarding entries for each VTN it participates in.

### 3.3. Gap Analysis of Existing Mechanisms

One candidate approach to build VTN is to use VTN specific Segment Routing (either SR-MPLS or SRv6) Identifiers in the data plane [I-D.ietf-spring-sr-for-enhanced-vpn], and define and distribute the associated topology and resource attribute of each VTN based on Multi-topology [RFC4915] [RFC5120] [I-D.ietf-lsr-isis-sr-vtn-mt], Flex-Algo [I-D.ietf-lsr-flex-algo] [I-D.zhu-lsr-isis-sr-vtn-flexalgo] or the combination of these mechanisms in the control plane. This mechanism is suitable for networks with a limited number of VTNs. As the number of VTNs increases, there may be several scalability challenges with this approach:

1. The number of SR SIDs needed will increase in proportion to the number of VTNs in the network, which will bring challenges both to the distribution of SIDs and the related information in the control plane, and to the installation of forwarding entries for VTN-specific SIDs in the data plane.
2. The number of route computation (e.g. SPF computation) will increase in proportion to the number of VTNs in the network, which may introduce significant overhead to the control plane of network nodes.
3. The maximum number of logical topologies supported by OSPF is 128, and the maximum number of Flex-Algo is 128, which may not meet the required number of VTNs in some network scenarios.

## 4. Possible Scalability Optimizations

### 4.1. Control Plane Optimizations

For the distributed control plane, several optimizations can be considered to reduce the control plane overhead and improve the scalability.

The first optimization mechanism is to reduce the amount of control plane sessions used for the establishment and maintenance of the VTNs. For multiple VTNs which have the same peering relationship between two adjacent network nodes, it is proposed that one single control protocol session is used for the establishment of multiple VTNs. The information of different VTNs can be exchanged over the same session, with necessary identification information to distinguish the VTNs in the control messages. This could reduce the overhead of maintaining a large number of control protocol sessions for different VTNs, and could also reduce the amount of control plane messages flooded in the network.

The second optimization mechanism is to decompose the attributes of a VTN into different groups, so that different types of VTN attribute can be advertised and processed separately in control plane. There are two basic types of attributes associated with a VTN: the topology attribute and the network resource attribute. In a network, it is possible that multiple VTNs share the same topology, and multiple VTNs may share the same set of network resources on particular network segments. Then it is more efficient if only one copy of the topology attribute is advertised, and multiple VTNs sharing the same topology could refer to this topology information. More importantly, with this approach the result of topology-based route computation could be shared by multiple VTNs, so that the overhead of per-VTN route computation could be reduced. Similarly, information of a subset of network resources reserved on a particular network segment could be advertised once and be referred to by multiple VTNs which share the same set of resources. This methodology could also apply to other attributes of VTN which may be introduced later and can be processed independently.

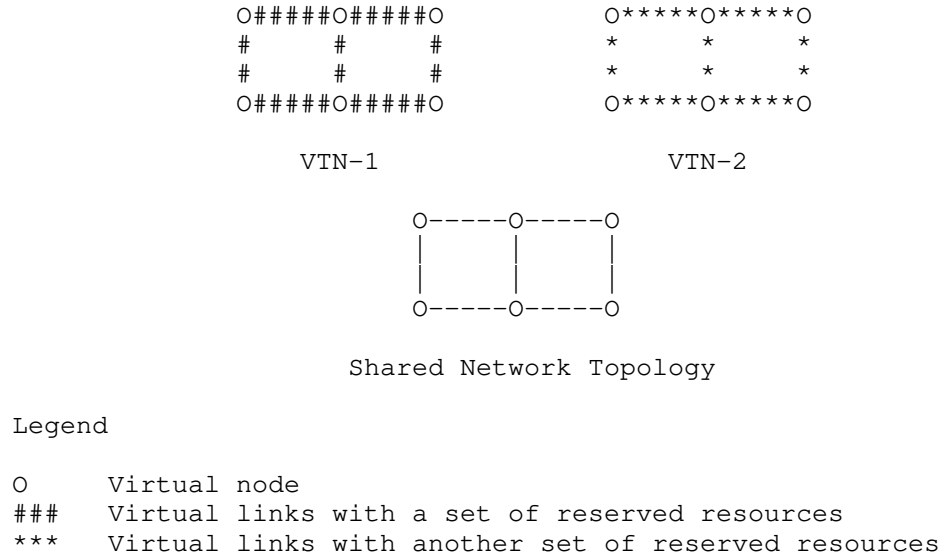


Figure 2. Topology Sharing between VTNs

FIG-2

Figure 2 gives an example of two VTNs which share the same logical topology attribute. As shown in the figure, VTN-1 and VTN-2 have the same topology, while the link resource attributes of each VTN are different. In this case, only one copy of the network topology information needs to be advertised, and the topology-based route computation result can be shared by the two VTNs to generate the corresponding routing and forwarding tables.

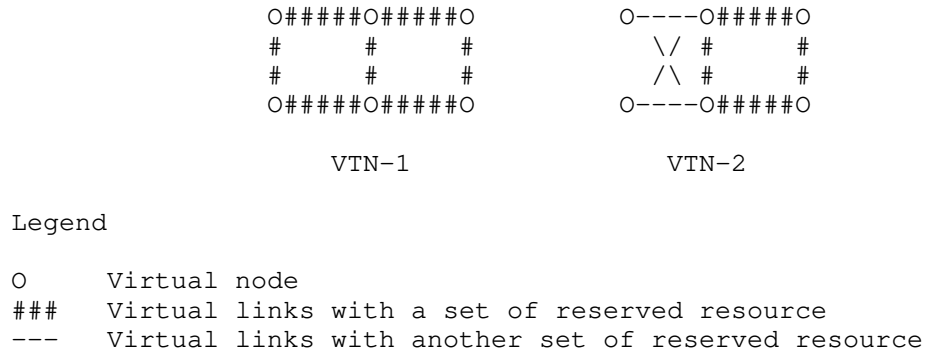


Figure 3. Resource Sharing between VTNs

Figure 3 gives another example of two VTNs which share the same set of network resources on some links. In this case, information about the reserved resource on each link only needs to be advertised once, then both VTN-1 and VTN-2 could refer to the link resource for constraint based path computation.

For the optimization of the centralized control plane, it is suggested that the centralized controller is used as a complementary mechanism to the distributed control plane rather than a replacement, so that the VTN specific path computation burden in control plane could be shared by both the centralized controller and the network nodes, thus the scalability of both systems could be improved.

#### 4.2. Data Plane Optimizations

To support more VPN+ services while keeping the amount of data plane state at a reasonable scale, one possible approach is to classify a set of VPN+ services which have similar service characteristics and performance requirements into a group, and such group of VPN+ services is mapped to one VTN, which is allocated with an aggregated set of network topology and resources to meet the service requirement of the whole group of VPN+. Different groups of VPN+ services need to be mapped to different VTNs with different set of network resources allocated. With appropriate grouping of VPN+ services, a reasonable number of VTNs with network resources reservation and aggregation could still meet the service requirements.

Another optimization in the data plane is to decouple the identifier used for topology-based forwarding and the identifier used for the resource-specific processing introduced by VTN. One possible mechanism is to introduce a dedicated VTN-ID in the packet header to uniquely identify the set of local network resources allocated to a VTN on each network node for the processing and forwarding of the received packet. Then the existing identifier in the packet header used for topology based forwarding is kept unchanged. The benefit is the amount of topology-specific identifiers is in proportion to the number of topologies rather than the number of VTNs, so that its scalability will not be impacted by the increased number of VTN. Since this new VTN-ID field will be used together with the existing fields to determine the VTN-specific packet forwarding, this MAY require network nodes to support a hierarchical forwarding table in the data plane. Figure 4 shows the concept of using different data plane identifiers for topology-based and VTN resource-based packet processing respectively.

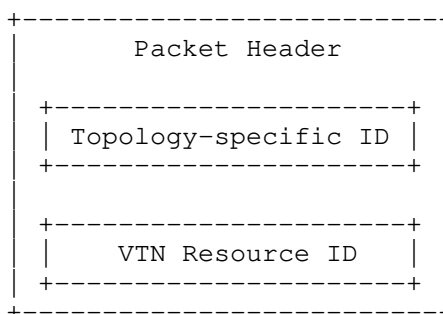


Figure 4. Decoupled Data Plane Identifiers

In an IPv6 [RFC8200] based network, this could be achieved by introducing a dedicated field in either the IPv6 fixed header or the extension headers to carry the VTN identifier for the resource-specific forwarding, while keeping the destination IP address field used for routing towards the destination prefix in the corresponding topology. Note that the VTN-ID needs to be parsed by every node along the path which is capable of VTN-specific forwarding. In an MPLS [RFC3032] based network, this may be achieved by introducing a dedicated MPLS label to identify the VTN, while the existing MPLS labels could be used for topology-based packet forwarding towards the associated destination prefix. This requires that both labels be parsed by each node along the forwarding path of the packet, and the forwarding behaviour depends on the position of the VTN label in the label stack. Another option with the MPLS data plane is to introduce a new MPLS extension header which follows the MPLS label stack to carry the VTN-ID and the associated information. The detailed extensions in IPv6 and MPLS data plane encapsulation are out of the scope of this document.

## 5. Solution Evolution for Improved Scalability

Based on the analysis in this document, the control plane and data plane for VPN+ needs to evolve to support the increasing number of VPN+ services in the network.

At the first step, by introducing resource-awareness to segment routing SIDs [I-D.ietf-spring-resource-aware-segments], and using Multi-Topology or Flex-Algo as the control plane, it could provide a solution for building a limited number of VTNs in the network to meet the requirement of a relatively small number of VPN+ services in the network. This mechanism is considered as the basic SR VTN.

As the number of required VPN+ services increases, more VTNs may be needed, then the control plane scalability could be improved by

decoupling the topology attribute from other attributes (e.g. resource attribute) of VTN, so that multiple VTNs could share the same topology or resource attribute. This mechanism is considered as the scalable SR VTN. Both the basic and the scalable SR VTN mechanisms are described in [I-D.ietf-spring-sr-for-enhanced-vpn].

If the data plane scalability becomes a concern, dedicated data plane VTN-ID can be introduced to decouple the topology-specific identifiers from the VTN-specific resource identifiers in the data plane, this could help to reduce the number of SR SIDs needed to support a large number of VTNs. This mechanism is considered as the Resource-Independent (RI) VTN.

## 6. Security Considerations

TBD

## 7. IANA Considerations

This document makes no request of IANA.

## 8. Contributors

Zhibo Hu  
Email: huzhibo@huawei.com

Hongjie Yang  
Email: hongjie.yang@huawei.com

## 9. Acknowledgments

The authors would like to thank Adrian Farrel for the review and discussion of this document.

## 10. References

### 10.1. Normative References

[I-D.ietf-teas-enhanced-vpn]

Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", draft-ietf-teas-enhanced-vpn-07 (work in progress), February 2021.

## 10.2. Informative References

- [I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-15 (work in progress), April 2021.
- [I-D.ietf-lsr-isis-sr-vtn-mt]  
Xie, C., Ma, C., Dong, J., and Z. Li, "Using IS-IS Multi-Topology (MT) for Segment Routing based Virtual Transport Network", draft-ietf-lsr-isis-sr-vtn-mt-00 (work in progress), March 2021.
- [I-D.ietf-spring-resource-aware-segments]  
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Introducing Resource Awareness to SR Segments", draft-ietf-spring-resource-aware-segments-02 (work in progress), February 2021.
- [I-D.ietf-spring-sr-for-enhanced-vpn]  
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Segment Routing based Virtual Transport Network (VTN) for Enhanced VPN", draft-ietf-spring-sr-for-enhanced-vpn-00 (work in progress), February 2021.
- [I-D.ietf-teas-ietf-network-slices]  
Farrel, A., Gray, E., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", draft-ietf-teas-ietf-network-slices-00 (work in progress), April 2021.
- [I-D.zhu-lsr-isis-sr-vtn-flexalgo]  
Zhu, Y., Dong, J., and Z. Hu, "Using Flex-Algo for Segment Routing based VTN", draft-zhu-lsr-isis-sr-vtn-flexalgo-02 (work in progress), February 2021.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <<https://www.rfc-editor.org/info/rfc4915>>.

- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [TS23501] "3GPP TS23.501", 2016, <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>>.

## Authors' Addresses

Jie Dong  
Huawei Technologies  
Huawei Campus, No. 156 Beiqing Road  
Beijing 100095  
China

Email: [jie.dong@huawei.com](mailto:jie.dong@huawei.com)

Zhenbin Li  
Huawei Technologies  
Huawei Campus, No. 156 Beiqing Road  
Beijing 100095  
China

Email: [lizhenbin@huawei.com](mailto:lizhenbin@huawei.com)

Liyan Gong  
China Mobile  
No. 32 Xuanwumenxi Ave., Xicheng District  
Beijing  
China

Email: [gongliyan@chinamobile.com](mailto:gongliyan@chinamobile.com)



Guangming Yang  
China Telecom  
No.109 West Zhongshan Ave., Tianhe District  
Guangzhou  
China

Email: yangguangm@chinatelecom.cn

James N Guichard  
Futurewei Technologies  
2330 Central Express Way  
Santa Clara  
USA

Email: james.n.guichard@futurewei.com

Gyan Mishra  
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Fengwei Qin  
China Mobile  
No. 32 Xuanwumenxi Ave., Xicheng District  
Beijing  
China

Email: qinfengwei@chinamobile.com

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 28 April 2022

J. Dong  
Z. Li  
Huawei Technologies  
L. Gong  
China Mobile  
G. Yang  
China Telecom  
J. Guichard  
Futurewei Technologies  
G. Mishra  
Verizon Inc.  
F. Qin  
China Mobile  
25 October 2021

Scalability Considerations for Enhanced VPN (VPN+)  
draft-dong-teas-enhanced-vpn-vtn-scalability-04

Abstract

Enhanced VPN (VPN+) aims to meet the needs of some customers or applications, including the customers and applications that are associated with 5G, which requires connectivity services with advanced characteristics, such as the assurance of some Service Level Objectives (SLOs) and specific Service Level Expectations (SLEs). VPN+ could be used for network slice realization both in the context of 5G and in more generic scenarios, such as enterprise services which have requirement on the performance assurance. With the demand for VPN+ services increases, scalability would become an important factor for the large scale deployment of VPN+. This document describes the scalability considerations about the network control plane and data plane in enabling VPN+ services, some optimization mechanisms are also proposed.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 28 April 2022.

#### Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .                                | 3  |
| 2. VPN+ Scalability Requirements . . . . .               | 4  |
| 3. VTN Scalability Considerations . . . . .              | 5  |
| 3.1. Control Plane Scalability . . . . .                 | 6  |
| 3.1.1. Distributed Control Plane . . . . .               | 6  |
| 3.1.2. Centralized Control Plane . . . . .               | 6  |
| 3.2. Data Plane Scalability . . . . .                    | 7  |
| 3.3. Gap Analysis of Existing Mechanisms . . . . .       | 8  |
| 4. Proposed Scalability Optimizations . . . . .          | 8  |
| 4.1. Control Plane Optimizations . . . . .               | 9  |
| 4.2. Data Plane Optimizations . . . . .                  | 11 |
| 5. Solution Evolution for Improved Scalability . . . . . | 12 |
| 6. Security Considerations . . . . .                     | 13 |
| 7. IANA Considerations . . . . .                         | 13 |
| 8. Contributors . . . . .                                | 13 |
| 9. Acknowledgments . . . . .                             | 13 |
| 10. References . . . . .                                 | 14 |
| 10.1. Normative References . . . . .                     | 14 |
| 10.2. Informative References . . . . .                   | 14 |
| Authors' Addresses . . . . .                             | 16 |

## 1. Introduction

Virtual Private Networks (VPNs) have served the industry well as a means of providing different customers with logically separated connectivity services over a common network infrastructure. The common or base network that is used to provide the VPNs is often referred to as the underlay, and the VPNs are often called the overlay. The underlay network is responsible for establishing the network connectivity and managing the network resources to meet specific service requirement. The overlay network is used to distribute the membership and reachability information of the customers, and provide logical separation in terms of service delivery between different customers in the shared network.

Enhanced VPN (VPN+) aims to meet the needs of some customers or applications, including the applications that are associated with 5G, which requires connectivity services with advanced characteristics, such as the assurance of Service Level Objectives (SLOs) and specific Service Level Expectations (SLEs).

[I-D.ietf-teas-ietf-network-slices] defines the terminologies and the general framework of IETF network slices. VPN+ could be used for IETF network slice realization both in the context of 5G and in more generic scenarios, such as enterprise services which have requirement on the performance assurance.

[I-D.ietf-teas-enhanced-vpn] describes the framework for delivering VPN+ services. To meet the requirement of some VPN+ services, a Virtual Transport Networks (VTNs) need to be created, which has a subset of network resources allocated from the physical network and is associated with a logical network topology to meet the requirements of one or a group of VPN+ services. VPN+ services can be delivered by mapping one or a group of overlay VPNs to the appropriate VTNs as the virtual underlay.

Section 6 of [I-D.ietf-teas-enhanced-vpn] provides some general analysis of the scalability of VPN+. This document gives further analysis of the scalability considerations when a large number of VPN+ services needs to be provided. Since the scalability of the overlay is usually not the major bottleneck, this document mainly focuses on the scalability of the VTNs in the underlay .

## 2. VPN+ Scalability Requirements

As described in [I-D.ietf-teas-enhanced-vpn], VPN+ services may require additional state to be introduced into the network to take advantage of the enhanced functionality. This may introduce some concerns about the network scalability. This section gives some analysis of the number of VPN+ services and the VTNs that might be needed in different network scenarios.

Since the typical use case of VPN+ is to deliver IETF network slice [I-D.ietf-teas-ietf-network-slices] for customers and services in 5G and other scenarios, the number of IETF network slices required could reflect the number of VPN+ needed in the network. With the development and evolution of 5G and other services, it is expected that an increasing number of IETF network slices will be deployed. The number of network slices required depends on how IETF network slices will be used, and the progress of network slicing for the vertical industrial services. The potential number of VPN+ services and VTNs is analyzed by classifying the network slice deployment into three typical scenarios:

1. IETF network slices can be used by a network operator for different types of services. For example, in a converged multi-service network, different IETF network slices can be created to carry mobile transport service, fixed broadband service and enterprise services respectively, each type of service could be managed by a separate department or management team. Some service types, such as multicast service may also be deployed in a dedicated network slice. In this case, a separate VTN may need to be created for each service type. It is also possible that a network infrastructure operator provides IETF network slices to other network operators as a wholesale service, and a VTN may also be needed for each wholesale service customer. In this scenario, the number of VTNs in a network could be relatively small, such as in the order of 10 or so. This could be one of the typical cases in the beginning of IETF network slice deployment.
2. IETF network slices can be requested by customers in vertical industries, where the assurance of SLOs and the fulfilment of SLEs are quite important. At the early stage of the vertical industrial services, a few top customers in some industries will begin to use IETF network slices to provide performance assurance to their business, such as smart grid, manufacturing, public safety, on-line gaming, etc. The realization of such IETF network slices typically requires to provide different VTNs for different industries, and some top customers can require dedicated VTNs for strict service performance guarantee.

Considering the number of vertical industries, and the number of top customers in each industry, the number of VTNs needed may be in the order of 100.

3. With the evolution of 5G and cloud networks, IETF network slices could be widely used by various vertical industrial customers and enterprise customers who require guaranteed or predictable service performance. The total amount of IETF network slices may increase to thousands or more, although it is expected that the number of IETF network slices would still be less than the number of traditional VPN services in the network. Accordingly, the number of VTNs needed may be in the order of 1000.

As defined by 3GPP [TS23501], a 5G network slice is identified using the Single Network Slice Selection Assistance Information (S-NSSAI), which is a 32-bit identifier comprised of 8-bit Slice/Service Type (SST) and 24-bit Slice Differentiator (SD). This allows the mobile networks (the RAN and mobile core networks) to support a large number of 5G network slices. Although it is likely that multiple 5G network slices are mapped to the same IETF network slice, in some cases the number of IETF network slices may still be comparable to the number of 5G network slices.

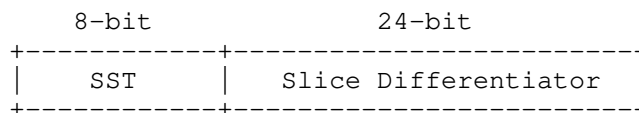


Figure 1. Format of S-NSSAI in 3GPP

Thus solution of VPN+ and VTN needs to meet the scalability requirement of IETF network slices in different scenarios. The increased number of VPN+ services will introduce additional complexity and overhead both to the control plane and the data plane, especially in the aspects related to the underlay VTNs. Although in many cases multiple VPN+ services can be mapped to the same VTN as the underlay, there still can be scalability challenges with the increased number of VTNs.

### 3. VTN Scalability Considerations

In this section, the scalability of VTN in the control plane and data plane is analyzed to understand the possible gaps in meeting the scalability requirement of VPN+ and VTN.

### 3.1. Control Plane Scalability

As described in [I-D.ietf-teas-enhanced-vpn], the control plane of VPN+ could be based on the hybrid of a centralized controller and the distributed control plane.

#### 3.1.1. Distributed Control Plane

At part of the delivery of VPN+ services, it is necessary to create multiple VTNs, each of which is allocated with a set of dedicated or shared network resources, and is associated with a customized logical topology. The topological and resource attributes and the state information of each VTN may need to be exchanged among the network nodes. The scalability of the distributed control plane used for the distribution of VTN information needs to be considered in the following aspects:

- \* The number of control protocol instances maintained on each node
- \* The number of protocol sessions maintained on each link
- \* The number of routes advertised by each node
- \* The amount of attributes associated with each route
- \* The number of route computation (i.e. SPF computation) executed by each node

As the number of VTNs increases, it is expected that in some of the above aspects, the overhead in the control plane may increase dramatically. For example, the overhead of maintaining separated control protocol instances (e.g. IGP instances) for different VTNs is considered higher than maintaining the information of separated VTNs in the same control protocol instance with appropriate separation, and the overhead of maintaining separate protocol sessions for different VTNs is considered higher than using a shared protocol session for the information exchange of multiple VTNs. To meet the requirement of the increasing number of VTNs, It is suggested to choose the control plane mechanisms which could improve the scalability while still provide the required functionality.

#### 3.1.2. Centralized Control Plane

By introducing the centralized network controller, the SDN approach can reduce the amount of control plane overhead in the distributed control plane, while it may also transfer some of the scalability concerns from network nodes to the centralized controller, thus the scalability of the controller also needs to be considered.

To provide global optimization for the Traffic Engineered (TE) paths in different VTNs, the controller needs to keep the topology and resource information of all the VTNs up-to-date. To achieve this, the controller may need to maintain a communication channel with each network node in the network. When there is significant change in the network, or multiple VTNs requires global optimization concurrently, there may be a heavy processing burden at the controller, and a heavy load in the network surrounding the controller for the distribution of the updated network state and the TE paths.

### 3.2. Data Plane Scalability

To provide different VPN+ services with the required SLOs and SLEs, it is necessary to allocate different subsets of network resources to different VTNs to avoid or reduce unexpected interruption. As the number of VTNs increases, it is required that the underlying network can provide fine-granular network resource partitioning, which means the amount of state about the partitioned network resources to be maintained on the network nodes will also increase.

In packet forwarding, VPN+ service traffic needs to be processed separately according to the topology and resource attributes of the VTN it mapped to, this means that some fields in the data packet needs to be used to identify the VTN topology and resources either directly or implicitly. Different approaches of encapsulating the VTN information in data packet can have different scalability implications.

One practical approach is to reuse some of the existing fields in the data packet to additionally identify the VTN the packet belongs to. For example, the destination IP addresses or the MPLS forwarding labels may be reused to further identify a VTN. This can avoid the cost of introducing new fields in the data packet, while since it introduces additional semantics to the existing fields, the processing of the existing fields in packet forwarding may need to be changed. Moreover, introducing VTN semantics to existing identifiers in the packet (e.g. IP addresses, MPLS forwarding labels, etc.) may result in the increase of the amount of the existing IDs in proportion to the number of the VTNs, which may cause scalability problem in networks where a relatively large number of VTNs is needed.



An alternative approach is to introduce a new dedicated field in the data packet for VTN identification. This could avoid the impacts to the existing fields in the packet. And if this new field carries a global-significant VTN identifier, it could be used together with the existing fields to determine the VTN-specific packet forwarding. The potential issue with this approach is the difficulty in introducing a new field in some of the data plane technologies.

In addition, the introduction of per VTN packet forwarding has impact on the scalability of the forwarding entries on network nodes, as a network node may need to maintain separate forwarding entries for each VTN it participates in.

### 3.3. Gap Analysis of Existing Mechanisms

One candidate mechanism to build VTN is to use VTN-specific Segment Routing (either SR-MPLS or SRv6) Identifiers in the data plane as described in [I-D.ietf-spring-sr-for-enhanced-vpn], and define and distribute the associated topology and resource attribute of each VTN based on either Multi-topology [I-D.ietf-lsr-isis-sr-vtn-mt], Flex-Algo [I-D.zhu-lsr-isis-sr-vtn-flexalgo] or the combination of these mechanisms in the control plane. This mechanism is suitable for networks where a small number of VTNs is needed. As the number of VTNs increases, there may be several scalability challenges with this approach:

1. The number of SR SIDs needed will increase in proportion to the number of VTNs in the network, which will bring challenges both to the distribution of SIDs and the related information in the control plane, and to the installation of forwarding entries for VTN-specific SIDs in the data plane.
2. The number of route computation (e.g. SPF computation) will increase in proportion to the number of VTNs in the network, which may introduce significant overhead to the control plane of network nodes.
3. The maximum number of logical topologies supported by OSPF is 128, and the maximum number of Flex-Algo is 128, which may not meet the required number of VTNs in some network scenarios.

### 4. Proposed Scalability Optimizations

#### 4.1. Control Plane Optimizations

For the distributed control plane, several optimizations can be considered to reduce the control plane overhead and improve the control plane scalability.

The first optimization mechanism is to reduce the amount of control plane sessions used for the establishment and maintenance of the VTNs. For multiple VTNs which have the same peering relationship between two adjacent network nodes, it is proposed that one single control protocol session is used for the establishment of multiple VTNs. The information of different VTNs can be exchanged over the same session, with necessary identification information to distinguish the VTNs in the control messages. This could reduce the overhead of maintaining a large number of control protocol sessions for different VTNs, and could also reduce the amount of control plane messages flooded in the network.

The second optimization mechanism is to decompose the attributes of a VTN into different groups, so that different types of VTN attribute can be advertised and processed separately in control plane. There are two basic types of attributes associated with a VTN: the topology attribute and the network resource attribute. In a network, it is possible that multiple VTNs share the same topology, and multiple VTNs may share the same set of network resources on particular network nodes and links. Then it is more efficient if only one copy of the topology information is advertised, and multiple VTNs sharing the same topology could refer to this topology information. More importantly, with this approach, the result of topology-based route computation could be shared by multiple VTNs, so that the overhead of per-VTN route computation could also be reduced. Similarly, information of a subset of network resources reserved on a particular network node or link could be advertised once and be referred to by multiple VTNs which share the same set of resources. This methodology could also apply to other attributes of VTN which may be introduced later and can be processed independently.

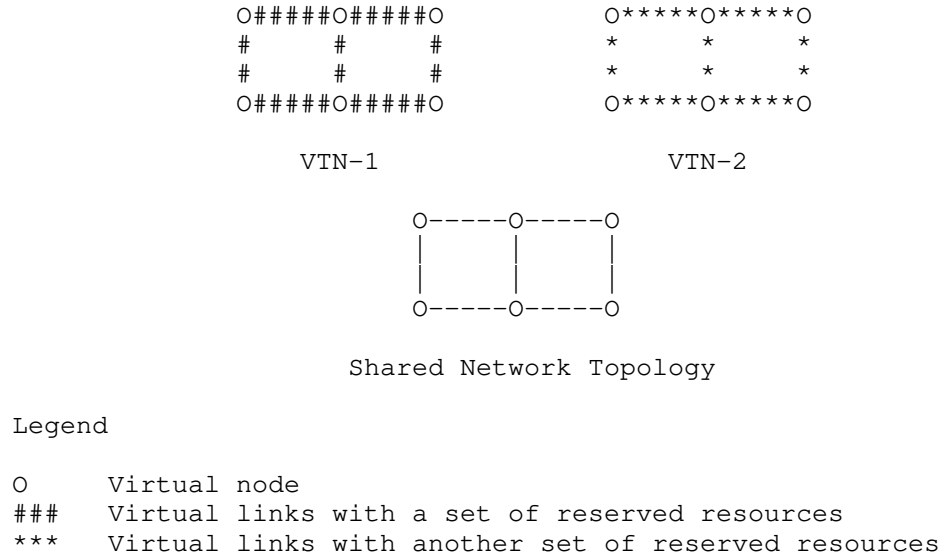


Figure 2. Topology Sharing between VTNs

Figure 1: FIG-2

Figure 2 gives an example of two VTNs which share the same logical topology. As shown in the figure, VTN-1 and VTN-2 are associated with the same topology, while the resource attributes of each VTN are different. In this case, only one copy of the network topology information needs to be advertised, and the topology-based route computation result can be shared by the two VTNs to generate the corresponding routing and forwarding tables.

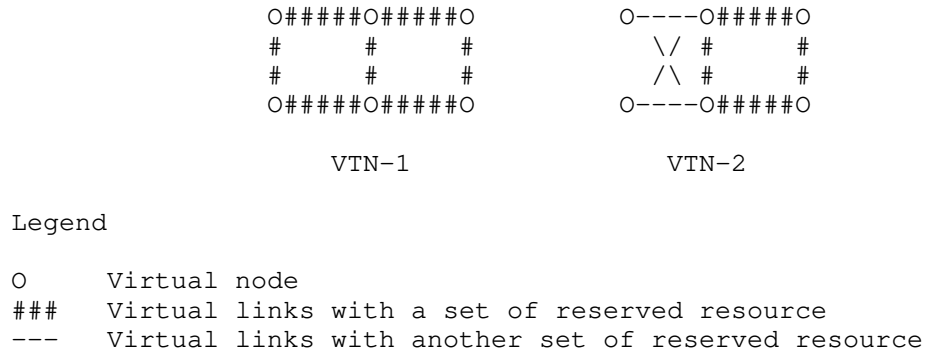


Figure 3. Resource Sharing between VTNs

Figure 3 gives another example of two VTNs which share the same set of network resources on some of the links. In this case, information about the resources allocated on each link only needs to be advertised once, then both VTN-1 and VTN-2 could refer to the reserved link resource for constraint based path computation.

For the optimization of the centralized control plane, it is suggested that the centralized controller is used as a complementary mechanism to the distributed control plane rather than a replacement, so that the workload for VTN specific path computation in control plane could be shared by both the centralized controller and the network nodes, and the scalability of both systems could be improved.

#### 4.2. Data Plane Optimizations

To support more VPN+ services while keeping the amount of data plane state at a reasonable scale, one typical approach is to classify a set of VPN+ services which have similar service characteristics and performance requirements into a group, and such group of VPN+ services are mapped to one VTN, which is allocated with an aggregated set of network resources and the union of the required logical topologies to meet the service requirement of the whole group of VPN+ services. Different groups of VPN+ services can be mapped to different VTNs with different set of network resources allocated. With appropriate grouping of VPN+ services, a reasonable number of VTNs with network resources reservation and aggregation could still meet the service requirements.

Another optimization in the data plane is to decouple the identifiers used for topology-based forwarding and the identifier used for the resource-specific processing introduced by VTN. One possible mechanism is to introduce a dedicated VTN Resource identifier in the packet header to uniquely identify the set of local network resources allocated to a VTN on each network node for the processing and forwarding of the received packets. Then the existing identifiers in the packet header used for topology based forwarding (e.g. the destination IP address, MPLS forwarding labels) are kept unchanged. The benefit is the amount of the existing topology-specific identifiers will not be impacted by the increasing number of VTNs. Since this new VTN Resource ID field will be used together with other existing fields to determine the VTN-specific packet forwarding, this may require network nodes to support a hierarchical forwarding table in data plane. Figure 4 shows the concept of using different data plane identifiers for topology-specific and resource-specific packet forwarding and processing in a VTN respectively.

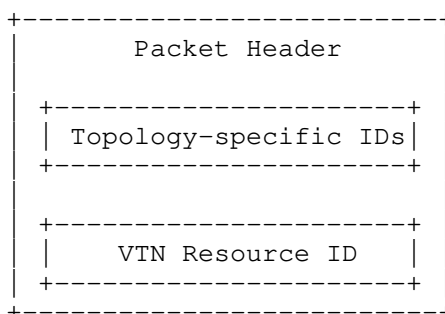


Figure 4. Decoupled Data Plane Topology and Resource Identifiers

In an IPv6 [RFC8200] based network, this could be achieved by introducing a dedicated field in either the IPv6 fixed header or the extension headers to carry the VTN resource identifier for the resource-specific forwarding, while keeping the destination IP address field used for routing towards the destination prefix in the corresponding topology. Note that the VTN resource ID needs to be parsed by every node along the path which is capable of VTN-specific forwarding. [I-D.dong-6man-enhanced-vpn-vtn-id] introduces the mechanism of carrying the VTN resource ID in IPv6 Hop-by-Hop extension header.

In an MPLS [RFC3032] based network, this may be achieved by introducing a dedicated VTN resource ID either in the MPLS label stack or following the MPLS label stack. This way, the existing MPLS forwarding labels could be used for topology-specific packet forwarding towards the destination node, and the VTN resource ID is used to determine the set of network resources for packet processing. This requires that both the forwarding label and the VTN Resource ID be parsed by nodes along the forwarding path of the packet, and the forwarding behavior may depend on the position of the VTN resource ID in the packet. The detailed extensions in MPLS data plane are out of the scope of this document.

## 5. Solution Evolution for Improved Scalability

Based on the analysis in this document, the control plane and data plane for VPN+ and VTN needs to evolve to support the increasing number of VPN+ services and the increasing number of VTNs in the network.

At the first step, by introducing resource-awareness to segment routing SIDs [I-D.ietf-spring-resource-aware-segments], and using Multi-Topology or Flex-Algo as the control plane, it could provide a solution for building a limited number of VTNs in the network to meet the requirement of a relatively small number of VPN+ services in the network. This mechanism is considered as the basic SR VTN.

As the required number of VPN+ services increases, more VTNs may be needed, then the control plane scalability could be improved by decoupling the topology attribute from the resource attribute and other attributes of VTN, so that multiple VTNs could share the same topology or resource attribute to reduce the control plane and data plane overhead. This mechanism is considered as the scalable SR VTN. Both the basic and the scalable SR VTN mechanisms are described in [I-D.ietf-spring-sr-for-enhanced-vpn].

If the data plane scalability becomes a concern, a dedicated VTN resource ID can be introduced in the data packet to decouple the topology-specific identifiers from the VTN resource identifiers in the data plane, this could help to reduce the number of SR SIDs needed to support a large number of VTNs. This mechanism is considered as the Resource-Independent (RI) VTN.

## 6. Security Considerations

This document describes the scalability considerations about the network control plane and data plane in enabling VPN+ services and the VTNs, and proposes several scalability optimization mechanisms. The security considerations in [I-D.ietf-teas-enhanced-vpn] applies to this document.

## 7. IANA Considerations

This document makes no request of IANA.

## 8. Contributors

Zhibo Hu  
Email: huzhibo@huawei.com

Hongjie Yang  
Email: hongjie.yang@huawei.com

## 9. Acknowledgments

The authors would like to thank Adrian Farrel for the review and discussion of this document.

## 10. References

### 10.1. Normative References

[I-D.ietf-teas-enhanced-vpn]  
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", Work in Progress, Internet-Draft, draft-ietf-teas-enhanced-vpn-08, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-enhanced-vpn-08.txt>>.

### 10.2. Informative References

[I-D.dong-6man-enhanced-vpn-vtn-id]  
Dong, J., Li, Z., Xie, C., Ma, C., and G. Mishra, "Carrying Virtual Transport Network Identifier in IPv6 Extension Header", Work in Progress, Internet-Draft, draft-dong-6man-enhanced-vpn-vtn-id-05, 8 September 2021, <<https://www.ietf.org/archive/id/draft-dong-6man-enhanced-vpn-vtn-id-05.txt>>.

[I-D.dong-lsr-sr-enhanced-vpn]  
Dong, J., Hu, Z., Li, Z., Tang, X., Pang, R., JooHeon, L., and S. Bryant, "IGP Extensions for Scalable Segment Routing based Enhanced VPN", Work in Progress, Internet-Draft, draft-dong-lsr-sr-enhanced-vpn-06, 11 July 2021, <<https://www.ietf.org/archive/id/draft-dong-lsr-sr-enhanced-vpn-06.txt>>.

[I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filmsils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", Work in Progress, Internet-Draft, draft-ietf-lsr-flex-algo-17, 6 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-lsr-flex-algo-17.txt>>.

[I-D.ietf-lsr-isis-sr-vtn-mt]  
Xie, C., Ma, C., Dong, J., and Z. Li, "Using IS-IS Multi-Topology (MT) for Segment Routing based Virtual Transport Network", Work in Progress, Internet-Draft, draft-ietf-lsr-isis-sr-vtn-mt-01, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-lsr-isis-sr-vtn-mt-01.txt>>.

[I-D.ietf-spring-resource-aware-segments]  
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Introducing Resource Awareness to SR

Segments", Work in Progress, Internet-Draft, draft-ietf-spring-resource-aware-segments-03, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-resource-aware-segments-03.txt>>.

[I-D.ietf-spring-sr-for-enhanced-vpn]

Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Segment Routing based Virtual Transport Network (VTN) for Enhanced VPN", Work in Progress, Internet-Draft, draft-ietf-spring-sr-for-enhanced-vpn-01, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-sr-for-enhanced-vpn-01.txt>>.

[I-D.ietf-teas-ietf-network-slices]

Farrel, A., Gray, E., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-04, 23 August 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-04.txt>>.

[I-D.zhu-lsr-isis-sr-vtn-flexalgo]

Zhu, Y., Dong, J., and Z. Hu, "Using Flex-Algo for Segment Routing based VTN", Work in Progress, Internet-Draft, draft-zhu-lsr-isis-sr-vtn-flexalgo-03, 11 July 2021, <<https://www.ietf.org/archive/id/draft-zhu-lsr-isis-sr-vtn-flexalgo-03.txt>>.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <<https://www.rfc-editor.org/info/rfc4915>>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.



[TS23501] "3GPP TS23.501", 2016,  
<[https://portal.3gpp.org/desktopmodules/Specifications/  
SpecificationDetails.aspx?specificationId=3144](https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144)>.

#### Authors' Addresses

Jie Dong  
Huawei Technologies  
Huawei Campus, No. 156 Beiqing Road  
Beijing  
100095  
China

Email: jie.dong@huawei.com

Zhenbin Li  
Huawei Technologies  
Huawei Campus, No. 156 Beiqing Road  
Beijing  
100095  
China

Email: lizhenbin@huawei.com

Liyan Gong  
China Mobile  
No. 32 Xuanwumenxi Ave., Xicheng District  
Beijing  
China

Email: gongliyan@chinamobile.com

Guangming Yang  
China Telecom  
No.109 West Zhongshan Ave., Tianhe District  
Guangzhou  
China

Email: yangguangm@chinatelecom.cn

James N Guichard  
Futurewei Technologies  
2330 Central Express Way  
Santa Clara,  
United States of America

Email: james.n.guichard@futurewei.com

Gyan Mishra  
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Fengwei Qin  
China Mobile  
No. 32 Xuanwumenxi Ave., Xicheng District  
Beijing  
China

Email: qinfengwei@chinamobile.com

TEAS Working Group  
Internet Draft  
Intended status: Informational

Fabio Peruzzini  
TIM  
Jean-Francois Bouquier  
Vodafone  
Italo Busi  
Huawei  
Daniel King  
Old Dog Consulting  
Daniele Ceccarelli  
Ericsson

Expires: January 2022

July 12, 2021

Applicability of Abstraction and Control of Traffic Engineered  
Networks (ACTN) to Packet Optical Integration (POI)

draft-ietf-teas-actn-poi-applicability-03

#### Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on April 9, 2021.

#### Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Abstract

This document considers the applicability of Abstraction and Control of TE Networks (ACTN) architecture to Packet Optical Integration (POI) in the context of IP/MPLS and Optical internetworking. It identifies the YANG data models being defined by the IETF to support this deployment architecture and specific scenarios relevant for Service Providers.

Existing IETF protocols and data models are identified for each multi-layer (packet over optical) scenario with a specific focus on the MPI (Multi-Domain Service Coordinator to Provisioning Network Controllers Interface) in the ACTN architecture.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction.....  | 3  |
| 2. Reference architecture and network scenario.....                     | 4  |
| 2.1. L2/L3VPN Service Request North Bound of MDSC.....                  | 9  |
| 2.2. Service and Network Orchestration.....                             | 10 |
| 2.2.1. Hard Isolation.....  | 13 |
| 2.2.2. Shared Tunnel Selection.....                                     | 13 |
| 2.3. IP/MPLS Domain Controller and NE Functions.....                    | 14 |
| 2.4. Optical Domain Controller and NE Functions.....                    | 15 |
| 3. Interface protocols and YANG data models for the MPIs.....           | 16 |
| 3.1. RESTCONF protocol at the MPIs.....                                 | 16 |
| 3.2. YANG data models at the MPIs.....                                  | 16 |
| 3.2.1. Common YANG data models at the MPIs.....                         | 16 |
| 3.2.2. YANG models at the Optical MPIs.....                             | 17 |
| 3.2.3. YANG data models at the Packet MPIs.....                         | 18 |
| 3.3. PCEP.....  | 19 |
| 4. Multi-layer and multi-domain services scenarios.....                 | 20 |
| 4.1. Scenario 1: inventory, service and network topology discovery..... | 21 |
| 4.1.1. Inter-domain link discovery.....                                 | 22 |
| 4.1.2. Multi-layer IP Link discovery.....                               | 23 |

|   |    |
|---|----|
| 4.1.3. Inventory discovery.....                             | 23 |
| 4.1.4. SR-TE paths discovery.....                           | 24 |
| 4.2. Establishment of L2VPN/L3VPN with TE requirements..... | 24 |
| 4.2.1. Optical Path Computation.....                        | 29 |
| 4.2.2. Multi-layer IP Link Setup and Update.....            | 29 |
| 4.2.3. SR-TE Path Setup and Update.....                     | 30 |
| 5. Security Considerations.....                             | 30 |
| 6. Operational Considerations.....                          | 31 |
| 7. IANA Considerations.....                                 | 31 |
| 8. References.....  | 31 |
| 8.1. Normative References.....                              | 31 |
| 8.2. Informative References.....                            | 33 |
| Appendix A. Multi-layer and multi-domain resiliency.....    | 35 |
| A.1. Maintenance Window.....                                | 35 |
| A.2. Router port failure.....                               | 35 |
| Acknowledgments.....  | 36 |
| Contributors.....   | 36 |
| Authors' Addresses.....                                     | 38 |

## 1. Introduction

The complete automation of the management and control of Service Providers transport networks (IP/MPLS, optical, and microwave transport networks) is vital for meeting emerging demand for high-bandwidth use cases, including 5G and fiber connectivity services. The Abstraction and Control of TE Networks (ACTN) architecture and interfaces facilitate the automation and operation of complex Optical and IP/MPLS networks through standard interfaces and data models. Thus allowing a wide range of transport connectivity services that can be requested by the upper layers fulfilling almost any kind of service level requirements from a network perspective (e.g. physical diversity, latency, bandwidth, topology, etc.)

Packet Optical Integration (POI) is an advanced use case of traffic engineering. In wide-area networks, a packet network based on the Internet Protocol (IP), and often Multiprotocol Label Switching (MPLS), is typically realized on top of an optical transport network that uses Dense Wavelength Division Multiplexing (DWDM) (and optionally an Optical Transport Network (OTN) layer).

In many existing network deployments, the packet and the optical networks are engineered and operated independently. As a result, there are technical differences between the technologies (e.g., routers compared to optical switches) and the corresponding network engineering and planning methods (e.g., inter-domain peering optimization in IP, versus dealing with physical impairments in

DWDM, or very different time scales). In addition, customers needs can be different between a packet and an optical network, and it is not uncommon to use different vendors in both domains. The operation of these complex packet and optical networks is often siloed, as these technology domains require specific skills sets.

The packet/optical network deployment and operation separation are inefficient for many reasons. Both capital expenditure (CAPEX) and operational expenditure (OPEX) could be significantly reduced by integrating the packet and the optical network. Multi-layer online topology insight can speed up troubleshooting (e.g., alarm correlation) and network operation (e.g., coordination of maintenance events), multi-layer offline topology inventory can improve service quality (e.g., detection of diversity constraint violations) and multi-layer traffic engineering can use the available network capacity more efficiently (e.g., coordination of restoration). In addition, provisioning workflows can be simplified or automated as needed across layers (e.g., to achieve bandwidth-on-demand or to perform maintenance events).

ACTN framework enables this complete multi-layer and multi-vendor integration of packet and optical networks through MDSC and packet and optical PNCs.

In this document, critical scenarios for POI are described from the packet service layer perspective and identify the required coordination between packet and optical layers to improve POI deployment and operation. Precise definitions of scenarios can help with achieving a common understanding across different disciplines. The focus of the scenarios are IP/MPLS networks operated as a client of optical DWDM networks. The scenarios are ordered by increasing the level of integration and complexity. For each multi-layer scenario, the document analyzes how to use the interfaces and data models of the ACTN architecture.

Understanding the level of standardization and the possible gaps will help assess the feasibility of integration between IP and Optical DWDM domain (and optionally OTN layer) in an end-to-end multi-vendor service provisioning perspective.

## 2. Reference architecture and network scenario

This document analyses several deployment scenarios for Packet and Optical Integration (POI) in which ACTN hierarchy is deployed to control a multi-layer and multi-domain network, with two Optical domains and two Packet domains, as shown in Figure 1:

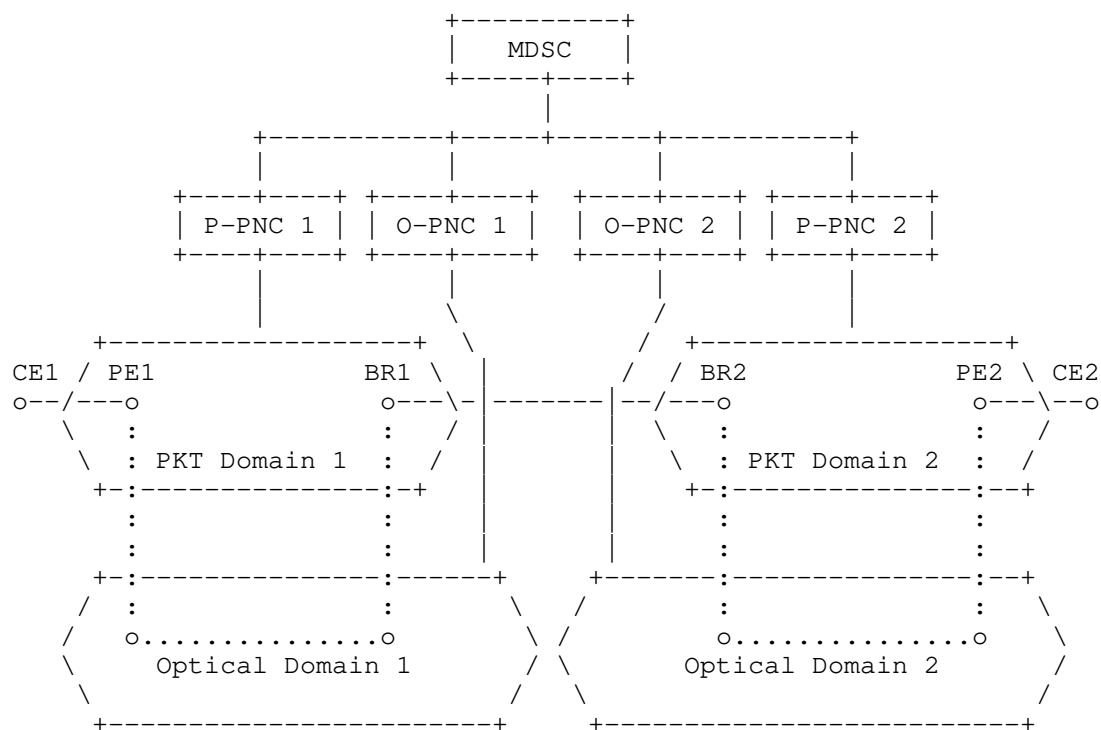


Figure 1 - Reference Scenario

The ACTN architecture, defined in [RFC8453], is used to control this multi-domain network where each Packet PNC (P-PNC) is responsible for controlling its IP domain, which can be either an Autonomous System (AS), [RFC1930], or an IGP area within the same operator network. Each Optical PNC (O-PNC) in the above topology is responsible for controlling its Optical Domain.

The routers between IP domains can be either AS Boundary Routers (ASBR) or Area Border Router (ABR): in this document, the generic term Border Router (BR) is used to represent either an ASBR or a ABR.

The MDSC is responsible for coordinating the whole multi-domain multi-layer (Packet and Optical) network. A specific standard

interface (MPI) permits MDSC to interact with the different Provisioning Network Controller (O/P-PNCs).

The MPI interface presents an abstracted topology to MDSC hiding technology-specific aspects of the network and hiding topology details depending on the policy chosen regarding the level of abstraction supported. The level of abstraction can be obtained based on P-PNC and O-PNC configuration parameters (e.g. provide the potential connectivity between any PE and any BR in an MPLS-TE network).

In the network scenario of Figure 1, it is assumed that:

- o The domain boundaries between the IP and Optical domains are congruent. In other words, one Optical domain supports connectivity between Routers in one and only one Packet Domain;
- o Inter-domain links exist only between Packet domains (i.e., between BR routers) and between Packet and Optical domains (i.e., between routers and Optical NEs). In other words, there are no inter-domain links between Optical domains;
- o The interfaces between the Routers and the Optical NEs are "Ethernet" physical interfaces;
- o The interfaces between the Border Routers (BRs) are "Ethernet" physical interfaces.

This version of the document assumes that the IP Link supported by the Optical network are always intra-AS (PE-BR, intra-domain BR-BR, PE-P, BR-P, or P-P) and that the BRs are co-located and connected by an IP Link supported by an Ethernet physical link.

The possibility to setup inter-AS/inter-area IP Links (e.g., inter-domain BR-BR or PE-PE), supported by optical network, is for further study.

Therefore, if inter-domain links between the Optical domains exist, they would be used to support multi-domain Optical services, which are outside the scope of this document.

The Optical NEs within the optical domains can be ROADMs or OTN switches, with or without a ROADM.



The MDSC in Figure 1 is responsible for multi-domain and multi-layer coordination across multiple Packet and Optical domains, as well as to provide L2/L3VPN services.

Although the new optical technologies (e.g. QSFP-DD ZR 400G) providing DWDM pluggable interfaces on the Routers, the deployment of those pluggable optics is not yet widely adopted by the operators. The reason is that most operators are not yet ready to manage Packet and Transport networks in a single unified domain. As a consequence, this draft is not addressing the unified scenario. Instead, the unified use case will be described in a different draft.

From an implementation perspective, the functions associated with MDSC and described in [RFC8453] may be grouped in different ways.

1. Both the service- and network-related functions are collapsed into a single, monolithic implementation, dealing with the end customer service requests received from the CMI (Customer MDSC Interface) and adapting the relevant network models. An example is represented in Figure 2 of [RFC8453]
2. An implementation can choose to split the service-related and the network-related functions into different functional entities, as described in [RFC8309] and in section 4.2 of [RFC8453]. In this case, MDSC is decomposed into a top-level Service Orchestrator, interfacing the customer via the CMI, and into a Network Orchestrator interfacing at the southbound with the PNCs. The interface between the Service Orchestrator and the Network Orchestrator is not specified in [RFC8453].
3. Another implementation can choose to split the MDSC functions between an H-MDSC responsible for packet-optical multi-layer coordination, interfacing with one Optical L-MDSC, providing multi-domain coordination between the O-PNCs and one Packet L-MDSC, providing multi-domain coordination between the P-PNCs (see for example Figure 9 of [RFC8453]).
4. Another implementation can also choose to combine the MDSC and the P-PNC functions together.

Please note that in the current service provider's network deployments, at the North Bound of the MDSC, instead of a CNC, typically there is an OSS/Orchestration layer. In this case, the MDSC would implement only the Network Orchestration functions, as in [RFC8309] and described in point 2 above. In this case, the MDSC is dealing with the network services requests received from the OSS/Orchestration layer.

[Editors'note:] Check for a better term to define the network services. It may be worthwhile defining what are the customer and network services.

The OSS/Orchestration layer is a vital part of the architecture framework for a service provider:

- o to abstract (through MDSC and PNCs) the underlying transport network complexity to the Business Systems Support layer;
- o to coordinate NFV, Transport (e.g. IP, Optical and Microwave networks), Fixed Access, Core and Radio domains enabling full automation of end-to-end services to the end customers;
- o to enable catalogue-driven service provisioning from external applications (e.g. Customer Portal for Enterprise Business services), orchestrating the design and lifecycle management of these end-to-end transport connectivity services, consuming IP and/or Optical transport connectivity services upon request.

The functionality of the OSS/Orchestration layer and the interface toward the MDSC are usually operator-specific and outside the scope of this draft. For example, this document assumes that the OSS/Orchestrator requests MDSC to set up L2VPN/L3VPN services through mechanisms that are outside the scope of this document.

There are two prominent cases when MDSC coordination of underlying PNCs for POI networking is initiated:

- o Initiated by a request from the OSS/Orchestration layer to setup L2VPN/L3VPN services that requires multi-layer/multi-domain coordination;
- o Initiated by the MDSC itself to perform multi-layer/multi-domain optimizations and/or maintenance activities (e.g. rerouting LSPs with their associated services when putting a resource, like a fibre, in maintenance mode during a maintenance window). Unlike service fulfillment, these workflows are not related to a service provisioning request being received from the OSS/Orchestration layer.

The two aforementioned MDSC workflow cases are in the scope of this draft. The workflow initiation is transparent at the MPI.

### 2.1. L2/L3VPN Service Request North Bound of MDSC

As explained in section 2, the OSS/Orchestration layer can request the MDSC to setup L2/L3VPN services (with or without TE requirements).

Although the OSS/Orchestration layer interface is usually operator-specific, typically it would be using a RESTCONF/YANG interface with a more abstracted version of the MPI YANG data models used for network configuration (e.g. L3NM, L2NM).

Figure 2 shows an example of possible control flow between the OSS/Orchestration layer and the MDSC to instantiate L2/L3VPN services, using the YANG models under the definition in [VN], [L2NM], [L3NM] and [TSM].

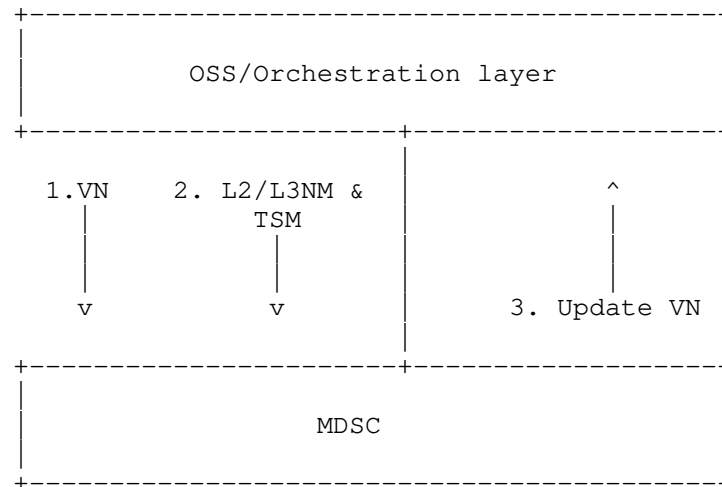


Figure 2 Service Request Process

- o The VN YANG model [VN], whose primary focus is the CMI, can also provide VN Service configuration from an orchestrated connectivity service point of view when the L2/L3VPN service has TE requirements. However, this model is not used to setup L2/L3VPN service with no TE requirements.

- o It provides the profile of VN in terms of VN members, each of which corresponds to an edge-to-edge link between customer end-points (VNAPs). It also provides the mappings between the VNAPs with the LTPs and the connectivity matrix with the VN member. The associated traffic matrix (e.g., bandwidth, latency, protection level, etc.) of VN member is expressed (i.e., via the TE-topology's connectivity matrix).
- o The model also provides VN-level preference information (e.g., VN member diversity) and VN-level admin-status and operational-status.
- o The L2NM YANG model [L2NM], whose primary focus is the MPI, can also be used to provide L2VPN service configuration and site information, from a orchestrated connectivity service point of view.
- o The L3NM YANG model [L3NM], whose primary focus is the MPI, can also be used to provide all L3VPN service configuration and site information, from a orchestrated connectivity service point of view.
- o The TE & Service Mapping YANG model [TSM] provides TE-service mapping as well as site mapping.
  - o TE-service mapping provides the mapping between a L2/L3VPN instance and the corresponding VN instances.
  - o The TE-service mapping also provides the service mapping requirement type as to how each L2/L3VPN/VN instance is created concerning the underlay TE tunnels (e.g., whether they require a new and isolated set of TE underlay tunnels or not). See Section 2.2 for a detailed discussion on the mapping requirement types.
  - o Site mapping provides the site reference information across L2/L3VPN Site ID, VN Access Point ID, and the LTP of the access link.

## 2.2. Service and Network Orchestration

From a functional standpoint, MDSC represented in Figure 2 interfaces with the OSS/Orchestration layer and decoupled L2/L3VPN service configuration functions from network configuration functions. Therefore in this document, the MDSC performs the functions of the Network Orchestrator, as defined in [RFC 8309].

One of the important MDSC functions is to identify which TE Tunnels should carry the L2/L3VPN traffic (e.g., from TE & Service Mapping configuration) and to relay this information to the P-PNCs, to ensure the PEs' forwarding tables (e.g., VRF) are properly populated, according to the TE binding requirement for the L2/L3VPN.

TE binding requirement types [TSM] are:

1. Hard Isolation with deterministic latency: The L2/L3VPN service requires a set of dedicated TE Tunnels providing deterministic latency performances and that cannot be not shared with other services, nor compete for bandwidth with other Tunnels.
2. Hard Isolation: This is similar to the above case without deterministic latency requirements.
3. Soft Isolation: The L2/L3VPN service requires a set of dedicated MPLS-TE tunnels that cannot be shared with other services, but which could compete for bandwidth with other Tunnels.
4. Sharing: The L2/L3VPN service allows sharing the MPLS-TE Tunnels supporting it with other services.

There could be additional TE binding requirements for the first three types with respect to different VN members of the same VN (on how different VN members, belonging to the same VN, can share or not network resources). For the first two cases, VN members can be hard-isolated, soft-isolated, or shared. For the third case, VN members can be soft-isolated or shared.

In order to fulfil the L2/L3VPN end-to-end TE requirements, including the TE binding requirements, the MDSC needs to perform multi-layer/multi-domain path computation to select the BRs, the intra-domain MPLS-TE Tunnels and the intra-domain Optical Tunnels.

Depending on the knowledge that MDSC has of the topology and configuration of the underlying network domains, three models for performing path computation are possible:

1. Summarization: MDSC has an abstracted TE topology view of all of the underlying domains, both packet and optical. MDSC does not have enough TE topology information to perform multi-layer/multi-domain path computation. Therefore MDSC delegates the P-PNCs and O-PNCs to perform a local path computation within their controlled domains and it uses the information returned by the P-PNCs and O-PNCs to compute the optimal multi-domain/multi-layer path. This model presents an issue to P-PNC, which does not have the capability of performing a single-domain/multi-layer path computation (that is, P-PNC does not have any possibility to retrieve the topology/configuration information from the Optical controller). A possible solution could be to include a CNC function in the P-PNC to request the MDSC multi-domain Optical path computation, as shown in Figure 10 of [RFC8453].
2. Partial summarization: MDSC has full visibility of the TE topology of the packet network domains and an abstracted view of the TE topology of the optical network domains. MDSC then has only the capability of performing multi-domain/single-layer path computation for the packet layer (the path can be computed optimally for the two packet domains). Therefore MDSC still needs to delegate the O-PNCs to perform local path computation within their respective domains and it uses the information received by the O-PNCs, together with its TE topology view of the multi-domain packet layer, to perform multi-layer/multi-domain path computation. The role of P-PNC is minimized, i.e. is limited to management.
3. Full knowledge: MDSC has the complete and enough detailed view of the TE topology of all the network domains (both optical and packet). In such case MDSC has all the information needed to perform multi-domain/multi-layer path computation, without relying on PNCs.

This model may present, as a potential drawback, scalability issues and, as discussed in section 2.2. of [PATH-COMPUTE], performing path computation for optical networks in the MDSC is quite challenging because the optimal paths depend also on vendor-specific optical attributes (which may be different in the two domains if they are provided by different vendors).

The current version of this draft assumes that MDSC supports at least model #2 (Partial summarization).

[Note: check with operators for some references on real deployment]

### 2.2.1. Hard Isolation

For example, when "Hard Isolation with, or without, deterministic latency" TE binding requirement is applied for a L2/L3VPN, new Optical Tunnels need to be setup to support dedicated IP Links between PEs and BRs.

The MDSC needs to identify the set of IP/MPLS domains and their BRs. This requires the MDSC to request each O-PNC to compute the intra-domain optical paths between each PEs/BRs pairs.

When requesting optical path computation to the O-PNC, the MDSC needs to take into account the inter-layer peering points, such as the interconnections between the PE/BR nodes and the edge Optical nodes (e.g., using the inter-layer link or the transitional link information, defined in [RFC8795]).

When the optimal multi-layer/multi-domain path has been computed, the MDSC requests each O-PNC to setup the selected Optical Tunnels and P-PNC to setup the intra-domain MPLS-TE Tunnels, over the selected Optical Tunnels. MDSC also properly configures its BGP speakers and PE/BR forwarding tables to ensure that the VPN traffic is properly forwarded.

### 2.2.2. Shared Tunnel Selection

In case of shared tunnel selection, the MDSC needs to check if there is a multi-domain path which can support the L2/L3VPN end-to-end TE service requirements (e.g., bandwidth, latency, etc.) using existing intra-domain MPLS-TE tunnels.

If such a path is found, the MDSC selects the optimal path from the candidate pool and request each P-PNC to setup the L2/L3VPN service using the selected intra-domain MPLS-TE tunnel, between PE/BR nodes.

Otherwise, the MDSC should detect if the multi-domain path can be setup using existing intra-domain MPLS-TE tunnels with modifications (e.g., increasing the tunnel bandwidth) or setting up new intra-domain MPLS-TE tunnel(s).

The modification of an existing MPLS-TE Tunnel and the setup of a new MPLS-TE Tunnel may also require multi-layer coordination e.g., in case the available bandwidth of underlying Optical Tunnels is not sufficient. Based on multi-domain/multi-layer path computation, the MDSC can decide for example to modify the bandwidth of an existing Optical Tunnel (e.g., ODUflex bandwidth increase) or to setup new

Optical Tunnels to be used as additional LAG members of an existing IP Link or as new IP Links to re-route the MPLS-TE Tunnel.

In all the cases, the labels used by the end-to-end tunnel are distributed in the PE and BR nodes by BGP. The MDSC is responsible to configure the BGP speakers in each P-PNC, if needed.

### 2.3. IP/MPLS Domain Controller and NE Functions

IP/MPLS networks are assumed to have multiple domains. Each domain, corresponding to either an IGP area or an Autonomous System (AS) within the same operator network, is controlled by an IP/MPLS domain controller (P-PNC).

Among the functions of the P-PNC, there are the setup or modification of the intra-domain MPLS-TE Tunnels, between PEs and BRs, and the configuration of the VPN services, such as the VRF in the PE nodes, as shown in Figure 3:

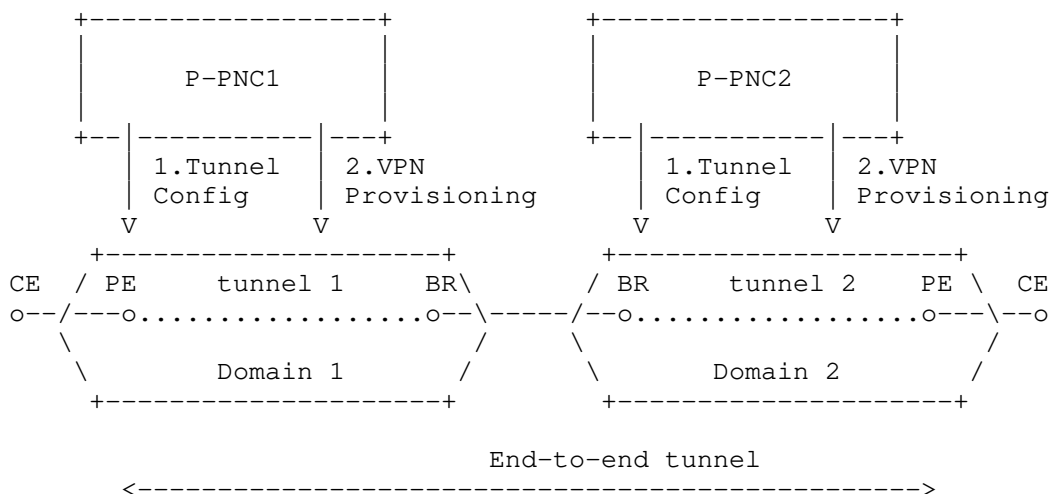


Figure 3 IP/MPLS Domain Controller & NE Functions

It is assumed that BGP is running in the inter-domain IP/MPLS networks for L2/L3VPN. The P-PNC controller is also responsible for configuring the BGP speakers within its control domain, if necessary.



The BGP would be responsible for the end-to-end tunnel label distribution on PE and BR nodes. The MDSC is responsible for selecting the BRs and the intra-domain MPLS-TE Tunnels between PE/BR nodes.

If new MPLS-TE Tunnels are needed or modifications (e.g., bandwidth increase) to existing MPLS-TE Tunnels are needed, as outlined in section 2.2, the MDSC would request their setup or modifications to the P-PNCs (step 1 in Figure 3). Then the MDSC would request the P-PNC to configure the VPN, including selecting the intra-domain TE Tunnel (step 2 in Figure 3).

The P-PNC should configure, using mechanisms outside the scope of this document, the ingress PE forwarding table, e.g., the VRF, to forward the VPN traffic, received from the CE, with the following three labels:

- o VPN label: assigned by the egress PE and distributed by BGP;
- o end-to-end LSP label: assigned by the egress BR, selected by the MDSC, and distributed by BGP;
- o MPLS-TE tunnel label, assigned by the next hop P node of the tunnel selected by the MDSC and distributed by mechanism internal to the IP/MPLS domain (e.g., RSVP-TE).

#### 2.4. Optical Domain Controller and NE Functions

The optical network provides the underlay connectivity services to IP/MPLS networks. The coordination of Packet/Optical multi-layer is done by the MDSC, as shown in Figure 1.

The O-PNC is responsible to:

- o provide to the MDSC an abstract TE topology view of its underlying optical network resources;
- o perform single-domain local path computation, when requested by the MDSC;
- o perform Optical Tunnel setup, when requested by the MDSC.

The mechanisms used by O-PNC to perform intra-domain topology discovery and path setup are usually vendor-specific and outside the scope of this document.

Depending on the type of optical network, TE topology abstraction, path computation and path setup can be single-layer (either OTN or WDM) or multi-layer OTN/WDM. In the latter case, the multi-layer coordination between the OTN and WDM layers is performed by the O-PNC.

### 3. Interface protocols and YANG data models for the MPIs

This section describes general assumptions applicable at all the MPI interfaces, between each PNC (Optical or Packet) and the MDSC, and all the scenarios discussed in this document.

#### 3.1. RESTCONF protocol at the MPIs

The RESTCONF protocol, as defined in [RFC8040], using the JSON representation defined in [RFC7951], is assumed to be used at these interfaces. In addition, extensions to RESTCONF, as defined in [RFC8527], to be compliant with Network Management Datastore Architecture (NMDA) defined in [RFC8342], are assumed to be used as well at these MPI interfaces and also at CMI interfaces.

#### 3.2. YANG data models at the MPIs

The data models used on these interfaces are assumed to use the YANG 1.1 Data Modeling Language, as defined in [RFC7950].

##### 3.2.1. Common YANG data models at the MPIs

As required in [RFC8040], the "ietf-yang-library" YANG module defined in [RFC8525] is used to allow the MDSC to discover the set of YANG modules supported by each PNC at its MPI.

Both Optical and Packet PNCs use the following common topology YANG models at the MPI to report their abstract topologies:

- o The Base Network Model, defined in the "ietf-network" YANG module of [RFC8345];
- o The Base Network Topology Model, defined in the "ietf-network-topology" YANG module of [RFC8345], which augments the Base Network Model;
- o The TE Topology Model, defined in the "ietf-te-topology" YANG module of [RFC8795], which augments the Base Network Topology Model with TE specific information.

These common YANG models are generic and augmented by technology-specific YANG modules as described in the following sections.

Both Optical and Packet PNCs must use the following common notifications YANG models at the MPI so that any network changes can be reported almost in real-time to MDSC by the PNCs:

- o Dynamic Subscription to YANG Events and Datastores over RESTCONF as defined in [RFC8650];
- o Subscription to YANG Notifications for Datastores updates as defined in [RFC8641].

PNCs and MDSCs must be compliant with subscription requirements as stated in [RFC7923].

### 3.2.2. YANG models at the Optical MPIS

The Optical PNC also uses at least the following technology-specific topology YANG models, providing WDM and Ethernet technology-specific augmentations of the generic TE Topology Model:

- o The WSON Topology Model, defined in the "ietf-wson-topology" YANG modules of [WSON-TOPO], or the Flexi-grid Topology Model, defined in the "ietf-flexi-grid-topology" YANG module of [Flexi-TOPO];
- o Optionally, when the OTN layer is used, the OTN Topology Model, as defined in the "ietf-otn-topology" YANG module of [OTN-TOPO];
- o The Ethernet Topology Model, defined in the "ietf-eth-te-topology" YANG module of [CLIENT-TOPO];
- o Optionally, when the OTN layer is used, the network data model for L1 OTN services (e.g. an Ethernet transparent service) as defined in "ietf-trans-client-service" YANG module of draft-ietf-ccamp-client-signal-yang [CLIENT-SIGNAL];
- o The WSON Topology Model or, alternatively, the Flexi-grid Topology model is used to report the DWDM network topology (e.g., ROADMs and links) depending on whether the DWDM optical network is based on fixed grid or flexible-grid.

The Ethernet Topology is used to report the access links between the IP routers and the edge ROADMs.

The optical PNC uses at least the following YANG models:

- o The TE Tunnel Model, defined in the "ietf-te" YANG module of [TE-TUNNEL];
- o The WSON Tunnel Model, defined in the "ietf-wson-tunnel" YANG modules of [WSON-TUNNEL], or the Flexi-grid Media Channel Model, defined in the "ietf-flexi-grid-media-channel" YANG module of [Flexi-MC];
- o Optionally, when the OTN layer is used, the OTN Tunnel Model, defined in the "ietf-otn-tunnel" YANG module of [OTN-TUNNEL];
- o The Ethernet Client Signal Model, defined in the "ietf-eth-tran-service" YANG module of [CLIENT-SIGNAL].

The TE Tunnel model is generic and augmented by technology-specific models such as the WSON Tunnel Model and the Flexi-grid Media Channel Model.

The WSON Tunnel Model, or the Flexi-grid Media Channel Model, may be used to setup connectivity within the DWDM network depending on whether the DWDM optical network is based on fixed grid or flexible-grid.

The Ethernet Client Signal Model is used to configure the steering of the Ethernet client traffic between Ethernet access links and TE Tunnels, which in this case could be either WSON Tunnels or Flexi-Grid Media Channels. This model is generic and applies to any technology-specific TE Tunnel: technology-specific attributes are provided by the technology-specific models which augment the generic TE-Tunnel Model.

### 3.2.3. YANG data models at the Packet MPIs

The Packet PNC also uses at least the following technology-specific topology YANG models, providing IP and Ethernet technology-specific augmentations of the generic Topology Models described in section 3.2.1:

- o The L3 Topology Model, defined in the "ietf-l3-unicast-topology" YANG module of [RFC8346], which augments the Base Network Topology Model;
- o The L3 specific data model including extended TE attributes (e.g. performance derived metrics like latency), defined in "ietf-l3-te-topology" and in "ietf-te-topology-packet" YANG modules of [L3-TE-TOPO];

- o When SR-TE is used, the SR Topology Model, defined in the "ietf-sr-mpls-topology" YANG module of [SR-TE-TOPO]: this YANG module is used together with other YANG modules to provide the SR-TE topology view as described in figure 2 of [SR-TE-TOPO];
- o The Ethernet Topology Model, defined in the "ietf-eth-te-topology" YANG module of [CLIENT-TOPO], which augments the TE Topology Model.

The Ethernet Topology Model is used to report the access links between the IP routers and the edge ROADMs as well as the inter-domain links between ASBRs, while the L3 Topology Model is used to report the IP network topology (e.g., IP routers and links).

- o The User Network Interface (UNI) Topology Model, being defined in the "ietf-uni-topology" module of the draft-ogondio-opsawg-uni-topology [UNI-TOPO] which augment "ietf-network" module defined in [RFC8345] adding service attachment points to the nodes to which L2VPN/L3VPN IP/MPLS services can be attached.
- o L3VPN network data model defined in "ietf-l3vpn-ntw" module of draft-ietf-opsawg-l3sm-l3nm [L3NM] used for non-ACTN MPI for L3VPN service provisioning
- o L2VPN network data model defined in "ietf-l2vpn-ntw" module of draft-ietf-barguil-opsawg-l2sm-l2nm [L2NM] used for non-ACTN MPI for L2VPN service provisioning

[Editor's note:] Add YANG models used for tunnel and service configuration.

### 3.3. PCEP

[RFC8637] examines the applicability of a Path Computation Element (PCE) [RFC5440] and PCE Communication Protocol (PCEP) to the ACTN framework. It further describes how the PCE architecture applies to ACTN and lists the PCEP extensions that are needed to use PCEP as an ACTN interface. The stateful PCE [RFC8231], PCE-Initiation [RFC8281], stateful Hierarchical PCE (H-PCE) [RFC8751], and PCE as a central controller (PCECC) [RFC8283] are some of the key extensions that enable the use of PCE/PCEP for ACTN.

Since the PCEP supports path computation in the packet and optical networks, PCEP is well suited for inter-layer path computation. [RFC5623] describes a framework for applying the PCE-based architecture to interlayer (G)MPLS traffic engineering. Furthermore,

the section 6.1 of [RFC8751] states the H-PCE applicability for inter-layer or POI.

[RFC8637] lists various PCEP extensions that apply to ACTN. It also list the PCEP extension for optical network and POI.

Note that the PCEP can be used in conjunction with the YANG models described in the rest of this document. Depending on whether ACTN is deployed in a greenfield or brownfield, two options are possible:

1. The MDSC uses a single RESTCONF/YANG interface towards each PNC to discover all the TE information and request TE tunnels. It may either perform full multi-layer path computation or delegate path computation to the underneath PNCs.

This approach is desirable for operators from an multi-vendor integration perspective as it is simple, and we need only one type of interface (RESTCONF) and use the relevant YANG data models depending on the operator use case considered. Benefits of having only one protocol for the MPI between MDSC and PNC have been already highlighted in [PATH-COMPUTE].

2. The MDSC uses the RESTCONF/YANG interface towards each PNC to discover all the TE information and requests the creation of TE tunnels. However, it uses PCEP for hierarchical path computation.

As mentioned in Option 1, from an operator perspective, this option can add integration complexity to have two protocols instead of one, unless the RESTCONF/YANG interface is added to an existing PCEP deployment (brownfield scenario).

Section 4 of this draft analyses the case where a single RESTCONF/YANG interface is deployed at the MPI (i.e., option 1 above).

#### 4. Multi-layer and multi-domain services scenarios

Multi-layer and multi-domain scenarios, based on reference network described in section 2, and very relevant for Service Providers, are described in the next sections. For each scenario, existing IETF protocols and data models are identified with particular focus on the MPI in the ACTN architecture. Non-ACTN IETF data models required for L2/L3VPN service provisioning between MDSC and packet PNCs are also identified.

#### 4.1. Scenario 1: inventory, service and network topology discovery

In this scenario, the MSDC needs to discover through the underlying PNCs, the network topology, at both WDM and IP layers, in terms of nodes and links, including inter-AS domain links as well as cross-layer links but also in terms of tunnels (MPLS or SR paths in IP layer and OCh and optionally ODUK tunnels in optical layer).

In addition, the MDSC should discover the IP/MPLS transport services (L2VPN/L3VPN) deployed, both intra-domain and inter-domain wise.

The O-PNC and P-PNC could discover and report the inventory information of their equipment that is used by the different management layers. In the context of POI, the inventory information of IP and WDM equipment can complement the topology views and facilitate the IP-Optical multi-layer view.

The MDSC could also discover the whole inventory information of both IP and WDM equipment and correlate this information with the links reported in the network topology.

Each PNC provides to the MDSC an abstracted or full topology view of the WDM or the IP topology of the domain it controls. This topology can be abstracted in the sense that some detailed NE information is hidden at the MPI. All or some of the NEs and related physical links are exposed as abstract nodes and logical (virtual) links, depending on the level of abstraction the user requires. This information is key to understanding both the inter-AS domain links (seen by each controller as UNI interfaces but as I-NNI interfaces by the MDSC) and the cross-layer mapping between IP and WDM layer.

The MDSC should also maintain up-to-date inventory, service and network topology databases of both IP and WDM layers (and optionally OTN layer) through the use of IETF notifications through MPI with the PNCs when any inventory/topology/service change occurs.

It should be possible also to correlate information coming from IP and WDM layers (e.g., which port, lambda/OTSi, and direction, is used by a specific IP service on the WDM equipment).

In particular, for the cross-layer links, it is key for MDSC to automatically correlate the information from the PNC network databases about the physical ports from the routers (single link or bundle links for LAG) to client ports in the ROADM.

It should be possible at MDSC level to easily correlate WDM and IP layers alarms to speed-up troubleshooting

Alarms and event notifications are required between MDSC and PNCs so that any network changes are reported almost in real-time to the MDSC (e.g. NE or link failure, MPLS tunnel switched from primary to back-up path etc.). As specified in [RFC7923], MDSC must subscribe to specific objects from PNC YANG datastores for notifications.

#### 4.1.1. Inter-domain link discovery

In the reference network of Figure 1, there are two types of inter-domain links:

- o Links between two IP domains (ASes)
- o Links between an IP router and a ROADM

Both types of links are Ethernet physical links.

The inter-domain link information is reported to the MDSC by the two adjacent PNCs, controlling the two ends of the inter-domain link. The MDSC needs to understand how to merge these inter-domain Ethernet links together.

This document considers the following two options for discovering inter-domain links:

1. Static configuration
2. LLDP [IEEE 802.1AB] automatic discovery

Other options are possible but not described in this document.

The MDSC can understand how to merge these inter-domain links using the plug-id attribute defined in the TE Topology Model [RFC8795], as described in section 4.3 of [RFC8795].

A more detailed description of how the plug-id can be used to discover inter-domain links is also provided in section 5.1.4 of [TNBI].

Both types of inter-domain links are discovered using the plug-id attributes reported in the Ethernet Topologies exposed by the two adjacent PNCs. In addition, the MDSC can also discover an inter-domain IP link/adjacency between the two IP LTPs, reported in



the IP Topologies exposed by the two adjacent P-PNCs, supported by the two ETH LTPs of an Ethernet Link discovered between these two P-PNCs.

The static configuration requires an administrative burden to configure network-wide unique identifiers: it is therefore more viable for inter-AS links. For the links between the IP routers and the Optical NEs, the automatic discovery solution based on LLDP snooping is preferable when LLDP snooping is supported by the Optical NEs.

As outlined in [TNBI], the encoding of the plug-id namespace and the LLDP information within the plug-id value is implementation specific and needs to be consistent across all the PNCs.

#### 4.1.2. Multi-layer IP Link discovery

All the intra-domain IP links are discovered by P-PNC, using LLDP [IEEE 802.1AB] or any other mechanisms which are outside the scope of this document, and reported at the MPI within the L3 Topology.

In case of a multi-layer IP link, the P-PNC also reports the two inter-domain ETH LTPs that supports the two IP LTPs terminating the multi-layer IP link.

The MDSC can therefore discover which Ethernet access link supports the multi-layer IP link as described in section 4.1.1.

The Optical Transponders, or the OTN access cards, are reported by the O-PNC as Trail Termination Points (TTPs), defined in [TE-TOPO], within the Optical Topology. The association between the Ethernet access link and the Optical TTP is reported using the Inter Layer Lock (ILL) identifiers, defined in [TE-TOPO], within the Ethernet Topology and Optical Topology, exposed by the O-PNC.

The MDSC can discover through the MPI the Optical Tunnels being setup by each O-PNC and in particular which Optical Tunnel has been setup between the two TTPs associated with the two Ethernet access links supporting an inter-domain IP Link.

#### 4.1.3. Inventory discovery

There are no YANG data models in IETF that could be used to report at the MPI the whole inventory information discovered by a PNC.

[RFC8345] has foreseen some work for inventory as an augmentation of the network model, but no YANG data model has been developed so far.

There are also no YANG data models in IETF that could be used to correlate topology information, e.g., a link termination point (LTP), with inventory information, e.g., the physical port supporting an LTP, if any.

Inventory information through MPI and correlation with topology information is identified as a gap requiring further work and outside of the scope of this draft.

#### 4.1.4. SR-TE paths discovery

This version of the draft assumes that discovery of existing SR-TE paths, including their bandwidth, at the MPI is done using the generic TE tunnel YANG data model, defined in [TE-TUNNEL], with SR-TE specific augmentations, as also outlined in section 1 of [TE-TUNNEL].

To enable MDSC to discover the full end-to-end SR-TE path configuration, the SR-TE specific augmentation of the [TE-TUNNEL] should allow the P-PNC to report the SID list assigned to an SR-TE path within its domain.

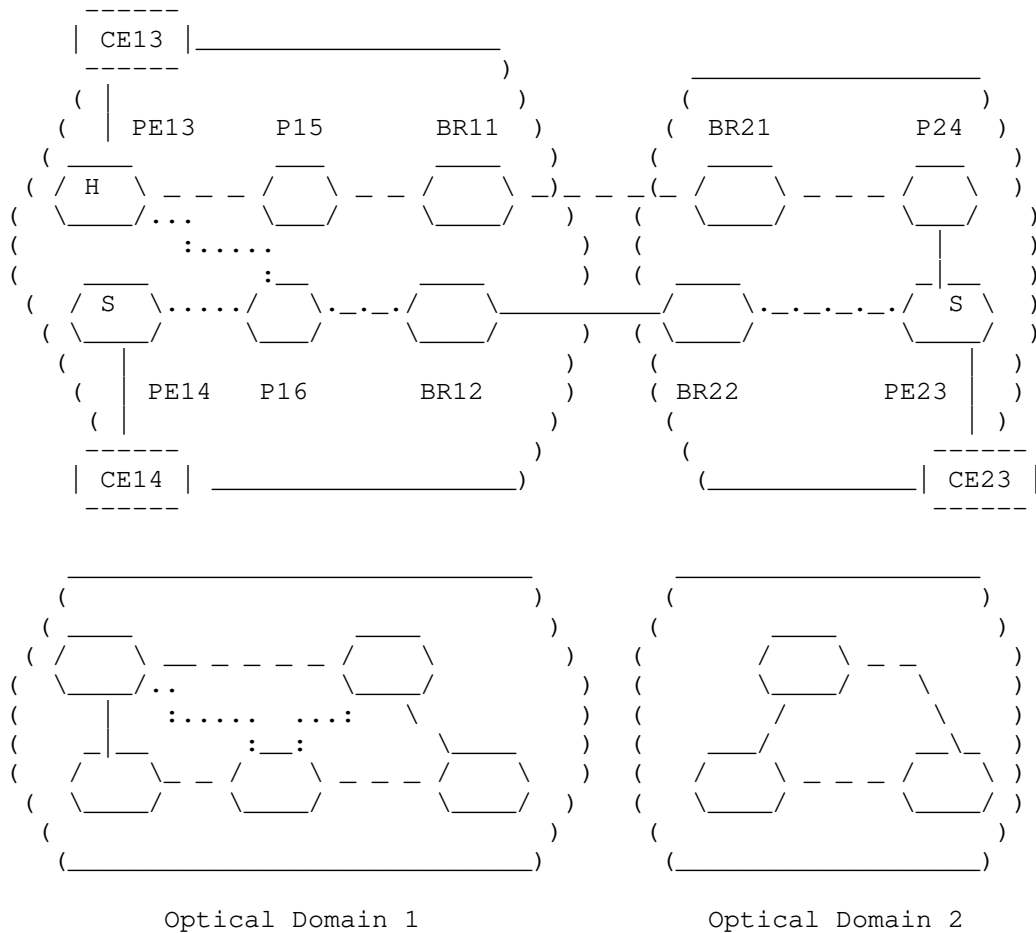
[Editors' note:] Need to check if SR-TE specific augmentation is required for SR-TE path discovery

For example, considering the L3VPN in Figure 4, the PE13-PE16-PE14 SR-TE path and the SR-TE path in the reverse direction (between PE14 and PE13) could be reported by the P-PNC1 to the MDSC as TE paths of the same TE tunnel instance. The bandwidth of these TE paths represents the bandwidth allocated by P-PNC1 to the two SR-TE paths, which can be symmetric or asymmetric in the two directions.

#### 4.2. Establishment of L2VPN/L3VPN with TE requirements

In this scenario the MDSC needs to setup a multi-domain L2VPN or a L3VPN with some SLA requirements.

Figure 4 provides an example of an hub&spoke L3VPN with three PEs where the hub PE (PE13) and one spoke PE (PE14) are within the same packet domain and the other spoke PE (PE23) is within a different packet domain.



H / S = Hub VRF / Spoke VRF  
 — = Inter-domain interconnections  
 ..... = SR policy Path 1  
 - - - = SR policy Path 2

Figure 4 Multi-domain L3VPN example

[Editors' note:] Update the SR policy paths to show the intra-domain PE13-P16-P14 and inter-domain PE13-BR11-BR12-P24-PE23 paths. No need to show the TI-LFA in this figure. Remove also the intra-domain TI-LFA.

There are many options to implement multi-domain L3VPN, including:

1. BGP-LU (seamless MPLS)
2. Inter-domain RSVP-TE
3. Inter-domain SR-TE

This version of the draft provides an analysis of the inter-domain SR-TE option. A future update of this draft will provide a high-level analysis of the BGP-LU option.

It is assumed that each packet domain in Figure 4 is implementing SR-TE and the stitching between two domains is done using end-to-end/multi-domain SR-TE. It is assumed that the bandwidth of each intra-domain SR-TE path is managed by its respective P-PNC and that binding SID is used for the end-to-end SR-TE path stitching. It is assumed that each packet domain in Figure 4 is using TI-LFA, with SRLG awareness, for local protection within each domain.

[Editor's note:] Analyze how TI-LFA can take into account multi-layer SRLG disjointness, providing that SRLG information is provided by the O-PNCs to the P-PNC through the MDSC.

It is assumed that the MDSC adopts the partial summarization model, described in section 2.2, having full visibility of the packet layer TE topology and an abstract view of the underlay optical layer TE topology.

The MDSC needs to translate the L3VPN SLA requirements to TE requirements (e.g., bandwidth, TE metric bounds, SRLG disjointness, nodes/links/domains inclusion/exclusion) and find the SR-TE paths between PE13 (hub PE) and, respectively, PE23 and PE14 (spoke PEs) that meet these TE requirements.

For each SR-TE path required to support the L3VPN, it is possible that:

1. A SR-TE path that meets the TE requirements already exist in the network.
2. An existing SR-TE path could be modified (e.g., through bandwidth increase) to meet the TE requirements:
  - a. The SR-TE path characteristics can be modified only in the packet layer.

- b. One or more new underlay Optical tunnels need to be setup to support the requested changes of the overlay SR-TE paths (multi-layer coordination is required).
- 3. A new SR-TE path needs to be setup:
  - a. The new SR-TE path reuses existing underlay optical tunnels;
  - b. One or more new underlay Optical tunnels need to be setup to support the setup of the new SR-TE path (multi-layer coordination is required).

For example, considering the L3VPN in Figure 4, the MDSC discovers that:

- o a PE13-P16-PE14 SR-TE path already exists but have not enough bandwidth to support the new L3VPN, as described in section 4.1.4;
- o the IP link(s) between P16 and PE14 has not enough bandwidth to support increasing the bandwidth of that SR-TE path, as described in section 4.1;
- o a new underlay optical tunnel could be setup to increase the bandwidth IP link(s) between P16 and PE14 to support increasing the bandwidth of that overlay SR-TE path, as described in section 4.2.1. The dimensioning of the underlay optical tunnel is decided by the MDSC based on the bandwidth requested by the SR-TE path and on its multi-layer optimization policy, which is an internal MDSC implementation issue.

The MDSC would therefore request:

- o the O-PNC1 to setup a new optical tunnel between the ROADMs connected to P16 and PE14, as described in section 4.2.2;
- o the P-PNC1 to update the configuration of the existing IP link, in case of LAG, or configure a new IP link, in case of ECMP, between P16 and PE14, as described in section 4.2.2;
- o the P-PNC1 to update the bandwidth of the selected SR-TE path between PE13 and PE14, as described in section 4.2.3.

For example, considering the L3VPN in Figure 4, the MDSC can also decide that a new multi-domain SR-TE path needs to be setup between PE13 and PE23.

As described in section 2.2, with partial summarization, the MDSC will use the TE topology information provided by the P-PNCs and the results of the path computation requests sent to the O-PNCs, as described in section 4.2.1, to compute the multi-layer/multi-domain path between PE13 and PE23.

For example, the multi-layer/multi-domain performed by the MDSC could require the setup of:

- o a new underlay optical tunnel between PE13 and BR11, supporting a new IP link, as described in section 4.2.2;
- o a new underlay optical tunnel between BR21 and P24 to increase the bandwidth of the IP link(s) between BR21 and P24, as described in section 4.2.2.

After that, the MDSC requests P-PNC2 to setup an SR-TE path between BR21 and PE23, with an explicit path (BR21, P24, PE23) as described in section 4.2.3. The P-PNC2, knowing the node and the adjacency SIDs assigned within its domain, can install the proper SR policy, or hierarchical policies, within BR21 and returns to the MDSC the assigned binding SID.

[Editor's Note] Further investigation is needed for the SR specific extensions to the TE tunnel model.

MDSC request P-PNC1 to setup an SR-TE path between PE13 and BR11, with an explicit path (PE13, BR11), specifying the inter-domain link toward BR21 and the binding SID to be used for the end-to-end SR-TE path stitching, as described in section 4.2.3. The P-PNC1, knowing also the node and the adjacency SIDs assigned within its domain and the EPE SID assigned by BR11 to the inter-domain link toward BR21, installs the proper policy, or policies, within PE13.

Once the SR-TE paths have been selected and, if needed, setup/modified, the MDSC can request to both P-PNCs to configure the L3VPN and its binding with the selected SR-TE paths using the [L3NM] and [TSM] YANG models.

[Editor's Note] Further investigation is needed to understand how the binding between a L3VPN and this new end-to-end SR-TE path can be configured.

#### 4.2.1. Optical Path Computation

As described in section 2.2, the optical path computation is usually performed by the Optical PNC.

When performing multi-layer/multi-domain path computation, the MDSC can delegate the Optical PNCs for single-domain optical path computation.

As discussed in [PATH-COMPUTE], there are two options to request an Optical PNC to perform optical path computation: either via a "compute-only" TE tunnel path, using the generic TE tunnel YANG data model defined in [TE-TUNNEL] or via the path computation RPC defined in [PATH-COMPUTE].

This draft assumes that the path computation RPC is used.

There are no YANG data models in IETF that could be used to augment the generic path computation RPC with technology-specific attributes.

Optical technology-specific augmentation for the path computation RPC is identified as a gap requiring further work outside of this draft's scope.

#### 4.2.2. Multi-layer IP Link Setup and Update

The MDSC requires the O-PNC to setup an Optical Tunnel (either a WSON Tunnel or a Flexi-grid Tunnel or an OTN Tunnel) within the Optical network between the two Optical Transponders (OTs), in case of DWDM network, or the two OTN access cards, in case of OTN networks, associated with the two access links.

The MDSC also requires the O-PNC to steer the Ethernet client traffic between the two access Ethernet Links over the Optical Tunnel.

After the Optical Tunnel has been setup and the client traffic steering configured, the two IP routers can exchange Ethernet packets between themselves, including LLDP messages.

If LLDP [IEEE 802.1AB] is used between the two routers, the P-PNC can automatically discover the IP Link being set up by the MDSC. The IP LTPs terminating this IP Link are supported by the ETH LTPs terminating the two access links.

Otherwise, the MDSC needs to require the P-PNC to configure an IP Link between the two routers: the MDSC also configures the two ETH LTPs which support the two IP LTPs terminating this IP Link.

[Editor's Note] Add text for IP link update and clarify that the IP link bandwidth increase can be done either by LAG or by ECMP. Both options are valid and widely deployed and more or less the same from POI perspective.

#### 4.2.3. SR-TE Path Setup and Update

This version of the draft assumes that SR-TE path setup and update at the MPI could be done using the generic TE tunnel YANG data model, defined in [TE TUNNEL], with SR TE specific augmentations, as also outlined in section 1 of [TE TUNNEL].

The MDSC can use the [TE-TUNNEL] model to request the P-PNC to setup TE paths specifying the explicit path to force the P-PNC to setup the actual path being computed by MDSC.

The [TE-TUNNEL] model supports requesting the setup of both end-to-end as well as segment TE paths (within one domain).

In the latter case, SR-TE specific augmentations of the [TE-TUNNEL] model should be defined to allow the MDSC to configure the binding SIDs to be used for the end to-end SR-TE path stitching and to allow the P-PNC to report the binding SID assigned to the segment TE paths.

The assigned binding SID should be persistent in case router or P-PNC rebooting.

The MDSC can also use the [TE-TUNNEL] model to request the P-PNC to increase the bandwidth allocated to an existing TE path, and, if needed, also on its reverse TE path. The [TE-TUNNEL] model supports both symmetric and asymmetric bandwidth configuration in the two directions.

SR-TE path setup and update (e.g., bandwidth increase) through MPI is identified as a gap requiring further work, which is outside of the scope of this draft.

#### 5. Security Considerations

Several security considerations have been identified and will be discussed in future versions of this document.



## 6. Operational Considerations

Telemetry data, such as collecting lower-layer networking health and consideration of network and service performance from POI domain controllers, may be required. These requirements and capabilities will be discussed in future versions of this document.

## 7. IANA Considerations

This document requires no IANA actions.

## 8. References

### 8.1. Normative References

- [RFC7950] Bjorklund, M. et al., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, August 2016.
- [RFC8040] Bierman, A. et al., "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8345] Clemm, A., Medved, J. et al., "A Yang Data Model for Network Topologies", RFC8345, March 2018.
- [RFC8346] Clemm, A. et al., "A YANG Data Model for Layer 3 Topologies", RFC8346, March 2018.
- [RFC8453] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC8453, August 2018.
- [RFC8525] Bierman, A. et al., "YANG Library", RFC 8525, March 2019.
- [RFC8795] Liu, X. et al., "YANG Data Model for Traffic Engineering (TE) Topologies", RFC8795, August 2020.
- [IEEE 802.1AB] IEEE 802.1AB-2016, "IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery", March 2016.
- [WSN-TOPO] Lee, Y. et al., " A YANG Data Model for WSON (Wavelength Switched Optical Networks)", draft-ietf-ccamp-wson-yang, work in progress.

- [Flexi-TOPO] Lopez de Vergara, J. E. et al., "YANG data model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang, work in progress.
- [OTN-TOPO] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang, work in progress.
- [CLIENT-TOPO] Zheng, H. et al., "A YANG Data Model for Client-layer Topology", draft-zheng-ccamp-client-topo-yang, work in progress.
- [L3-TE-TOPO] Liu, X. et al., "YANG Data Model for Layer 3 TE Topologies", draft-ietf-teas-yang-l3-te-topo, work in progress.
- [SR-TE-TOPO] Liu, X. et al., "YANG Data Model for SR and SR TE Topologies on MPLS Data Plane", draft-ietf-teas-yang-sr-te-topo, work in progress.
- [TE-TUNNEL] Saad, T. et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.
- [WSO-TUNNEL] Lee, Y. et al., "A Yang Data Model for WSON Tunnel", draft-ietf-ccamp-wson-tunnel-model, work in progress.
- [Flexi-MC] Lopez de Vergara, J. E. et al., "YANG data model for Flexi-Grid media-channels", draft-ietf-ccamp-flexigrid-media-channel-yang, work in progress.
- [OTN-TUNNEL] Zheng, H. et al., "OTN Tunnel YANG Model", draft-ietf-ccamp-otn-tunnel-model, work in progress.
- [PATH-COMPUTE] Busi, I., Belotti, S. et al, "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation, work in progress.
- [CLIENT-SIGNAL] Zheng, H. et al., "A YANG Data Model for Transport Network Client Signals", draft-ietf-ccamp-client-signal-yang, work in progress.
- [L2NM] S. Barguil, et al., "A Layer 2 VPN Network YANG Model", draft-ietf-opsawg-l2nm, work in progress.

- [L3NM] S. Barguil, et al., "A Layer 3 VPN Network YANG Model", draft-ietf-opsawg-l3sm-l3nm, work in progress.
- [TSM] Y. Lee, et al., "Traffic Engineering and Service Mapping Yang Model", draft-ietf-teas-te-service-mapping-yang, work in progress.

## 8.2. Informative References

- [RFC1930] J. Hawkinson, T. Bates, "Guideline for creation, selection, and registration of an Autonomous System (AS)", RFC 1930, March 1996.
- [RFC4364] E. Rosen and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC4761] K. Kompella, Ed., Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.
- [RFC6074] E. Rosen, B. Davie, V. Radoaca, and W. Luo, "Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)", RFC 6074, January 2011.
- [RFC6624] K. Kompella, B. Kothari, and R. Cherukuri, "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling", RFC 6624, May 2012.
- [RFC7209] A. Sajassi, R. Aggarwal, J. Uttaro, N. Bitar, W. Henderickx, and A. Isaac, "Requirements for Ethernet VPN (EVPN)", RFC 7209, May 2014.
- [RFC7432] A. Sajassi, Ed., et al., "BGP MPLS-Based Ethernet VPN", RFC 7432, February 2015.
- [RFC7436] H. Shah, E. Rosen, F. Le Faucheur, and G. Heron, "IP-Only LAN Service (IPLS)", RFC 7436, January 2015.
- [RFC8214] S. Boutros, A. Sajassi, S. Salam, J. Drake, and J. Rabadan, "Virtual Private Wire Service Support in Ethernet VPN", RFC 8214, August 2017.
- [RFC8299] Q. Wu, S. Litkowski, L. Tomotaki, and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, January 2018.

- [RFC8309] Q. Wu, W. Liu, and A. Farrel, "Service Model Explained", RFC 8309, January 2018.
- [RFC8466] G. Fiocco, ed., "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC8466, October 2018.
- [TNBI] Busi, I., Daniel, K. et al., "Transport Northbound Interface Applicability Statement", draft-ietf-ccamp-transport-nbi-app-statement, work in progress.
- [VN] Y. Lee, et al., "A Yang Data Model for ACTN VN Operation", draft-ietf-teas-actn-vn-yang, work in progress.
- [ACTN-PM] Y. Lee, et al., "YANG models for VN & TE Performance Monitoring Telemetry and Scaling Intent Autonomics", draft-lee-teas-actn-pm-telemetry-autonomics, work in progress.
- [BGP-L3VPN] D. Jain, et al. "Yang Data Model for BGP/MPLS L3 VPNs", draft-ietf-bess-l3vpn-yang, work in progress.

## Appendix A. Multi-layer and multi-domain resiliency

### A.1. Maintenance Window

Before planned maintenance operation on DWDM network takes place, IP traffic should be moved hitless to another link.

MDSC must reroute IP traffic before the events takes place. It should be possible to lock IP traffic to the protection route until the maintenance event is finished, unless a fault occurs on such path.

### A.2. Router port failure

The focus is on client-side protection scheme between IP router and reconfigurable ROADM. Scenario here is to define only one port in the routers and in the ROADM muxponder board at both ends as back-up ports to recover any other port failure on client-side of the ROADM (either on router port side or on muxponder side or on the link between them). When client-side port failure occurs, alarms are raised to MDSC by IP-PNC and O-PNC (port status down, LOS etc.). MDSC checks with OP-PNC(s) that there is no optical failure in the optical layer.

There can be two cases here:

- a) LAG was defined between the two end routers. MDSC, after checking that optical layer is fine between the two end ROADMs, triggers the ROADM configuration so that the router back-up port with its associated muxponder port can reuse the OCh that was already in use previously by the failed router port and adds the new link to the LAG on the failure side.

While the ROADM reconfiguration takes place, IP/MPLS traffic is using the reduced bandwidth of the IP link bundle, discarding lower priority traffic if required. Once back-up port has been reconfigured to reuse the existing OCh and new link has been added to the LAG then original Bandwidth is recovered between the end routers.

Note: in this LAG scenario let assume that BFD is running at LAG level so that there is nothing triggered at MPLS level when one of the link member of the LAG fails.

- b) If there is no LAG then the scenario is not clear since a router port failure would automatically trigger (through BFD failure) first a sub-50ms protection at MPLS level :FRR (MPLS RSVP-TE case) or TI-LFA (MPLS based SR-TE case) through a protection port. At the same time MDSC, after checking that optical network connection is still fine, would trigger the reconfiguration of the back-up port of the router and of the ROADM muxponder to re-use the same OCh as the one used originally for the failed router port. Once everything has been correctly configured, MDSC Global PCE could suggest to the operator to trigger a possible re-optimization of the back-up MPLS path to go back to the MPLS primary path through the back-up port of the router and the original OCh if overall cost, latency etc. is improved. However, in this scenario, there is a need for protection port PLUS back-up port in the router which does not lead to clear port savings.

#### Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Some of this analysis work was supported in part by the European Commission funded H2020-ICT-2016-2 METRO-HAUL project (G.A. 761727).

#### Contributors

Sergio Belotti  
Nokia

Email: sergio.belotti@nokia.com

Gabriele Galimberti  
Cisco

Email: ggalimbe@cisco.com

Zheng Yanlei  
China Unicom

Email: zhengyanlei@chinaunicom.cn

Anton Snitser  
Sedona

Email: antons@sedonasys.com

Washington Costa Pereira Correia  
TIM Brasil

Email: wcorreia@timbrasil.com.br

Michael Scharf  
Hochschule Esslingen - University of Applied Sciences

Email: michael.scharf@hs-esslingen.de

Young Lee  
Sung Kyun Kwan University

Email: younglee.tx@gmail.com

Jeff Tantsura  
Apstra

Email: jefftant.ietf@gmail.com

Paolo Volpato  
Huawei

Email: paolo.volpato@huawei.com

Brent Foster  
Cisco

Email: brfoster@cisco.com

Authors' Addresses

Fabio Peruzzini  
TIM

Email: fabio.peruzzini@telecomitalia.it

Jean-Francois Bouquier  
Vodafone

Email: jeff.bouquier@vodafone.com

Italo Busi  
Huawei

Email: Italo.busi@huawei.com

Daniel King  
Old Dog Consulting

Email: daniel@olddog.co.uk

Daniele Ceccarelli  
Ericsson

Email: daniele.ceccarelli@ericsson.com





TEAS Working Group  
Internet Draft  
Intended status: Informational

Fabio Peruzzini  
TIM  
Jean-Francois Bouquier  
Vodafone  
Italo Busi  
Huawei  
Daniel King  
Old Dog Consulting  
Daniele Ceccarelli  
Ericsson

Expires: September 2022

March 7, 2022

Applicability of Abstraction and Control of Traffic Engineered  
Networks (ACTN) to Packet Optical Integration (POI)

draft-ietf-teas-actn-poi-applicability-06

Abstract

This document considers the applicability of Abstraction and Control of TE Networks (ACTN) architecture to Packet Optical Integration (POI) in the context of IP/MPLS and optical internetworking. It identifies the YANG data models being defined by the IETF to support this deployment architecture and specific scenarios relevant for Service Providers.

Existing IETF protocols and data models are identified for each multi-layer (packet over optical) scenario with a specific focus on the MPI (Multi-Domain Service Coordinator to Provisioning Network Controllers Interface) in the ACTN architecture.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on April 9, 2021.

#### Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|   |    |
|---|----|
| 1. Introduction.....  | 3  |
| 1.1. Terminology.....   | 5  |
| 2. Reference network architecture.....                        | 7  |
| 2.1. Multi-domain Service Coordinator (MDSC) functions.....   | 9  |
| 2.1.1. Multi-domain L2/L3 VPN network services.....           | 11 |
| 2.1.2. Multi-domain and multi-layer path computation.....     | 14 |
| 2.2. IP/MPLS Domain Controller and NE Functions.....          | 17 |
| 2.3. Optical Domain Controller and NE Functions.....          | 19 |
| 3. Interface protocols and YANG data models for the MPIs..... | 19 |
| 3.1. RESTCONF protocol at the MPIs.....                       | 19 |
| 3.2. YANG data models at the MPIs.....                        | 20 |
| 3.2.1. Common YANG data models at the MPIs.....               | 20 |
| 3.2.2. YANG models at the Optical MPIs.....                   | 21 |
| 3.2.3. YANG data models at the Packet MPIs.....               | 21 |
| 3.3. PCEP.....  | 22 |
| 4. Inventory, service and network topology discovery.....     | 23 |
| 4.1. Optical topology discovery.....                          | 25 |
| 4.2. Optical path discovery.....                              | 26 |
| 4.3. Packet topology discovery.....                           | 27 |
| 4.4. SR-TE path discovery.....                                | 27 |
| 4.5. Inter-domain link discovery.....                         | 28 |

|   |    |
|---|----|
| 4.5.1. Cross-layer link discovery.....                              | 29 |
| 4.5.2. Inter-domain IP link discovery.....                          | 31 |
| 4.6. Multi-layer IP link discovery.....                             | 33 |
| 4.6.1. Single-layer intra-domain IP links.....                      | 36 |
| 4.7. LAG discovery.....   | 38 |
| 4.8. L2/L3 VPN network services discovery.....                      | 38 |
| 4.9. Inventory discovery.....                                       | 38 |
| 5. Establishment of L2/L3 VPN network services with TE requirements | 38 |
| 5.1. Optical Path Computation.....                                  | 40 |
| 5.2. Multi-layer IP link Setup.....                                 | 41 |
| 5.3. SR-TE Path Setup and Update.....                               | 42 |
| 6. Conclusions.....   | 43 |
| 7. Security Considerations.....                                     | 44 |
| 8. Operational Considerations.....                                  | 44 |
| 9. IANA Considerations.....   | 44 |
| 10. References.....   | 44 |
| 10.1. Normative References.....                                     | 44 |
| 10.2. Informative References.....                                   | 46 |
| Appendix A. OSS/Orchestration Layer.....                            | 49 |
| A.1. MDSC NBI.....  | 49 |
| Appendix B. Multi-layer and multi-domain resiliency.....            | 52 |
| B.1. Maintenance Window.....  | 52 |
| B.2. Router port failure.....                                       | 52 |
| Acknowledgments.....  | 53 |
| Contributors.....   | 53 |
| Authors' Addresses.....   | 55 |

## 1. Introduction

The complete automation of the management and control of Service Providers transport networks (IP/MPLS, optical, and microwave transport networks) is vital for meeting emerging demand for high-bandwidth use cases, including 5G and fiber connectivity services. The Abstraction and Control of TE Networks (ACTN) architecture and interfaces facilitate the automation and operation of complex optical and IP/MPLS networks through standard interfaces and data models. This allows a wide range of network services that can be requested by the upper layers fulfilling almost any kind of service level requirements from a network perspective (e.g. physical diversity, latency, bandwidth, topology, etc.)

Packet Optical Integration (POI) is an advanced use case of traffic engineering. In wide-area networks, a packet network based on the Internet Protocol (IP), and often Multiprotocol Label Switching (MPLS) or Segment Routing (SR), is typically realized on top of an

optical transport network that uses Dense Wavelength Division Multiplexing (DWDM) (and optionally an Optical Transport Network (OTN) layer).

In many existing network deployments, the packet and the optical networks are engineered and operated independently. As a result, there are technical differences between the technologies (e.g., routers compared to optical switches) and the corresponding network engineering and planning methods (e.g., inter-domain peering optimization in IP, versus dealing with physical impairments in DWDM, or very different time scales). In addition, customers needs can be different between a packet and an optical network, and it is not uncommon to use different vendors in both domains. The operation of these complex packet and optical networks is often siloed, as these technology domains require specific skills sets.

The packet/optical network deployment and operation separation are inefficient for many reasons. Both capital expenditure (CAPEX) and operational expenditure (OPEX) could be significantly reduced by integrating the packet and the optical networks. Multi-layer online topology insight can speed up troubleshooting (e.g., alarm correlation) and network operation (e.g., coordination of maintenance events), multi-layer offline topology inventory can improve service quality (e.g., detection of diversity constraint violations) and multi-layer traffic engineering can use the available network capacity more efficiently (e.g., coordination of restoration). In addition, provisioning workflows can be simplified or automated as needed across layers (e.g., to achieve bandwidth-on-demand or to perform activities during maintenance windows).

ACTN framework enables this complete multi-layer and multi-vendor integration of packet and optical networks through Multi-Domain Service Coordinator (MDSC) and packet and optical Provisioning Network Controllers (PNCs).

In this document, critical scenarios for POI are described from the packet service layer perspective and identified the required coordination between packet and optical layers to improve POI deployment and operation. Precise definitions of scenarios can help with achieving a common understanding across different disciplines. The focus of the scenarios are multi-domain packet networks operated as a client of optical networks.

This document analyses the case where the packet networks support multi-domain SR-TE paths and the optical networks could be either a DWDM network or an OTN network (without DWDM layer) or multi-layer

OTN/DWDM network. DWDM networks could be either fixed-grid or flexible-grid.

Multi-layer and multi-domain scenarios, based on reference network described in section 2, and very relevant for Service Providers, are described in section 4 and in section 5.

For each scenario, existing IETF protocols and data models, identified in section 3.1 and section 3.2, are analysed with particular focus on the MPI in the ACTN architecture.

For each multi-layer scenario, the document analyzes how to use the interfaces and data models of the ACTN architecture.

A summary of the gaps identified in this analysis is provided in section 6.

Understanding the level of standardization and the possible gaps will help assess the feasibility of integration between packet and optical DWDM domains (and optionally OTN layer) in an end-to-end multi-vendor service provisioning perspective.

### 1.1. Terminology

This document uses the ACTN terminology defined in [RFC8453]

In addition this document uses the following terminology.

Customer service:

the end-to-end service from CE to CE

Network service:

the PE to PE configuration including both the network service layer (VRFs, RT import/export policies configuration) and the network transport layer (e.g. RSVP-TE LSPs). This includes the configuration (on the PE side) of the interface towards the CE (e.g. VLAN, IP address, routing protocol etc.)

Port:

the physical entity that transmits and receives physical signals

**Interface:**

a physical or logical entity that transmits and receives traffic

**Link:**

an association between two interfaces that can exchange traffic directly

**Ethernet link:**

a link between two Ethernet interfaces

**IP link:**

a link between two IP interfaces

**Cross-layer link:**

an Ethernet link between an Ethernet interface on a router and an Ethernet interface on an optical NE

**Intra-domain single-layer Ethernet link:**

an Ethernet link between between two Ethernet interfaces on physically adjacent routers that belong to the same P-PNC domain

**Intra-domain single-layer IP link:**

an IP link supported by an intra-domain single-layer Ethernet link

**Inter-domain single-layer Ethernet link:**

an Ethernet link between between two Ethernet interfaces on physically adjacent routers which belong to different P-PNC domains

**Inter-domain single-layer IP link:**

an IP link supported by an inter-domain single-layer Ethernet link.

**Intra-domain multi-layer Ethernet link:**

an Ethernet link supported by two cross-layer links and an optical tunnel in between

Intra-domain multi-layer IP link:

an IP link supported an intra-domain multi-layer Ethernet link

## 2. Reference network architecture

This document analyses several deployment scenarios for Packet and Optical Integration (POI) in which ACTN hierarchy is deployed to control a multi-layer and multi-domain network, with two optical domains and two packet domains, as shown in Figure 1:

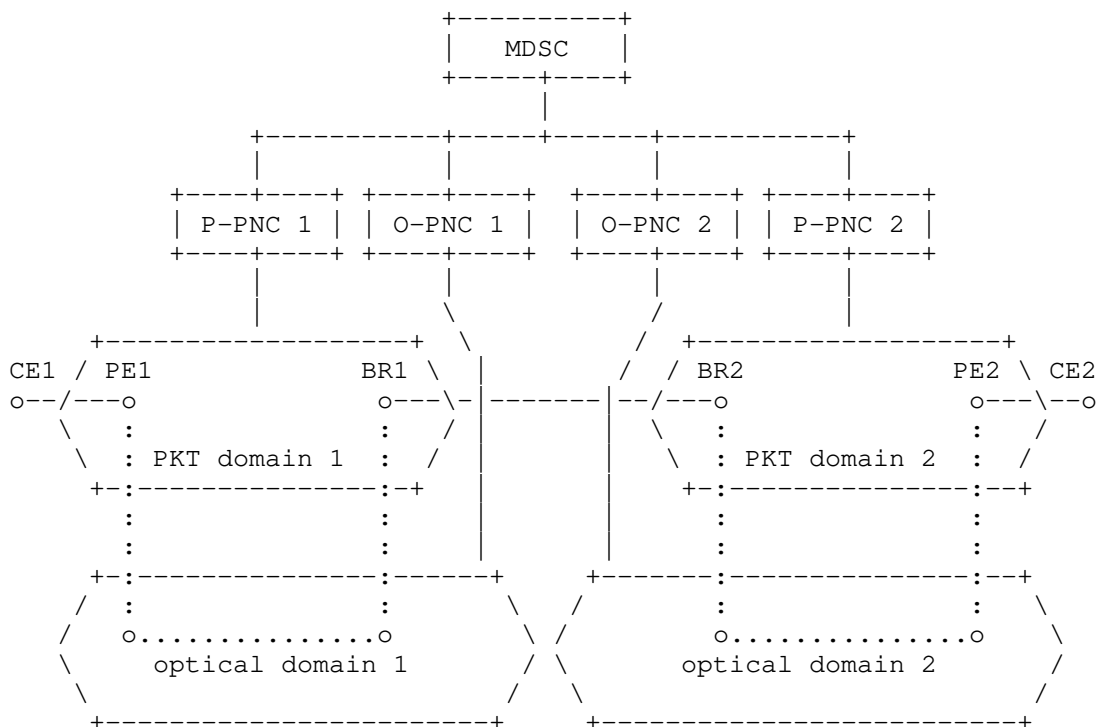


Figure 1 - Reference Network

The ACTN architecture, defined in [RFC8453], is used to control this multi-layer and multi-domain network where each Packet PNC (P-PNC) is responsible for controlling its packet domain and where each Optical PNC (O-PNC) in the above topology is responsible for controlling its



optical domain. The packet domains controlled by the P-PNCs can be Autonomous Systems (ASes), defined in [RFC1930], or IGP areas, within the same operator network.

The routers between the packet domains can be either AS Boundary Routers (ASBR) or Area Border Router (ABR): in this document, the generic term Border Router (BR) is used to represent either an ASBR or an ABR.

The MDSC is responsible for coordinating the whole multi-domain multi-layer (packet and optical) network. A specific standard interface (MPI) permits MDSC to interact with the different Provisioning Network Controller (O/P-PNCs).

The MPI interface presents an abstracted topology to MDSC hiding technology-specific aspects of the network and hiding topology details depending on the policy chosen regarding the level of abstraction supported. The level of abstraction can be obtained based on P-PNC and O-PNC configuration parameters (e.g., provide the potential connectivity between any PE and any BR in an SR-TE network).

In the reference network of Figure 1, it is assumed that:

- o The domain boundaries between the packet and optical domains are congruent. In other words, one optical domain supports connectivity between routers in one and only one packet domain;
- o There are no inter-domain physical links between optical domains. Inter-domain physical links exist only:
  - o between packet domains (i.e., between BRs belonging to different packet domains): these links are called inter-domain Ethernet or IP links within this document;
  - o between packet and optical domains (i.e., between routers and optical NEs): these links are called cross-layer links within this document;
  - o between customer sites and the packet network (i.e., between CE devices and PE routers): these links are called access links within this document.
- o All the physical interfaces at inter-domain links are Ethernet physical interfaces.

Although the new optical technologies (e.g., QSFP-DD ZR 400G) allows providing DWDM pluggable interfaces on the routers, the deployment of those pluggable optics is not yet widely adopted by the operators. The reason is that most operators are not yet ready to manage packet and optical networks in a single unified domain. The analysis of the unified use case is outside the scope of this draft.

This document analyses scenarios where all the multi-layer IP links, supported by the optical network, are intra-domain (intra-AS/intra-area), such as PE-BR, PE-P, BR-P, P-P IP links. Therefore the inter-domain IP links are always single-layer links supported by Ethernet physical links.

The analysis of scenarios with multi-layer inter-domain IP links is outside the scope of this document.

Therefore, if inter-domain links between the optical domains exist, they would be used to support multi-domain optical services, which are outside the scope of this document.

The optical network elements (NEs) within the optical domains can be ROADMs or OTN switches, with or without an integrated ROADM function.

## 2.1. Multi-domain Service Coordinator (MDSC) functions

The MDSC in Figure 1 is responsible for multi-domain and multi-layer coordination across multiple packet and optical domains, as well as to provide multi-layer/multi-domain L2/L3 VPN network services requested by an OSS/Orchestration layer.

From an implementation perspective, the functions associated with MDSC and described in [RFC8453] may be grouped in different ways.

1. Both the service- and network-related functions are collapsed into a single, monolithic implementation, dealing with the end customer service requests received from the CMI (Customer MDSC Interface) and adapting the relevant network models. An example is represented in Figure 2 of [RFC8453].
2. An implementation can choose to split the service-related and the network-related functions into different functional entities, as described in [RFC8309] and in section 4.2 of [RFC8453]. In this case, MDSC is decomposed into a top-level Service Orchestrator, interfacing the customer via the CMI, and into a Network Orchestrator interfacing at the southbound with the PNCs. The interface between the Service Orchestrator and the Network Orchestrator is not specified in [RFC8453].

3. Another implementation can choose to split the MDSC functions between an "higher-level MDSC" (MDSC-H) responsible for packet and optical multi-layer coordination, interfacing with one Optical "lower-level MDSC" (MDSC-L), providing multi-domain coordination between the O-PNCs and one Packet MDSC-L, providing multi-domain coordination between the P-PNCs (see for example Figure 9 of [RFC8453]).
4. Another implementation can also choose to combine the MDSC and the P-PNC functions together.

In the current service provider's network deployments, at the North Bound of the MDSC, instead of a CNC, typically there is an OSS/Orchestration layer. In this case, the MDSC would implement only the Network Orchestration functions, as in [RFC8309] and described in point 2 above. Therefore, the MDSC is dealing with the network services requests received from the OSS/Orchestration layer.

The functionality of the OSS/Orchestration layer and the interface toward the MDSC are usually operator-specific and outside the scope of this draft. Therefore, this document assumes that the OSS/Orchestrator requests the MDSC to set up L2/L3 VPN network services through mechanisms that are outside the scope of this document.

There are two prominent workflow cases when the MDSC multi-layer coordination is initiated:

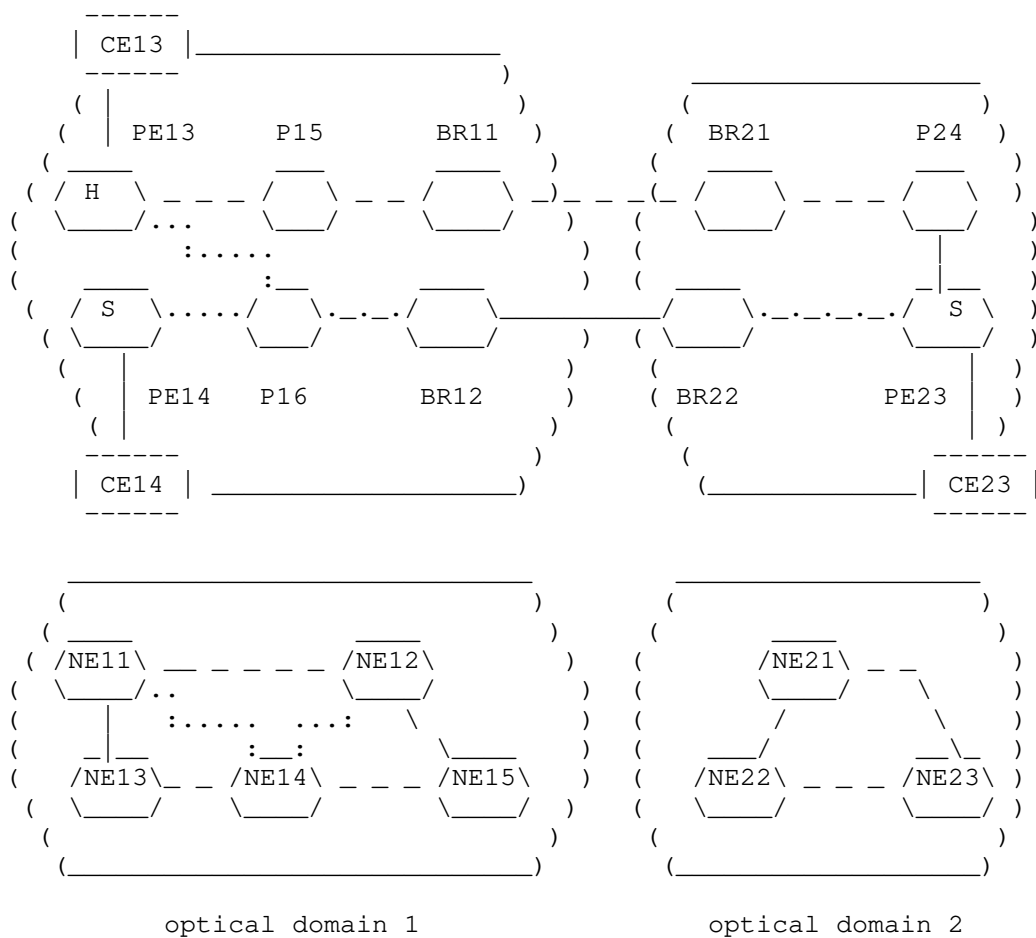
- o Initiated by a request from the OSS/Orchestration layer to setup L2/L3 VPN network services that requires multi-layer/multi-domain coordination;
- o Initiated by the MDSC itself to perform multi-layer/multi-domain optimizations and/or maintenance activities (e.g. rerouting LSPs with their associated services when putting a resource, like a fibre, in maintenance mode during a maintenance window). Unlike service fulfillment, these workflows are not related to a network service provisioning request being received from the OSS/Orchestration layer.

The latter workflow cases are outside the scope of this document.

This document analyses the use cases where multi-layer coordination is triggered by a network service request received from the OSS/Orchestration layer.

#### 2.1.1. Multi-domain L2/L3 VPN network services

Figure 2 provides an example of an hub & spoke multi-domain L2/L3 VPN with three PEs where the hub PE (PE13) and one spoke PE (PE14) are within the same packet domain and the other spoke PE (PE23) is within a different packet domain.



H / S = Hub VRF / Spoke VRF  
 — = Inter-domain interconnections  
 ..... = SR policy Path 1  
 \_ \_ \_ = SR policy Path 2

Figure 2 - Multi-domain L3VPN example

There are many options to implement multi-domain L2/L3 VPNs, including:

1. BGP-LU (seamless MPLS)
2. Inter-domain RSVP-TE

### 3. Inter-domain SR-TE

This document provides an analysis of the inter-domain SR-TE option. The analysis of other options is outside the scope of this draft.

It is also assumed that:

- o each packet domain in Figure 2 is implementing SR-TE and the stitching between two domains is done using end-to-end/multi-domain SR-TE;
- o the bandwidth of each intra-domain SR-TE path is managed by its respective P-PNC;
- o binding SID is used for the end-to-end SR-TE path stitching;
- o each packet domain in Figure 2 is using TI-LFA, with SRLG awareness, for local protection within each domain.

In this scenario, one of the key MDSC functions is to identify the multi-domain/multi-layer SR-TE paths to be used to carry the L2/L3 VPN traffic between PEs belonging to different packet domains and to relay this information to the P-PNCs, to ensure that the PEs' forwarding tables (e.g., VRF) are properly configured to steer the L2/L3 VPN traffic over the intended multi-domain/multi-layer SR-TE paths.

The selection of the SR-TE path should take into account the TE requirements and the binding requirements for the L2/L3 VPN network service.

In general the binding requirements for a network service (e.g L2/L3 VPN), can be summarized within three cases:

1. The customer is asking for VPN isolation dynamically creating and binding tunnels to the service such that they are not shared by others services (e.g. VPN).  
The level of isolation can be different:
  - a) Hard isolation with deterministic latency that means L2/L3 VPN requiring a set of dedicated TE Tunnels (neither sharing with other services nor competing for bandwidth with other tunnels) providing deterministic latency performances
  - b) Hard isolation but without deterministic characteristics

- c) Soft isolation that means the tunnels associated with L2/L3 VPN are dedicated to that but can compete for bandwidth with other tunnels.
- 2. The customer does not ask isolation, and could request a VPN service where associated tunnels can be shared across multiple VPNs.

For each SR-TE path required to support the L2/L3 VPN network service, it is possible that:

1. A SR-TE path that meets the TE and binding requirements already exist in the network.
2. An existing SR-TE path could be modified (e.g., through bandwidth increase) to meet the TE and binding requirements:
  - a. The SR-TE path characteristics can be modified only in the packet layer.
  - b. One or more new underlay optical tunnels need to be setup to support the requested changes of the overlay SR-TE paths (multi-layer coordination is required).
3. A new SR-TE path needs to be setup to meet the TE and binding requirements:
  - a. The new SR-TE path reuses existing underlay optical tunnels;
  - b. One or more new underlay optical tunnels need to be setup to support the setup of the new SR-TE path (multi-layer coordination is required).

#### 2.1.2. Multi-domain and multi-layer path computation

When a new SR-TE path needs to be setup, the MDSC is also responsible to coordinate the multi-layer/multi-domain path computation.

Depending on the knowledge that MDSC has of the topology and configuration of the underlying network domains, three approaches for performing multi-layer/multi-domain path computation are possible:

1. Full Summarization: In this approach, the MDSC has an abstracted TE topology view of all of its, packet and optical, underlying domains.

In this case, the MDSC does not have enough TE topology information to perform multi-layer/multi-domain path computation. Therefore the MDSC delegates the P-PNCs and O-PNCs to perform local path computation within their respective controlled domains and it uses the information returned by the P-PNCs and O-PNCs to compute the optimal multi-domain/multi-layer path.

This approach presents an issue to P-PNC, which does not have the capability of performing a single-domain/multi-layer path computation, since it can not retrieve the topology information from the O-PNCs nor delegate the O-PNC to perform optical path computation.

A possible solution could be to include a CNC function within the P-PNC to request the MDSC multi-domain optical path computation, as shown in Figure 10 of [RFC8453].

Another solution could be to rely on the MDSC recursive hierarchy, as defined in section 4.1 of [RFC8453], where, for each IP and optical domain pair, a "lower-level MDSC" (MDSC-L) provides the essential multi-layer correlation and the "higher-level MDSC" (MDSC-H) provides the multi-domain coordination.

In this case, the MDSC-H can get an abstract view of the underlying multi-layer domain topologies from its underlying MDSC-L. Each MDSC-L gets the full view of the IP domain topology from P-PNC and can get an abstracted view of the optical domain topology from its underlying O-PNC. In other words, topology abstraction is possible at the MPis between MDSC-L and O-PNC and between MDSC-L and MDSC-H.



2. Partial summarization: In this approach, the MDSC has full visibility of the TE topology of the packet network domains and an abstracted view of the TE topology of the optical network domains.

The MDSC then has only the capability of performing multi-domain/single-layer path computation for the packet layer (the path can be computed optimally for the two packet domains).

Therefore, the MDSC still needs to delegate the O-PNCs to perform local path computation within their respective domains and it uses the information received by the O-PNCs, together with its TE topology view of the multi-domain packet layer, to perform multi-layer/multi-domain path computation.

3. Full knowledge: In this approach, the MDSC has the complete and enough detailed view of the TE topology of all the network domains (both optical and packet).

In such case MDSC has all the information needed to perform multi-domain/multi-layer path computation, without relying on PNCs.

This approach may present, as a potential drawback, scalability issues and, as discussed in section 2.2. of [PATH-COMPUTE], performing path computation for optical networks in the MDSC is quite challenging because the optimal paths depend also on vendor-specific optical attributes (which may be different in the two domains if they are provided by different vendors).

This document analyses scenarios where the MDSC uses the partial summarization approach to coordinate multi-domain/multi-layer path computation.

Typically, the O-PNCs are responsible for the optical path computation of services across their respective single domains. Therefore, when setting up the network service, they must consider the connection requirements such as bandwidth, amplification, wavelength continuity, and non-linear impairments that may affect the network service path.

The methods and types of path requirements and impairments, such as those detailed in [OIA-TOPO], used by the O-PNC for optical path computation are not exposed at the MPI and therefore out of scope for this document.

## 2.2. IP/MPLS Domain Controller and NE Functions

As highlighted in section 2.1.1, SR-TE is used in the packet domain. Each domain, corresponding to either an IGP area or an Autonomous System (AS) within the same operator network, is controlled by a packet domain controller (P-PNC).

P-PNCs are responsible to setup the SR-TE paths between any two PEs or BRs in their respective controlled domains, as requested by MDSC, and to provide topology information to the MDSC.

With reference to Figure 2, a bidirectional SR-TE path from PE13 in domain 1 to PE23 in domain 2 requires the MDSC to coordinate the actions of:

- o P-PNC1 to push a SID list to PE13 including the Binding SID associated to the SR-TE path in Domain 2 with PE23 as the target destination (forward direction);
- o P-PNC2 to push a SID list to PE23 with including the Binding SID associated to the SR-TE path in Domain 1 with PE13 as the target destination (reverse direction).

With reference to Figure 3, P-PNCs are then responsible:

1. To expose to MDSC their respective detailed TE topology
2. To perform single-layer single-domain local SR-TE path computation, when requested by MDSC between two PEs (for single-domain end-to-end SR-TE path) or between PEs and BRs for an inter-domain SR-TE path selected by MDSC;
3. To configure the ingress PE or BR router in their respective domain with the SID list associated with an SR-TE path;
4. To configure finally the VRF and PE-CE interfaces (Service access points) of the intra-domain and inter-domain network services requested by the MDSC.

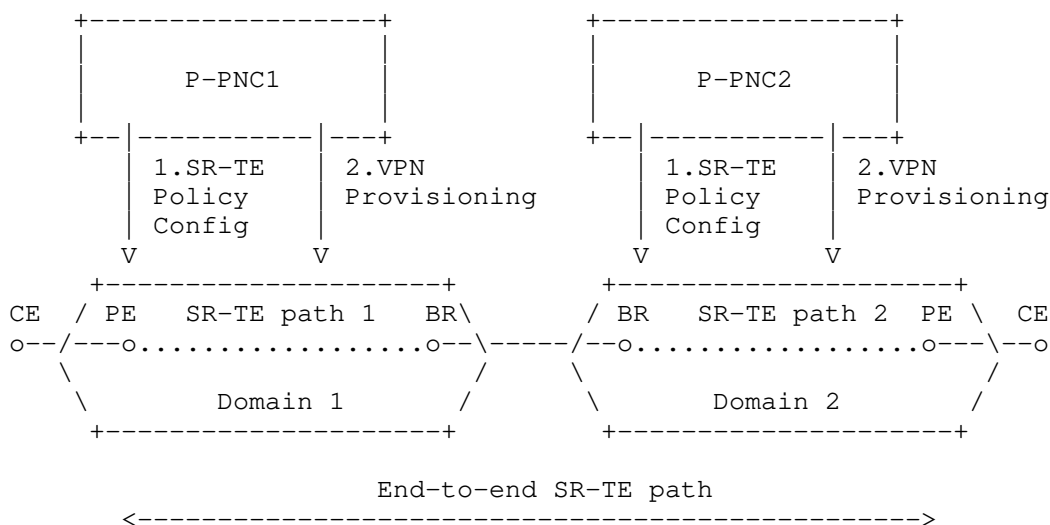


Figure 3 Domain Controller &amp; NE Functions

When requesting the setup of a new SR-TE path, the MDSC provides the P-PNCs with the explicit path to be created or modified. In other words, the MDSC can communicate to the P-PNCs the full list of nodes involved in the path (strict mode). In this case, the P-PNC is just responsible to push to headend PE or BR the list of SIDs to create that explicit SR-TE path.

For scalability purposes, in large packet domains, where multiple engineered paths are available between any two nodes, the MDSC can request a loose path, together with per-domain TE constraints, to allow the P-PNC selecting the intra-domain SR-TE path meeting these constraints.

In such a case it is mandatory that P-PNC signals back to the MDSC which path it has chosen so that the MDSC keeps track of the relevant resources utilization.

An example of that comes from Figure 2. The SR-TE path requested by the MDSC touches PE13 - P16 - BR12 - BR21 - PE23. P-PNC2 knows of two possible paths with the same topology metric, e.g. BR21 - P24 - PE23 and BR21 - BR22 - PE23, but with different load. It may prefer then to steer the traffic on the latter because it is less loaded.

This exception is mentioned here for the sake of completeness but since the network considered in this document does not fall in this scenario, in the rest of the paper the assumption is that the MDSC always provides the explicit list of SID(s) to the P-PNCs to setup or modify the SR-TE path.

### 2.3. Optical Domain Controller and NE Functions

The optical network provides the underlay connectivity services to IP/MPLS networks. The packet and optical multi-layer coordination is done by the MDSC, as shown in Figure 1.

The O-PNC is responsible to:

- o provide to the MDSC an abstract TE topology view of its underlying optical network resources;
- o perform single-domain local path computation, when requested by the MDSC;
- o perform optical tunnel setup, when requested by the MDSC.

The mechanisms used by O-PNC to perform intra-domain topology discovery and path setup are usually vendor-specific and outside the scope of this document.

Depending on the type of optical network, TE topology abstraction, path computation and path setup can be single-layer (either OTN or WDM) or multi-layer OTN/WDM. In the latter case, the multi-layer coordination between the OTN and WDM layers is performed by the O-PNC.

## 3. Interface protocols and YANG data models for the MPIs

This section describes general assumptions applicable at all the MPI interfaces, between each PNC (Optical or Packet) and the MDSC, to support the scenarios discussed in this document.

### 3.1. RESTCONF protocol at the MPIs

The RESTCONF protocol, as defined in [RFC8040], using the JSON representation defined in [RFC7951], is assumed to be used at these interfaces. In addition, extensions to RESTCONF, as defined in [RFC8527], to be compliant with Network Management Datastore Architecture (NMDA) defined in [RFC8342], are assumed to be used as well at these MPI interfaces and also at MDSC NBI interfaces.

### 3.2. YANG data models at the MPIs

The data models used on these interfaces are assumed to use the YANG 1.1 Data Modeling Language, as defined in [RFC7950].

#### 3.2.1. Common YANG data models at the MPIs

As required in [RFC8040], the "ietf-yang-library" YANG module defined in [RFC8525] is used to allow the MDSC to discover the set of YANG modules supported by each PNC at its MPI.

Both Optical and Packet PNCs use the following common topology YANG data models at the MPI:

- o The Base Network Model, defined in the "ietf-network" YANG module of [RFC8345];
- o The Base Network Topology Model, defined in the "ietf-network-topology" YANG module of [RFC8345], which augments the Base Network Model;
- o The TE Topology Model, defined in the "ietf-te-topology" YANG module of [RFC8795], which augments the Base Network Topology Model.

Both Optical and Packet PNCs use the common TE Tunnel Model, defined in the "ietf-te" YANG module of [TE-TUNNEL], at the MPI.

All the common YANG data models are generic and augmented by technology-specific YANG modules, as described in the following sections.

Both Optical and Packet PNCs also use the Ethernet Topology Model, defined in the "ietf-eth-te-topology" YANG module of [CLIENT-TOPO], which augments the TE Topology Model with Ethernet technology-specific information.

Both Optical and Packet PNCs use the following common notifications YANG data models at the MPI:

- o Dynamic Subscription to YANG Events and Datastores over RESTCONF as defined in [RFC8650];
- o Subscription to YANG Notifications for Datastores updates as defined in [RFC8641].

PNCs and MDSCs are compliant with subscription requirements as stated in [RFC7923].

### 3.2.2. YANG models at the Optical MPIS

The Optical PNC uses at least one of the following technology-specific topology YANG data models, which augment the generic TE Topology Model:

- o The WSON Topology Model, defined in the "ietf-wson-topology" YANG module of [RFC9094];
- o the Flexi-grid Topology Model, defined in the "ietf-flexi-grid-topology" YANG module of [Flexi-TOPO];
- o the OTN Topology Model, as defined in the "ietf-otn-topology" YANG module of [OTN-TOPO].

The optical PNC uses at least one of the following technology-specific tunnel YANG data models, which augments the generic TE Tunnel Model:

- o The WSON Tunnel Model, defined in the "ietf-wson-tunnel" YANG modules of [WSON-TUNNEL];
- o the Flexi-grid Tunnel Model, defined in the "ietf-flexi-grid-tunnel" YANG module of [Flexi-TUNNEL];
- o the OTN Tunnel Model, defined in the "ietf-otn-tunnel" YANG module of [OTN-TUNNEL].

The optical PNC can optionally use the generic Path Computation YANG RPC, defined in the "ietf-te-path-computation" YANG module of [PATH-COMPUTE].

Note that technology-specific augmentations of the generic path computation RPC for WSON, Flexi-grid and OTN path computation RPCs have been identified as a gap.

The optical PNC uses the Ethernet Client Signal Model, defined in the "ietf-eth-tran-service" YANG module of [CLIENT-SIGNAL].

### 3.2.3. YANG data models at the Packet MPIS

The Packet PNC also uses at least the following technology-specific topology YANG data models:

- o The L3 Topology Model, defined in the "ietf-l3-unicast-topology" YANG module of [RFC8346], which augments the Base Network Topology Model;
- o the L3 specific data model including extended TE attributes (e.g. performance derived metrics like latency), defined in "ietf-l3-te-topology" and in "ietf-te-topology-packet" YANG modules of [L3-TE-TOPO];
- o the SR Topology Model, defined in the "ietf-sr-mpls-topology" YANG module of [SR-TE-TOPO].

Need to check the need/applicability of the "ietf-l3-te-topology" in this scenario since it is not described in [SR-TE-TOPO].

The packet PNC uses at least the following YANG data models:

- o L3VPN Network Model (L3NM), defined in the "ietf-l3vpn-ntw" YANG module of [RFC9182];
- o L3NM TE Service Mapping, defined in the "ietf-l3nm-te-service-mapping" YANG module of [TSM];
- o L2VPN Network Model (L2NM), defined in the "ietf-l2vpn-ntw" YANG module of [L2NM];
- o L2NM TE Service Mapping, defined in the "ietf-l2nm-te-service-mapping" YANG module of [TSM].

### 3.3. PCEP

[RFC8637] examines the applicability of a Path Computation Element (PCE) [RFC5440] and PCE Communication Protocol (PCEP) to the ACTN framework. It further describes how the PCE architecture applies to ACTN and lists the PCEP extensions that are needed to use PCEP as an ACTN interface. The stateful PCE [RFC8231], PCE-Initiation [RFC8281], stateful Hierarchical PCE (H-PCE) [RFC8751], and PCE as a central controller (PCECC) [RFC8283] are some of the key extensions that enable the use of PCE/PCEP for ACTN.

Since the PCEP supports path computation in the packet and optical networks, PCEP is well suited for inter-layer path computation. [RFC5623] describes a framework for applying the PCE-based architecture to interlayer (G)MPLS traffic engineering. Furthermore, the section 6.1 of [RFC8751] states the H-PCE applicability for inter-layer or POI.

[RFC8637] lists various PCEP extensions that apply to ACTN. It also lists the PCEP extension for optical network and POI.

Note that the PCEP can be used in conjunction with the YANG data models described in the rest of this document. Depending on whether ACTN is deployed in a greenfield or brownfield, two options are possible:

1. The MDSC uses a single RESTCONF/YANG interface towards each PNC to discover all the TE information and request TE tunnels. It may either perform full multi-layer path computation or delegate path computation to the underneath PNCs.

This approach is desirable for operators from an multi-vendor integration perspective as it is simple, and we need only one type of interface (RESTCONF) and use the relevant YANG data models depending on the operator use case considered. Benefits of having only one protocol for the MPI between MDSC and PNC have been already highlighted in [PATH-COMPUTE].

4. The MDSC uses the RESTCONF/YANG interface towards each PNC to discover all the TE information and requests the creation of TE tunnels. However, it uses PCEP for hierarchical path computation.

As mentioned in Option 1, from an operator perspective, this option can add integration complexity to have two protocols instead of one, unless the RESTCONF/YANG interface is added to an existing PCEP deployment (brownfield scenario).

Section 4 and section 5 of this draft analyse the case where a single RESTCONF/YANG interface is deployed at the MPI (i.e., option 1 above).

#### 4. Inventory, service and network topology discovery

In this scenario, the MDSC needs to discover through the underlying PNCs:

- o the network topology, at both optical and IP layers, in terms of nodes and links, including the access links, inter-domain IP links as well as cross-layer links;
- o the optical tunnels supporting multi-layer intra-domain IP links;
- o both intra-domain and inter-domain L2/L3 VPN network services deployed within the network;



- o the SR-TE paths supporting those L2/L3 VPN network services;
- o the hardware inventory information of IP and optical equipment.

The O-PNC and P-PNC could discover and report the hardware network inventory information of their equipment that is used by the different management layers. In the context of POI, the inventory information of IP and optical equipment can complement the topology views and facilitate the packet/optical multi-layer view, e.g., by providing a mapping between the lowest level LTPs in the topology view and corresponding physical port in the network inventory view.

The MDSC could also discover the entire network inventory information of both IP and optical equipment and correlate this information with the links reported in the network topology.

Reporting the entire inventory and detailed topology information of packet and optical networks to the MDSC may present, as a potential drawback, scalability issues. The analysis of the scalability of this approach and mechanisms to address potential issues is outside the scope of this document.

Each PNC provides to the MDSC the topology view of the domain it controls, as described in section 4.1 and 4.3. The MDSC uses this information to discover the complete topology view of the multi-layer multi-domain network it controls.

The MDSC should also maintain up-to-date inventory, service and network topology databases of both IP and optical layers through the use of IETF notifications through MPI with the PNCs when any network inventory/topology/service change occurs.

It should be possible also to correlate information coming from IP and optical layers (e.g., which port, lambda/OTSi, and direction, is used by a specific IP service on the WDM equipment).

In particular, for the cross-layer links, it is key for MDSC to automatically correlate the information from the PNC network databases about the physical ports from the routers (single link or bundle links for LAG) to client ports in the ROADM.

The analysis of multi-layer fault management is outside the scope of this document. However, the discovered information should be sufficient for the MDSC to easily correlate optical and IP layers alarms to speed-up troubleshooting.

Alarms and event notifications are required between MDSC and PNCs so that any network changes are reported almost in real-time to the MDSC (e.g., NE or link failure). As specified in [RFC7923], MDSC must subscribe to specific objects from PNC YANG datastores for notifications.

#### 4.1. Optical topology discovery

The WSON Topology Model or, alternatively, the Flexi-grid Topology model is used to report the DWDM network topology (e.g., ROADM nodes and links), depending on whether the DWDM optical network is based on fixed grid or flexible-grid.

The OTN Topology Model is used to report the OTN network topology (e.g., OTN switching nodes and links), when the OTN switching layer is deployed within the optical domain.

In order to allow the MDSC to discover the complete multi-layer and multi-domain network topology and to correlate it with the hardware inventory information, the O-PNCs report an abstract optical network topology where:

- o one TE node is reported for each optical NE deployed within the optical network domain; and
- o one TE link is reported for each OMS link and, optionally, for each OTN link.

The Ethernet Topology Model is used to report the Ethernet client LTPs that terminate the cross-layer links: one Ethernet client LTP is reported for each Ethernet client interface on the optical NEs.

Since the MDSC delegates optical path computation to its underlay O-PNCs, the following information can be abstracted and not reported at the MPI:

- o the optical parameters required for optical path computation, such as those detailed in [OIA-TOPO];
- o the underlay OTS links and ILAs of OMS links;
- o the physical connectivity between the optical transponders and the ROADMs.

The optical transponders and, optionally, the OTN access cards, are abstracted at MPI by the O-PNC as Trail Termination Points (TTPs),

defined in [RFC8795], within the optical network topology. This abstraction is valid independently of the fact that optical transponders are physically integrated within the same WDM node or are physically located on a device external to the WDM node since in both cases the optical transponders and the WDM node are under the control of the same O-PNC.

The association between the Ethernet LTPs terminating the Ethernet cross-layer links and the optical TTPs is reported using the Inter Layer Lock (ILL) identifiers, defined in [RFC8795].

All the optical links are intra-domain and they are discovered by O-PNCs, using mechanisms which are outside the scope of this document, and reported at the MPIs within the optical network topology.

In case of a multi-layer DWDM/OTN network domain, multi-layer intra-domain OTN links are supported by underlay DWDM tunnels, which can be either WSON tunnels or, alternatively, Flexi-grid tunnels, depending on whether the DWDM optical network is based on fixed grid or flexible-grid. This relationship is reported by the mechanisms described in section 4.2.

#### 4.2. Optical path discovery

The WSON Tunnel Model or, alternatively, the Flexi-grid Tunnel model, depending on whether the DWDM optical network is based on fixed grid or flexible-grid, is used to report all the DWDM tunnels established within the optical network.

When the OTN switching layer is deployed within the optical domain, the OTN Tunnel Model is used to report all the OTN tunnels established within the optical network.

The Ethernet client signal Model is used to report all the Ethernet connectivity provided by the underlay optical tunnels between Ethernet client LTPs. The underlay optical tunnels can be either DWDM tunnels or, when the optional OTN switching layer is deployed, OTN tunnels.

The DWDM tunnels can be used as underlay tunnels to support either Ethernet client connectivity or multi-layer intra-domain OTN links. In the latter case, the hierarchical-link container, defined in [TE-TUNNEL], is used to reference which multi-layer intra-domain OTN links are supported by the underlay DWDM tunnels.

The O-PNCs report in their operational datastores all the Ethernet client connectivities and all the optical tunnels deployed within their optical domain regardless of the mechanisms being used to set them up, such as the mechanisms described in section 5.2, as well as other mechanism (e.g., static configuration), which are outside the scope of this document.

#### 4.3. Packet topology discovery

The L3 Topology Model, SR Topology Model, TE Topology Model and the TE Packet Topology Model are used together to report the SR-TE network topology, as described in figure 2 of [SR-TE-TOPO].

In order to allow the MDSC to discover the complete multi-layer and multi-domain network topology and to correlate it with the hardware inventory information as well as to perform multi-domain SR-TE path computation, the P-PNCs report the full SR-TE network, including all the information that is required by the MDSC to perform SR-TE path computation. In particular, one TE node is reported for each router and one TE link is reported for each intra-domain IP link. The SR-TE topology also reports the IP LTPs terminating the inter-domain IP links.

All the intra-domain IP links are discovered by the P-PNCs, using mechanisms, such as LLDP [IEEE 802.1AB], which are outside the scope of this document, and reported at the MPIs within the SR-TE network topology.

The Ethernet Topology Model is used to report the intra-domain Ethernet links supporting the intra-domain IP links as well as the Ethernet LTPs that might terminate cross-layer links, inter-domain Ethernet links or access links, as described in detail in section 4.5 and in section 4.6.

#### 4.4. SR-TE path discovery

This version of the draft assumes that discovery of existing SR-TE paths, including their bandwidth, at the MPI is done using the generic TE tunnel YANG data model, defined in [TE-TUNNEL], with SR-TE specific augmentations, as outlined in section 1 of [TE-TUNNEL].

Note that technology-specific augmentations of the generic path TE tunnel model for SR-TE path setup and discovery have been identified as a gap.

To enable MDSC to discover the full end-to-end SR-TE path configuration, the SR-TE specific augmentation of the [TE-TUNNEL] should allow the P-PNC to report the SID list assigned to an SR-TE path within its domain.

For example, considering the L3VPN in Figure 2, the PE13-PE16-PE14 SR-TE path and the SR-TE path in the reverse direction (between PE14 and PE13) could be reported by the P-PNC1 to the MDSC as TE paths of the same TE tunnel instance. The bandwidth of these TE paths represents the bandwidth allocated by P-PNC1 to the two SR-TE paths, which can be symmetric or asymmetric in the two directions.

The P-PNCs use the TE tunnel model to report, at the MPI, all the SR-TE paths established within their packet domain regardless of the mechanism being used to set them up. In other words, the TE tunnel data model reports within the operational datastore both the SR-TE paths being setup by the MDSC at the MPI, using the mechanisms described in section 5.3, as well as the SR-TE paths being setup by other means, such as static configuration, which are outside the scope of this document.

#### 4.5. Inter-domain link discovery

In the reference network of Figure 1, there are three types of inter-domain links:

- o Inter-domain Ethernet links supporting inter-domain IP links between two adjacent IP domains;
- o Cross-layer links between an IP domain and an adjacent optical domain;
- o Access links between a CE device and a PE router.

All the three types of links are Ethernet links.

It is worth noting that the P-PNC may not be aware whether an Ethernet interface terminates a cross-layer link, an inter-domain Ethernet link or an access link.

It is not yet clarified which model can be used to report the access links between CEs and PEs (e.g., by using the Ethernet Topology Model defined in [CLIENT-TOPO] or by using the SAP Model defined in [SAP]). This has been identified as a gap.

The inter-domain Ethernet links and cross-layer links are discovered by the MDSC using the plug-id attribute, as described in section 4.3 of [RFC8795].

More detailed description of how the plug-id can be used to discover inter-domain links is also provided in section 5.1.4 of [TNBI].

This document considers the following two options for discovering inter-domain links:

1. Static configuration
2. LLDP [IEEE 802.1AB] automatic discovery

Other options are possible but not described in this document.

As outlined in [TNBI], the encoding of the plug-id namespace and the specific LLDP information reported within the plug-id value, such as the Chassis ID and Port ID mandatory TLVs, is implementation specific and needs to be consistent across all the PNCs within the network.

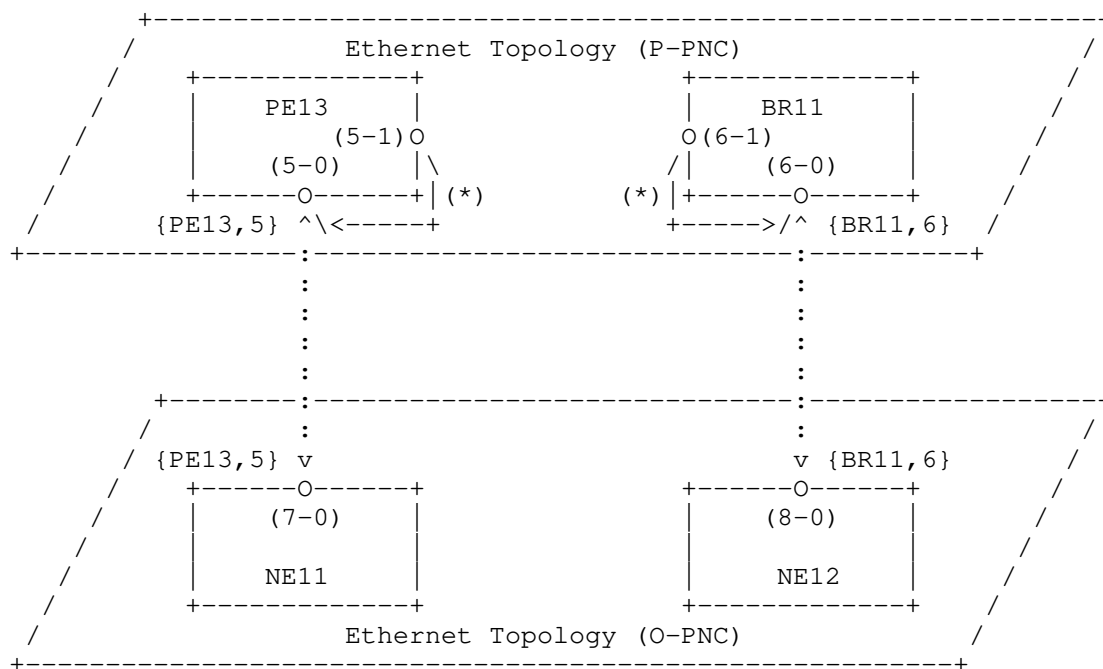
The static configuration requires an administrative burden to configure network-wide unique identifiers: it is therefore more viable for inter-domain Ethernet links. For the cross-layer links, the automatic discovery solution based on LLDP snooping is preferable when possible.

The routers exchange standard LLDP packets as defined in [IEEE 802.1AB] and the optical NEs snoop the LLDP packets received from the local Ethernet interface and report to the O-PNCs the extracted information, such as the Chassis ID, the Port ID, System Name TLVs.

Note that the optical NEs do not actively participate in the LLDP packet exchange and does not send any LLDP packets.

#### 4.5.1. Cross-layer link discovery

The MDSC can discover a cross-layer link by matching the plug-id values of the two Ethernet LTPs reported by two adjacent O-PNC and P-PNC: in case LLDP snooping is used, the P-PNC reports the LLDP information sent by the corresponding Ethernet interface on the router while the O-PNC reports the LLDP information received by the corresponding Ethernet interface on the optical NE, e.g., between LTP 5-0 on PE13 and LTP 7-0 on NE11, as shown in Figure 4.



## Notes:

=====

(\*) Supporting LTP

## Legenda:

=====

O LTP

----&gt; Supporting LTP

&lt;...&gt; Link discovered by the MDSC

{ } LTP Plug-id reported by the PNC

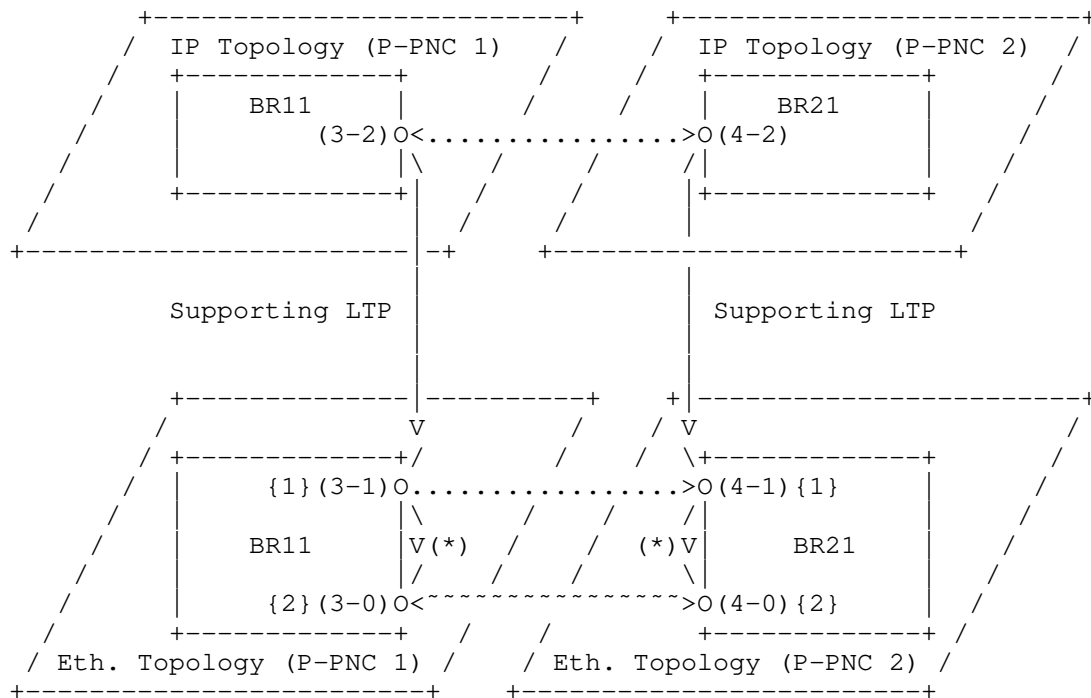
Figure 4 - Cross-layer link discovery

It is worth noting that the discovery of cross-layer links is based only on the LLDP information sent by the Ethernet interfaces of the routers and received by the Ethernet interfaces of the optical NEs, Therefore the MDSC can discover these links also before overlay multi-layer IP links are setup.

#### 4.5.2. Inter-domain IP link discovery

The MDSC can discover an inter-domain Ethernet link which supports an inter-domain IP link, by matching the plug-id values of the two Ethernet LTPs reported by the two adjacent P-PNCs: the two P-PNCs report the LLDP information being sent and being received from the corresponding Ethernet interfaces, e.g., between the Ethernet LTP 3-1 on BR11 and the Ethernet LTP 4-1 on BR21 shown in Figure 5.





## Notes:

=====

(\*) Supporting LTP  
 {1} {BR11,3,BR21,4}  
 {2} {BR11,3}

## Legenda:

=====

O LTP  
 ----> Supporting LTP  
 <...> Link discovered by the MDSC  
 <~~~> Link inferred by the MDSC  
 { } LTP Plug-id reported by the PNC

Figure 5 - Inter-domain Ethernet and IP link discovery

Different information is required to be encoded within the plug-id attribute of the Etherent LTPs to discover cross-layer links and inter-domain Ethernet links.

If the P-PNC does not know a priori whether an Ethernet interface on a router terminates a cross-layer link or an inter-domain Ethernet link, it has to report at the MPI two Ethernet LTPs representing the same Ethernet interface, e.g., both the Ethernet LTP 3-0 and the Ethernet LTP 3-1, supported by LTP 3-0, shown in Figure 5:

- o The physical Ethernet LTP is used to represent the physical adjacency between the router Ethernet interface and either the adjacent router Ethernet interface (in case of a single-layer Ethernet link) or the optical NE Ethernet interface (in case of a multi-layer Ethernet link). Therefore, this LTP reports, within the plug-id attribute, the LLDP information sent by the corresponding router Ethernet interface;
- o The logical Ethernet LTP, supported by a physical Ethernet LTP, is used to discover the logical adjacency between router Ethernet interfaces, which can be either single-layer or multi-layer. Therefore, this LTP reports, within the plug-id attribute, the LLDP information sent and received by the corresponding router Ethernet interface.

It is worth noting that in case of an inter-domain single-layer Ethernet link, the physical adjacency between the two router Ethernet interfaces cannot be discovered by the MDSC, using the LLDP information reported in the plug-id attributes, as shown in Figure 5. However, the MDSC may infer these links if it knows a priori, using mechanisms which are outside the scope of this document, that inter-domain Ethernet links are always single-layer, e.g., as shown in Figure 5.

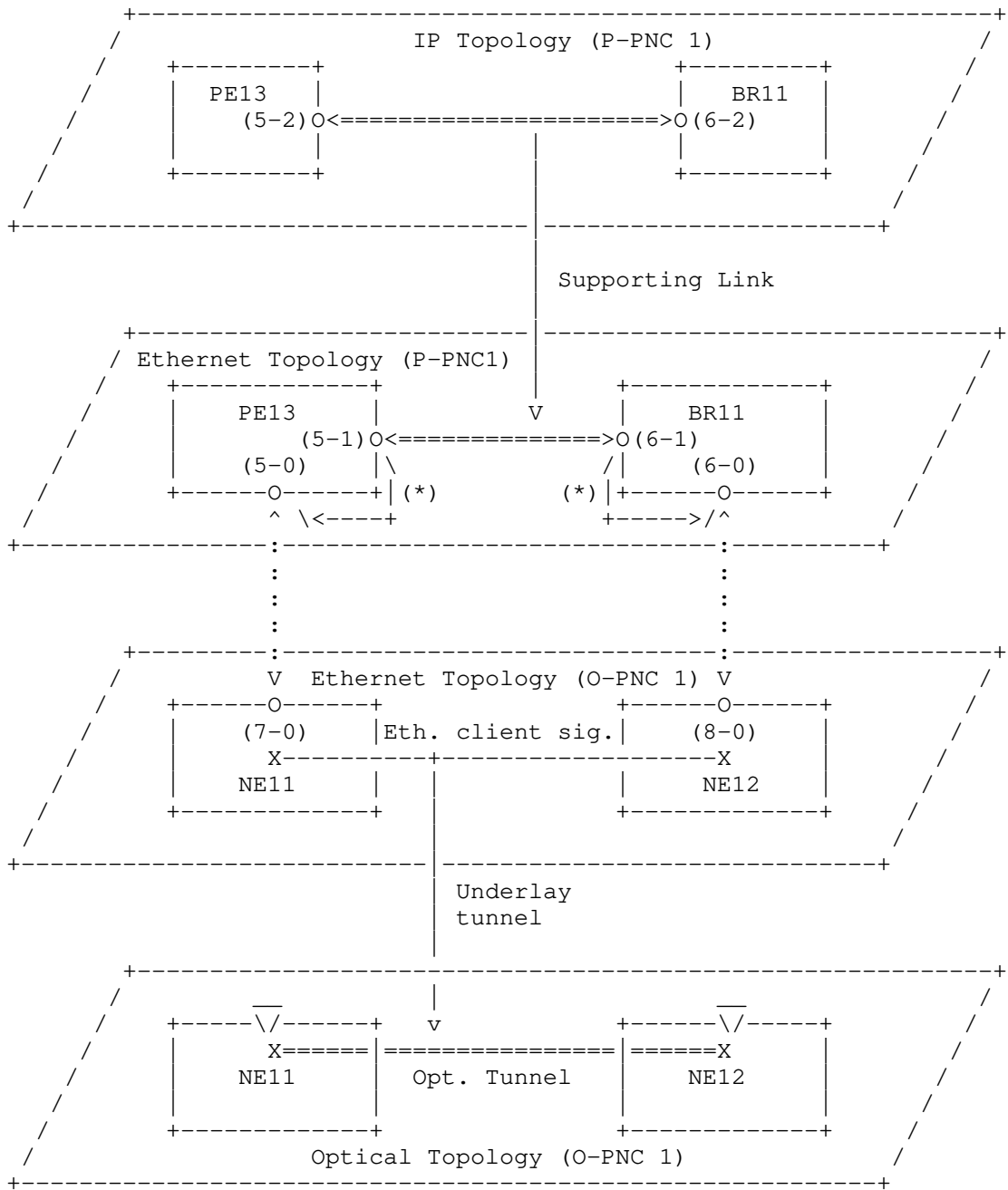
The P-PNC can omit reporting the physical Ethernet LTPs when it knows, by mechanisms which are outside the scope of this document, that the corresponding router Ethernet interfaces terminate single-layer inter-domain Ethernet links.

The MDSC can then discover an inter-domain IP link between the two IP LTPs that are supported by the two Ethernet LTPs terminating an inter-domain Ethernet link, discovered as described in section 4.5.2, e.g., between the IP LTP 3-2 on BR21 and the IP LTP 4-2 on BR22, supported respectively by the Ethernet LTP 3-1 on BR11 and by the Ethernet LTP 4-1 on BR21, as shown in Figure 5.

#### 4.6. Multi-layer IP link discovery

A multi-layer intra-domain IP link and its supporting multi-layer intra-domain Ethernet link are discovered by the P-PNC like any other

intra-domain IP and Ethernet links, as described in section 4.3, and reported at the MPI within the SR-TE and Ethernet network topologies, e.g., as shown in Figure 6.



## Notes:

=====

(\*) Supporting LTP

## Legenda:

=====

O LTP

----&gt; Supporting LTP or Supporting Link or Underlay tunnel

&lt;====&gt; Link discovered by the PNC and reported at the MPI

&lt;...&gt; Link discovered by the MDSC

&lt;~~~&gt; Link inferred by the MDSC

x---x Ethernet client signal

X===X Optical tunnel

Figure 6 - Multi-layer intra-domain Ethernet and IP link discovery

The P-PNC does not report any plug-id information on the Ethernet LTPs terminating intra-domain Ethernet links since these links are discovered by the PNC.

In addition, the P-PNC also reports the physical Ethernet LTPs that terminate the cross-layer links supporting the multi-layer intra-domain Ethernet links, e.g., the Ethernet LTP 5-0 on PE13 and the Ethernet LTP 6-0 on BR11, shown in Figure 6.

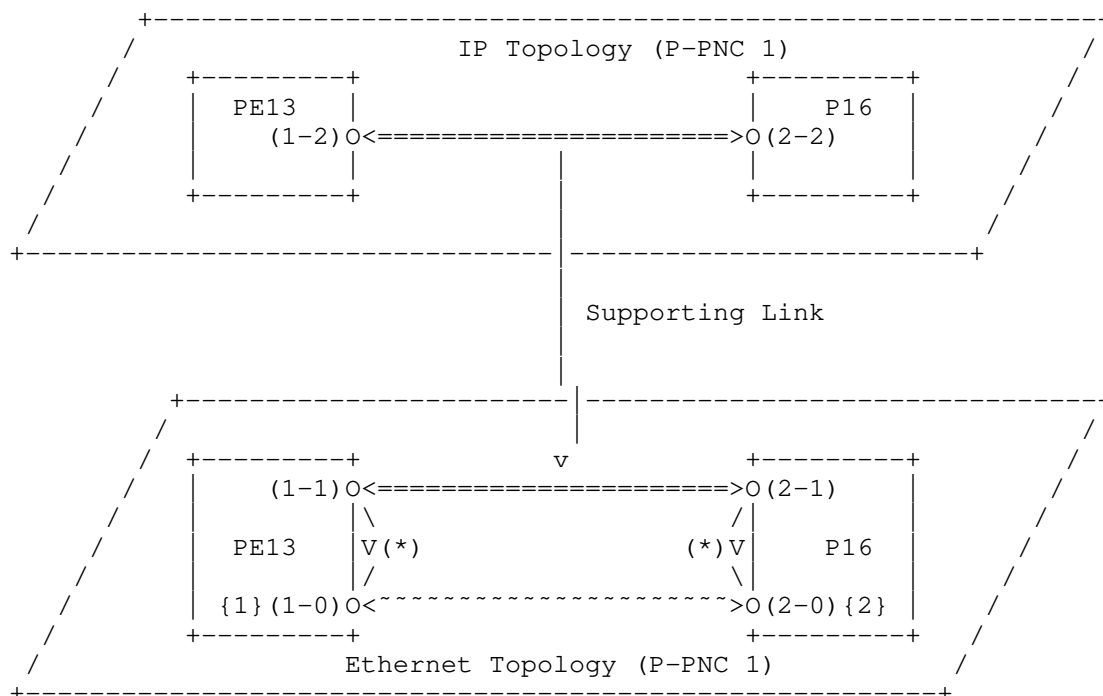
The MDSC discovers, using the mechanisms described in section 4.5, which Ethernet cross-layer links support the multi-layer intra-domain Ethernet links, e.g. as shown in Figure 6.

The MDSC also discovers, from the information provided by the O-PNC and described in section 4.2, which optical tunnels support the multi-layer intra-domain IP links and therefore the path within the optical network that supports a multi-layer intra-domain IP link, e.g., as shown in Figure 6.

## 4.6.1. Single-layer intra-domain IP links

It is worth noting that the P-PNC may not be aware of whether an Ethernet interface on the router terminates a multi-layer or a single-layer intra-domain Ethernet link.

In this case, the P-PNC, always reports two Ethernet LTPs for each Ethernet interface on the router, e.g., the Ethernet LTP 1-0 and 1-1 on PE13, shown in Figure 7.



## Notes:

=====

(\*) Supporting LTP

{1} {PE13,1}

{2} {P16,2}

## Legenda:

=====

O LTP

----&gt; Supporting LTP

&lt;===&gt; Link discovered by the PNC and reported at the MPI

&lt;~~~&gt; Link inferred by the MDSC

{ } LTP Plug-id reported by the PNC

Figure 7 - Single-layer intra-domain Ethernet and IP link discovery

In this case, the MDSC, using the plug-id information reported in the physical Ethernet LTPs, does not discover any cross-layer link being terminated by the corresponding Ethernet interface. The MDSC may infer the physical intra-domain Ethernet link, e.g., between LTP 1-0 on PE13 and LTP 2-0 on P16, as shown in Figure 7, if it knows a

priori, by mechanisms which are outside the scope of this document, that all the Ethernet interfaces on the routers either terminates a cross-layer link or a single-layer intra-domain Ethernet link.

The P-PNC can omit reporting the physical Ethernet LTP if it knows, by mechanisms which are outside the scope of this document, that the intra-domain Ethernet link is single-layer.

#### 4.7. LAG discovery

TBA

#### 4.8. L2/L3 VPN network services discovery

TBA

#### 4.9. Inventory discovery

There are no YANG data models in IETF that could be used to report at the MPI the whole inventory information discovered by a PNC.

[RFC8345] had foreseen some work for inventory as an augmentation of the network model, but no YANG data model has been developed so far.

There are also no YANG data models in IETF that could be used to correlate topology information, e.g., a link termination point (LTP), with inventory information, e.g., the physical port supporting an LTP, if any.

Inventory information through MPI and correlation with topology information is identified as a gap requiring further work and outside of the scope of this draft.

### 5. Establishment of L2/L3 VPN network services with TE requirements

In this scenario the MDSC needs to setup a multi-domain L2VPN or a multi-domain L3VPN with some SLA requirements.

The MDSC receives the request to setup a L2/L3 VPN network service from the OSS/Orchestration layer (see Appendix A).

The MDSC translates the L2/L3 VPN SLA requirements into TE requirements (e.g., bandwidth, TE metric bounds, SRLG disjointness, nodes/links/domains inclusion/exclusion) and find the SR-TE paths that meet these TE requirements (see section 2.1.1).

For example, considering the L3VPN in Figure 2, the MDSC finds that:

- o a PE13-P16-PE14 SR-TE path already exists but have not enough bandwidth to support the new L3VPN, as described in section 4.4;
- o the IP link(s) between P16 and PE14 has not enough bandwidth to support increasing the bandwidth of that SR-TE path, as described in section 4.3;
- o a new underlay optical tunnel could be setup to increase the bandwidth IP link(s) between P16 and PE14 to support increasing the bandwidth of that overlay SR-TE path, as described in section 5.2. The dimensioning of the underlay optical tunnel is decided by the MDSC based on the bandwidth requested by the SR-TE path and on its multi-layer optimization policy, which is an internal MDSC implementation issue.

Considering for example the L3VPN in Figure 2, the MDSC can also decide that a new multi-domain SR-TE path needs to be setup between PE13 and PE23, e.g., either because existing SR-TE paths between PE13 and PE23 are not able to meet the TE and binding requirements of the L2/L3 VPN service or because there is no SR-TE path between PE13 and PE23.

As described in section 2.1.2, with partial summarization, the MDSC will use the TE topology information provided by the P-PNCs and the results of the path computation requests sent to the O-PNCs, as described in section 5.1, to compute the multi-layer/multi-domain path between PE13 and PE23.

For example, the multi-layer/multi-domain performed by the MDSC could require the setup of:

- o a new underlay optical tunnel between PE13 and BR11, supporting a new IP link, as described in section 5.2;
- o a new underlay optical tunnel between BR21 and P24 to increase the bandwidth of the IP link(s) between BR21 and P24, as described in section 5.2.

When the setup of the L2/L3 VPN network service requires multi-domain and multi-layer coordination, the MDSC is also responsible for coordinating the network configuration required to realize the request network service across the appropriate optical and packet domains.



The MDSC would therefore request:

- o the O-PNC1 to setup a new optical tunnel between the ROADMs connected to P16 and PE14, as described in section 5.2;
- o the P-PNC1 to update the configuration of the existing IP link, in case of LAG, or configure a new IP link, in case of ECMP, between P16 and PE14, as described in section 5.2;
- o the P-PNC1 to update the bandwidth of the selected SR-TE path between PE13 and PE14, as described in section 5.3.

After that, the MDSC requests P-PNC2 to setup an SR-TE path between BR21 and PE23, with an explicit path (BR21, P24, PE23) to constraint this new SR-TE path to use the new underlay optical tunnel setup between BR21 and P24, as described in section 5.3. The P-PNC2, knowing the node and the adjacency SIDs assigned within its domain, can install the proper SR policy, or hierarchical policies, within BR21 and returns to the MDSC the binding SID it has assigned to this policy in BR21.

Then the MDSC requests P-PNC1 to setup an SR-TE path between PE13 and BR11, with an explicit path (PE13, BR11) to constraint this new SR-TE path to use the new underlay optical tunnel setup between PE13 and BR11, specifying also which inter-domain link should be used to send traffic to BR21 and the binding SID that has been assigned by P-PNC2 to the corresponding SR policy in BR21, to be used for the end-to-end SR-TE path stitching, as described in section 5.3. The P-PNC1, knowing also the node and the adjacency SIDs assigned within its domain and the EPE SID assigned by P-PNC1 to the inter-domain link between BR11 and BR21, and the binding SID assigned by P-PNC2, installs the proper policy, or policies, within PE13.

Once the SR-TE paths have been selected and, if needed, setup/modified, the MDSC can request to both P-PNCs to configure the L3VPN and its binding with the selected SR-TE paths using the [RFC9182] and [TSM] YANG data models.

[Editor's Note] Further investigation is needed to understand how the binding between a L3VPN and this new end-to-end SR-TE path can be configured.

### 5.1. Optical Path Computation

As described in section 2.1.2, the optical path computation is usually performed by the O-PNCs.

When performing multi-layer/multi-domain path computation, the MDSC can delegate the O-PNC for single-domain optical path computation.

As discussed in [PATH-COMPUTE], there are two options to request an O-PNC to perform optical path computation: either via a "compute-only" TE tunnel path, using the generic TE tunnel YANG data model defined in [TE-TUNNEL] or via the path computation RPC defined in [PATH-COMPUTE].

This draft assumes that the path computation RPC is used.

As described in sections 4.1 and 4.5, there is a one-to-one relationship between the router ports, the cross-layer links and the optical TTPs. Therefore, the properties of an optical path between two optical TTPs, as computed by the O-PNC, can be used by the MDSC to infer the properties of the multi-layer single-domain IP link between the router ports associated with the two optical TTPs.

There are no YANG data models in IETF that could be used to augment the generic path computation RPC with technology-specific attributes.

Optical technology-specific augmentation for the path computation RPC is identified as a gap requiring further work outside of this draft's scope.

## 5.2. Multi-layer IP link Setup

To setup a new multi-layer IP link between two router ports, the MDSC requires the O-PNC to setup an optical tunnel (either a WSON Tunnel or a Flexi-grid Tunnel or an OTN Tunnel) within the optical network between the two TTPs associated, as described in section 5.1, with these two router Ethernet interfaces.

The MDSC also requires the O-PNC to steer the Ethernet client traffic between the two cross-layer links over the optical tunnel using the Ethernet Client Signal Model.

After the optical tunnel has been setup and the client traffic steering configured, the two IP routers can exchange Ethernet packets between themselves, including LLDP messages.

If LLDP [IEEE 802.1AB] or any other discovery mechanisms, which are outside the scope of this document, is used between the adjacency between the two routers' ports, the P-PNC can automatically discover the underlay multi-layer single-domain Ethernet link being set up by the MDSC and report it to the P-PNC.

Otherwise, if there are no automatic discovery mechanisms, the MDSC can configure this multi-layer single-domain Ethernet link at the MPI of the P-PNC.

The two Ethernet LTPs terminating this multi-layer single-domain Ethernet link are supported by the two underlay Ethernet LTPs terminating the two cross-layer links, e.g., as shown in Figure 6.

After the multi-layer single-domain Ethernet link has been configured, the corresponding multi-layer single-domain IP link can also be configured either by the MDSC or by the P-PNC.

This document assumes that this IP link is configured by the P-PNC, when the underlying multi-layer single-domain Ethernet link is either discovered by the P-PNC or configured by the MDSC at the MPI.

[Editor's Note] Add text for IP link update in case of LAG either here or in a new section.

[Editor's Note] Add text about the configuration of multi-layer SRLG information (issue #45).

It is worth noting that the list of SRLGs for a multi-layer IP link can be quite long. Implementation-specific mechanisms can be implemented by the MDSC or by the P-PNC to summarize the SRLGs of an optical tunnel. These mechanisms are implementation-specific and have no impact on the YANG models nor on the interoperability at the MPI, but cares have to be taken to avoid missing information.

### 5.3. SR-TE Path Setup and Update

This version of the draft assumes that SR-TE path setup and update at the MPI could be done using the generic TE tunnel YANG data model, defined in [TE-TUNNEL], with SR-TE specific augmentations, as also outlined in section 1 of [TE-TUNNEL].

When a new SR-TE path needs to be setup, the MDSC can use the [TE-TUNNEL] model to request the P-PNC to setup TE paths, properly specifying the path constraints, such as the explicit path, to force the P-PNC to setup an SR-TE path that meets the end-to-end TE and bidding constraints and uses the optical tunnels setup by the MDSC for the purpose of supporting this new SR-TE path.

The [TE-TUNNEL] model supports requesting the setup of both end-to-end as well as segment TE tunnels (within one domain).

In the latter case, SR-TE specific augmentations of the [TE-TUNNEL] model should be defined to allow the MDSC to configure the binding SIDs to be used for the end to-end SR-TE path stitching and to allow the P-PNC to report the binding SID assigned to the segment TE paths.

The assigned binding SID should be persistent in case router or P-PNC rebooting.

The MDSC can also use the [TE-TUNNEL] model to request the P-PNC to increase the bandwidth allocated to an existing TE path, and, if needed, also on its reverse TE path. The [TE-TUNNEL] model supports both symmetric and asymmetric bandwidth configuration in the two directions.

[Editor's Note:] Add some text about the protection options (to further discuss whether to put this text here or in section 4.2.2).

The MDSC also request the P-PNC to configure TI-LFA local protection: the mechanisms to request the configuration TI-LFA local protection for SR-TE paths using the [TE-TUNNEL] are a gap in the current YANG models.

The TI-LFA local protection within the P-PNC domain is configured by the P-PNC through implementation specific mechanisms which are outside the scope of this document. The P-PNC takes into account the multi-layer SRLG information, configured by the MDSC as described in section 5.2, when computing the TI-LFA post-convergence path for multi-layer single-domain IP links.

SR-TE path setup and update (e.g., bandwidth increase) through MPI is identified as a gap requiring further work, which is outside of the scope of this draft.

## 6. Conclusions

The analysis provided in this document has shown that the IETF YANG models described in 3.2 provides useful support for Packet Optical Integration (POI) scenarios for resource discovery (network topology, service, tunnels and network inventory discovery) as well as for supporting multi-layer/multi-domain L2/L3 VPN network services.

Few gaps have been identified to be addressed by the relevant IETF Working Groups:

- o network inventory model: this gap has been identified in section 4.9 and the solution in [NETWORK-INVENTORY] has been proposed to resolve it;
- o technology-specific augmentations of the path computation RPC, defined in [PATH-COMPUTE] for optical networks: this gap has been identified in section 5.1 and the solution in [OPTICAL-PATH-COMPUTE] has been proposed to resolve it;
- o relationship between a common discovery mechanisms applicable to access links, inter-domain IP links and cross-layer links and the UNI topology discover mechanism defined in [SAP]: this gap has been identified in section 4.3;
- o a mechanism applicable to the P-PNC NBI to configure the SR-TE paths. Technology-specific augmentations of TE Tunnel model, defined in [TE-TUNNEL], are foreseen in section 1 of [TE-TUNNEL] but not yet defined: this gap has been identified in section 5.3.

## 7. Security Considerations

Several security considerations have been identified and will be discussed in future versions of this document.

## 8. Operational Considerations

Telemetry data, such as collecting lower-layer networking health and consideration of network and service performance from POI domain controllers, may be required. These requirements and capabilities will be discussed in future versions of this document.

## 9. IANA Considerations

This document requires no IANA actions.

## 10. References

### 10.1. Normative References

- [RFC7923] Voit, E. et al., "Requirements for Subscription to YANG Datastores", RFC 7923, June 2016.
- [RFC7950] Bjorklund, M. et al., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.

- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, August 2016.
- [RFC8040] Bierman, A. et al., "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8342] Bjorklund, M. et al., "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.
- [RFC8345] Clemm, A., Medved, J. et al., "A Yang Data Model for Network Topologies", RFC8345, March 2018.
- [RFC8346] Clemm, A. et al., "A YANG Data Model for Layer 3 Topologies", RFC8346, March 2018.
- [RFC8453] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC8453, August 2018.
- [RFC8525] Bierman, A. et al., "YANG Library", RFC 8525, March 2019.
- [RFC8527] Bjorklund, M. et al., "RESTCONF Extensions to Support the Network Management Datastore Architecture", RFC 8527, March 2019.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, September 2019.
- [RFC8650] Voit, E. et al., "Dynamic Subscription to YANG Events and Datastores over RESTCONF", RFC 8650, November 2019.
- [RFC8795] Liu, X. et al., "YANG Data Model for Traffic Engineering (TE) Topologies", RFC8795, August 2020.
- [RFC9094] Zheng H., Lee, Y. et al., "A YANG Data Model for Wavelength Switched Optical Networks (WSOs)", RFC 9094, August 2021.
- [IEEE 802.1AB] IEEE 802.1AB-2016, "IEEE Standard for Local and metropolitan area networks – Station and Media Access Control Connectivity Discovery", March 2016.
- [Flexi-TOPO] Lopez de Vergara, J. E. et al., "YANG data model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang, work in progress.

- [OTN-TOPO] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang, work in progress.
- [CLIENT-TOPO] Zheng, H. et al., "A YANG Data Model for Client-layer Topology", draft-zheng-ccamp-client-topo-yang, work in progress.
- [L3-TE-TOPO] Liu, X. et al., "YANG Data Model for Layer 3 TE Topologies", draft-ietf-teas-yang-l3-te-topo, work in progress.
- [SR-TE-TOPO] Liu, X. et al., "YANG Data Model for SR and SR TE Topologies on MPLS Data Plane", draft-ietf-teas-yang-sr-te-topo, work in progress.
- [TE-TUNNEL] Saad, T. et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.
- [WSON-TUNNEL] Lee, Y. et al., "A Yang Data Model for WSON Tunnel", draft-ietf-ccamp-wson-tunnel-model, work in progress.
- [Flexi-TUNNEL] Lopez de Vergara, J. E. et al., "A YANG Data Model for Flexi-Grid Tunnels ", draft-ietf-ccamp-flexigrid-tunnel-yang, work in progress.
- [OTN-TUNNEL] Zheng, H. et al., "OTN Tunnel YANG Model", draft-ietf-ccamp-otn-tunnel-model, work in progress.
- [PATH-COMPUTE] Busi, I., Belotti, S. et al, "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation, work in progress.
- [CLIENT-SIGNAL] Zheng, H. et al., "A YANG Data Model for Transport Network Client Signals", draft-ietf-ccamp-client-signal-yang, work in progress.

## 10.2. Informative References

- [RFC1930] J. Hawkinson, T. Bates, "Guideline for creation, selection, and registration of an Autonomous System (AS)", RFC 1930, March 1996.
- [RFC5440] Vasseur, JP. et al., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

- [RFC5623] Oki, E. et al., "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009.
- [RFC8231] Crabbe, E. et al., "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, September 2017.
- [RFC8281] Crabbe, E. et al., "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, December 2017.
- [RFC8283] Farrel, A. et al., "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", RFC 8283, December 2017.
- [RFC8309] Q. Wu, W. Liu, and A. Farrel, "Service Model Explained", RFC 8309, January 2018.
- [RFC8637] Dhody, D. et al., "Applicability of the Path Computation Element (PCE) to the Abstraction and Control of TE Networks (ACTN)", RFC 8637, July 2019.
- [RFC8751] Dhody, D. et al., "Hierarchical Stateful Path Computation Element (PCE)", RFC 8751, March 2020.
- [RFC9182] S. Barguil, et al., "A YANG Network Data Model for Layer 3 VPNs", RFC 9182, February 2022.
- [L2NM] S. Barguil, et al., "A Layer 2 VPN Network YANG Model", draft-ietf-opsawg-l2nm, work in progress.
- [TSM] Y. Lee, et al., "Traffic Engineering and Service Mapping Yang Model", draft-ietf-teas-te-service-mapping-yang, work in progress.
- [TNBI] Busi, I., Daniel, K. et al., "Transport Northbound Interface Applicability Statement", draft-ietf-ccamp-transport-nbi-app-statement, work in progress.
- [VN] Y. Lee, et al., "A Yang Data Model for ACTN VN Operation", draft-ietf-teas-actn-vn-yang, work in progress.
- [OIA-TOPO] Lee Y. et al., "A YANG Data Model for Optical Impairment-aware Topology", draft-ietf-ccamp-optical-impairment-topology-yang, work in progress.



- [SAP] Gonzalez de Dios O. et al., "A Network YANG Model for Service Attachment Points (SAPs)", draft-ietf-opsawg-sap, work in progress.
- [NETWORK-INVENTORY] Yu C. et al., "A YANG Data Model for Optical Network Inventory", draft-yg3bp-ccamp-optical-inventory-yang, work in progress.
- [OPTICAL-PATH-COMPUTE] Busi I. et al., "YANG Data Models for requesting Path Computation in Optical Networks", draft-gbb-ccamp-optical-path-computation-yang, work in progress.

## Appendix A. OSS/Orchestration Layer

The OSS/Orchestration layer is a vital part of the architecture framework for a service provider:

- o to abstract (through MDSC and PNCs) the underlying transport network complexity to the Business Systems Support layer;
- o to coordinate NFV, Transport (e.g. IP, optical and microwave networks), Fixed Access, Core and Radio domains enabling full automation of end-to-end services to the end customers;
- o to enable catalogue-driven service provisioning from external applications (e.g. Customer Portal for Enterprise Business services), orchestrating the design and lifecycle management of these end-to-end transport connectivity services, consuming IP and/or optical transport connectivity services upon request.

As discussed in section 2.1, in this document, the MDSC interfaces with the OSS/Orchestration layer and, therefore, it performs the functions of the Network Orchestrator, defined in [RFC8309].

The OSS/Orchestration layer requests the creation of a network service to the MDSC specifying its end-points (PEs and the interfaces towards the CEs) as well as the network service SLA and then proceeds to configuring accordingly the end-to-end customer service between the CEs in the case of an operator managed service.

### A.1. MDSC NBI

As explained in section 2, the OSS/Orchestration layer can request the MDSC to setup L2/L3VPN network services (with or without TE requirements).

Although the OSS/Orchestration layer interface is usually operator-specific, typically it would be using a RESTCONF/YANG interface with a more abstracted version of the MPI YANG data models used for network configuration (e.g. L3NM, L2NM).

Figure 8 shows an example of possible control flow between the OSS/Orchestration layer and the MDSC to instantiate L2/L3 VPN network services, using the YANG data models under the definition in [VN], [L2NM], [RFC9182] and [TSM].

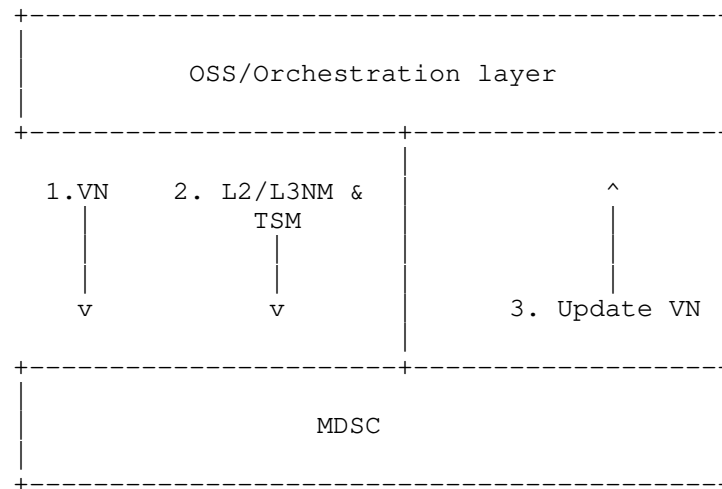


Figure 8 Service Request Process

- o The VN YANG data model, defined in [VN], whose primary focus is the CMI, can also provide VN Service configuration from an orchestrated network service point of view when the L2/L3 VPN network service has TE requirements. However, this model is not used to setup L2/L3 VPN service with no TE requirements.
  - o It provides the profile of VN in terms of VN members, each of which corresponds to an edge-to-edge link between customer end-points (VNAPs). It also provides the mappings between the VNAPs with the LTPs and the connectivity matrix with the VN member. The associated traffic matrix (e.g., bandwidth, latency, protection level, etc.) of VN member is expressed (i.e., via the TE-topology's connectivity matrix).
  - o The model also provides VN-level preference information (e.g., VN member diversity) and VN-level admin-status and operational-status.
- o The L2NM and L3NM YANG data models, defined in [L2NM] and [RFC9182], whose primary focus is the MPI, can also be used to provide L2VPN and L3VPN network service configuration from a orchestrated connectivity service point of view.
- o The TE & Service Mapping YANG data model [TSM] provides TE-service mapping.

- o TE-service mapping provides the mapping between a L2/L3 VPN instance and the corresponding VN instances.
- o The TE-service mapping also provides the binding requirements as to how each L2/L3 VPN/VN instance is created concerning the underlay TE tunnels (e.g., whether they require a new and isolated set of TE underlay tunnels or not).
- o Site mapping provides the site reference information across L2/L3 VPN Site ID, VN Access Point ID, and the LTP of the access link.

## Appendix B. Multi-layer and multi-domain resiliency

### B.1. Maintenance Window

Before planned maintenance operation on DWDM network takes place, IP traffic should be moved hitless to another link.

MDSC must reroute IP traffic before the events takes place. It should be possible to lock IP traffic to the protection route until the maintenance event is finished, unless a fault occurs on such path.

### B.2. Router port failure

The focus is on client-side protection scheme between IP router and reconfigurable ROADM. Scenario here is to define only one port in the routers and in the ROADM muxponder board at both ends as back-up ports to recover any other port failure on client-side of the ROADM (either on router port side or on muxponder side or on the link between them). When client-side port failure occurs, alarms are raised to MDSC by IP-PNC and O-PNC (port status down, LOS etc.). MDSC checks with OP-PNC(s) that there is no optical failure in the optical layer.

There can be two cases here:

- a) LAG was defined between the two end routers. MDSC, after checking that optical layer is fine between the two end ROADMs, triggers the ROADM configuration so that the router back-up port with its associated muxponder port can reuse the OCh that was already in use previously by the failed router port and adds the new link to the LAG on the failure side.

While the ROADM reconfiguration takes place, IP/MPLS traffic is using the reduced bandwidth of the IP link bundle, discarding lower priority traffic if required. Once back-up port has been reconfigured to reuse the existing OCh and new link has been added to the LAG then original Bandwidth is recovered between the end routers.

Note: in this LAG scenario let assume that BFD is running at LAG level so that there is nothing triggered at MPLS level when one of the link member of the LAG fails.

- b) If there is no LAG then the scenario is not clear since a router port failure would automatically trigger (through BFD failure) first a sub-50ms protection at MPLS level :FRR (MPLS RSVP-TE case) or TI-LFA (MPLS based SR-TE case) through a protection port. At the same time MDSC, after checking that optical network connection is still fine, would trigger the reconfiguration of the back-up port of the router and of the ROADM muxponder to re-use the same OCh as the one used originally for the failed router port. Once everything has been correctly configured, MDSC Global PCE could suggest to the operator to trigger a possible re-optimization of the back-up MPLS path to go back to the MPLS primary path through the back-up port of the router and the original OCh if overall cost, latency etc. is improved. However, in this scenario, there is a need for protection port PLUS back-up port in the router which does not lead to clear port savings.

#### Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Some of this analysis work was supported in part by the European Commission funded H2020-ICT-2016-2 METRO-HAUL project (G.A. 761727).

#### Contributors

Sergio Belotti  
Nokia

Email: sergio.belotti@nokia.com

Gabriele Galimberti  
Cisco

Email: ggalimbe@cisco.com

Zheng Yanlei  
China Unicom

Email: zhengyanlei@chinaunicom.cn

Anton Snitser  
Sedona

Email: antons@sedonasys.com

Washington Costa Pereira Correia  
TIM Brasil

Email: wcorreia@timbrasil.com.br

Michael Scharf  
Hochschule Esslingen - University of Applied Sciences

Email: michael.scharf@hs-esslingen.de

Young Lee  
Sung Kyun Kwan University

Email: younglee.tx@gmail.com

Jeff Tantsura  
Apstra

Email: jefftant.ietf@gmail.com

Paolo Volpato  
Huawei

Email: paolo.volpato@huawei.com

Brent Foster  
Cisco

Email: brfoster@cisco.com

Authors' Addresses

Fabio Peruzzini  
TIM

Email: fabio.peruzzini@telecomitalia.it

Jean-Francois Bouquier  
Vodafone

Email: jeff.bouquier@vodafone.com

Italo Busi  
Huawei

Email: Italo.busi@huawei.com

Daniel King  
Old Dog Consulting

Email: daniel@olddog.co.uk

Daniele Ceccarelli  
Ericsson

Email: daniele.ceccarelli@ericsson.com





Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: November 24, 2021

A. Farrel, Ed.  
Old Dog Consulting  
E. Gray  
Independent  
J. Drake  
Juniper Networks  
R. Rokui  
Nokia  
S. Homma  
NTT  
K. Makhijani  
Futurewei  
LM. Contreras  
Telefonica  
J. Tantsura  
Juniper Networks  
May 23, 2021

Framework for IETF Network Slices  
draft-ietf-teas-ietf-network-slices-03

Abstract

This document describes network slicing in the context of networks built from IETF technologies. It defines the term "IETF Network Slice" and establishes the general principles of network slicing in the IETF context.

The document discusses the general framework for requesting and operating IETF Network Slices, the characteristics of an IETF Network Slice, the necessary system components and interfaces, and how abstract requests can be mapped to more specific technologies. The document also discusses related considerations with monitoring and security.

This document also provides definitions of related terms to enable consistent usage in other IETF documents that describe or use aspects of IETF Network Slices.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 24, 2021.

#### Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                                 | 3  |
| 1.1. Background . . . . .                                 | 4  |
| 2. Terms and Abbreviations . . . . .                      | 5  |
| 2.1. Core Terminology . . . . .                           | 6  |
| 3. IETF Network Slice Objectives . . . . .                | 6  |
| 3.1. Definition and Scope of IETF Network Slice . . . . . | 6  |
| 4. IETF Network Slice System Characteristics . . . . .    | 7  |
| 4.1. Objectives for IETF Network Slices . . . . .         | 7  |
| 4.1.1. Service Level Objectives . . . . .                 | 8  |
| 4.1.2. Service Level Expectations . . . . .               | 10 |
| 4.2. IETF Network Slice Endpoints . . . . .               | 12 |
| 4.2.1. IETF Network Slice Connectivity Types . . . . .    | 14 |
| 4.3. IETF Network Slice Decomposition . . . . .           | 14 |
| 5. Framework . . . . .                                    | 15 |
| 5.1. IETF Network Slice Stakeholders . . . . .            | 15 |
| 5.2. Expressing Connectivity Intents . . . . .            | 15 |
| 5.3. IETF Network Slice Controller (NSC) . . . . .        | 17 |
| 5.3.1. IETF Network Slice Controller Interfaces . . . . . | 19 |
| 5.3.2. Northbound Interface (NBI) . . . . .               | 20 |
| 5.4. IETF Network Slice Structure . . . . .               | 21 |
| 6. Realizing IETF Network Slices . . . . .                | 22 |

|  |    |
|--|----|
| 6.1. Procedures to Realize IETF Network Slices . . . . .       | 22 |
| 6.2. Applicability of ACTN to IETF Network Slices . . . . .    | 23 |
| 6.3. Applicability of Enhanced VPNs to IETF Network Slices . . | 23 |
| 6.4. Network Slicing and Slice Aggregation in IP/MPLS Networks | 24 |
| 7. Isolation in IETF Network Slices . . . . .                  | 24 |
| 7.1. Isolation as a Service Requirement . . . . .              | 24 |
| 7.2. Isolation in IETF Network Slice Realization . . . . .     | 24 |
| 8. Management Considerations . . . . .                         | 25 |
| 9. Security Considerations . . . . .                           | 25 |
| 10. Privacy Considerations . . . . .                           | 26 |
| 11. IANA Considerations . . . . .                              | 26 |
| 12. Informative References . . . . .                           | 26 |
| Acknowledgments . . . . .                                      | 30 |
| Contributors . . . . .   | 31 |
| Authors' Addresses . . . . .                                   | 31 |

## 1. Introduction

A number of use cases benefit from network connections that along with the connectivity provide assurance of meeting a specific set of objectives with respect to network resources use. This connectivity and resource commitment is referred to as a network slice. Since the term network slice is rather generic, the qualifying term "IETF" is used in this document to limit the scope of network slice to network technologies described and standardized by the IETF. This document defines the concept of IETF Network Slices that provide connectivity coupled with a set of specific commitments of network resources between a number of endpoints over a shared network infrastructure. Services that might benefit from IETF Network Slices include, but are not limited to:

- o 5G services (e.g. eMBB, URLLC, mMTC) (See [TS23501])
- o Network wholesale services
- o Network infrastructure sharing among operators
- o NFV connectivity and Data Center Interconnect

IETF Network Slices are created and managed within the scope of one or more network technologies (e.g., IP, MPLS, optical). They are intended to enable a diverse set of applications that have different requirements to coexist on the shared network infrastructure. A request for an IETF Network Slice is technology-agnostic so as to allow a customer to describe their network connectivity objectives in a common format, independent of the underlying technologies used.

This document also provides a framework for discussing IETF Network Slices. This framework is intended as a structure for discussing interfaces and technologies. It is not intended to specify a new set of concrete interfaces or technologies. Rather, the idea is that existing or under-development IETF technologies (plural) can be used to realize the concepts expressed herein.

For example, virtual private networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated access to a common network. The common or base network that is used to support the VPNs is often referred to as an underlay network, and the VPN is often called an overlay network. An overlay network may, in turn, serve as an underlay network to support another overlay network.

Note that it is conceivable that extensions to these IETF technologies are needed in order to fully support all the ideas that can be implemented with slices. Evaluation of existing technologies, proposed extensions to existing protocols and interfaces, and the creation of new protocols or interfaces is outside the scope of this document.

### 1.1. Background

Driven largely by needs surfacing from 5G, the concept of network slicing has gained traction ([NGMN-NS-Concept], [TS23501], [TS28530], and [BBF-SD406]). In [TS23501], a Network Slice is defined as "a logical network that provides specific network capabilities and network characteristics", and a Network Slice Instance is defined as "A set of Network Function instances and the required resources (e.g. compute, storage and networking resources) which form a deployed Network Slice." According to [TS28530], an end-to-end network slice consists of three major types of network segments: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN). An IETF Network Slice provides the required connectivity between different entities in RAN and CN segments of an end-to-end network slice, with a specific performance commitment. For each end-to-end network slice, the topology and performance requirement on a customer's use of IETF Network Slice can be very different, which requires the underlay network to have the capability of supporting multiple different IETF Network Slices.

While network slices are commonly discussed in the context of 5G, it is important to note that IETF Network Slices are a narrower concept, and focus primarily on particular network connectivity aspects. Other systems, including 5G deployments, may use IETF Network Slices as a component to create entire systems and concatenated constructs that match their needs, including end-to-end connectivity.

A IETF Network Slice could span multiple technologies and multiple administrative domains. Depending on the IETF Network Slice customer's requirements, an IETF Network Slice could be isolated from other, often concurrent IETF Network Slices in terms of data, control and management planes.

The customer expresses requirements for a particular IETF Network Slice by specifying what is required rather than how the requirement is to be fulfilled. That is, the IETF Network Slice customer's view of an IETF Network Slice is an abstract one.

Thus, there is a need to create logical network structures with required characteristics. The customer of such a logical network can require a degree of isolation and performance that previously might not have been satisfied by traditional overlay VPNs. Additionally, the IETF Network Slice customer might ask for some level of control of their virtual networks, e.g., to customize the service paths in a network slice.

This document specifies definitions and a framework for the provision of an IETF Network Slice service. Section 6 briefly indicates some candidate technologies for realizing IETF Network Slices.

## 2. Terms and Abbreviations

The following abbreviations are used in this document.

- o NBI: NorthBound Interface
- o NSC: Network Slice Controller
- o NSE: Network Slice Endpoint
- o SBI: SouthBound Interface
- o SLA: Service Level Agreement
- o SLI: Service Level Indicator
- o SLO: Service Level Objective

The meaning of these abbreviations is defined in greater details in the remainder of this document.

## 2.1. Core Terminology

The following terms are presented here to give context. Other terminology is defined in the remainder of this document.

**Customer:** A customer is the requester of an IETF Network Slice service. Customers may request monitoring of SLOs. A customer may be an entity such as an enterprise network or a network operator, an individual working at such an entity, a private individual contracting for a service, or an application or software component. A customer may be an external party (classically a paying customer) or a division of a network operator that uses the service provided by another division of the same operator. Other terms that have been applied to the customer role are "client" and "consumer".

**Provider:** A provider is the organization that delivers an IETF Network Slice service. A provider is the network operator that controls the network resources used to construct the network slice (that is, the network that is sliced). The provider's network maybe a physical network or may be a virtual network supplied by another service provider.

## 3. IETF Network Slice Objectives

It is intended that IETF Network Slices can be created to meet specific requirements, typically expressed as bandwidth, latency, latency variation, and other desired or required characteristics. Creation is initiated by a management system or other application used to specify network-related conditions for particular traffic flows.

It is also intended that, once created, these slices can be monitored, modified, deleted, and otherwise managed.

It is also intended that applications and components will be able to use these IETF Network Slices to move packets between the specified end-points in accordance with specified characteristics.

### 3.1. Definition and Scope of IETF Network Slice

The definition of a network slice in IETF context is as follows:

An IETF Network Slice is a logical network topology connecting a number of endpoints using a set of shared or dedicated network resources that are used to satisfy specific Service Level Objectives (SLOs).

An IETF Network Slice combines the connectivity resource requirements and associated network behaviors such as bandwidth, latency, jitter, and network functions with other resource behaviors such as compute and storage availability. IETF Network Slices are independent of the underlying infrastructure connectivity and technologies used. This is to allow an IETF Network Slice service customer to describe their network connectivity and relevant objectives in a common format, independent of the underlying technologies used.

IETF Network Slices may be combined hierarchically, so that a network slice may itself be sliced. They may also be combined sequentially so that various different networks can each be sliced and the network slices placed into a sequence to provide an end-to-end service. This form of sequential combination is utilized in some services such as in 3GPP's 5G network [TS23501].

An IETF Network Slice is technology-agnostic, and the means for IETF Network Slice realization can be chosen depending on several factors such as: service requirements, specifications or capabilities of underlying infrastructure. The structure and different characteristics of IETF Network Slices are described in the following sections.

Term "Slice" refers to a set of characteristics and behaviours that separate one type of user-traffic from another. IETF Network Slice assumes that an underlying network is capable of changing the configurations of the network devices on demand, through in-band signaling or via controller(s) and fulfilling all or some of SLOs to all of the traffic in the slice or to specific flows.

#### 4. IETF Network Slice System Characteristics

The following subsections describe the characteristics of IETF Network Slices.

##### 4.1. Objectives for IETF Network Slices

An IETF Network Slice service is defined in terms of quantifiable characteristics known as Service Level Objectives (SLOs) and unquantifiable characteristics known as Service Level Expectations (SLEs). SLOs are expressed in terms Service Level Indicators (SLIs), and together with the SLEs form the contractual agreement between service customer and service provider known as a Service Level Agreement (SLA).

The terms are defined as follows:



- o A Service Level Indicator (SLI) is a quantifiable measure of an aspect of the performance of a network. For example, it may be a measure of throughput in bits per second, or it may be a measure of latency in milliseconds.
- o A Service Level Objective (SLO) is a target value or range for the measurements returned by observation of an SLI. For example, an SLO may be expressed as "SLI <= target", or "lower bound <= SLI <= upper bound". A customer can determine whether the provider is meeting the SLOs by performing measurements on the traffic.
- o A Service Level Expectation (SLE) is an expression of an unmeasurable service-related request that a customer of an IETF network slice makes of the provider. An SLE is distinct from an SLO because the customer may have little or no way of determining whether the SLE is being met, but they still contract with the provider for a service that meets the expectation.
- o A Service Level Agreement (SLA) is an explicit or implicit contract between the customer of an IETF Network Slice and the provider of the slice. The SLA is expressed in terms of a set of SLOs and SLEs that are to be applied to the connections between the service endpoints, and may include commercial terms as well as the consequences of missing/violating the SLOs they contain.

#### 4.1.1. Service Level Objectives

SLOs define a set of network attributes and characteristics that describe an IETF Network Slice. SLOs do not describe how the IETF Network Slices are implemented or realized in the underlying network layers. Instead, they are defined in terms of dimensions of operation (time, capacity, etc.), availability, and other attributes. An IETF Network Slice can have one or more SLOs associated with it. The SLOs are combined in an SLA. The SLOs are defined for sets of two or more endpoints and apply to specific directions of traffic flow. That is, they apply to specific source endpoints and specific connections between endpoints within the set of endpoints and connections in the IETF Network Slice.

SLOs define a set of measurable network attributes and characteristics that describe an IETF Network Slice service. SLOs do not describe how the IETF network slices are implemented or realized in the underlying network layers. Instead, they are defined in terms of dimensions of operation (time, capacity, etc.), availability, and other attributes. An IETF Network Slice service can have one or more SLOs associated with it. The SLOs are combined with Service Level Expectations in an SLA.

An IETF network slice service may include multiple connection constructs that associate sets of endpoints. SLOs apply to sets of two or more endpoints and apply to specific directions of traffic flow. That is, they apply to a specific source endpoint and the connection to specific destination endpoints.

#### 4.1.1.1. Some Common SLOs

SLOs can be described as 'Directly Measurable Objectives': they are always measurable. See Section 4.1.2 for the description of Service Level Expectations which are unmeasurable service-related requests sometimes known as 'Indirectly Measurable Objectives'.

Objectives such as guaranteed minimum bandwidth, guaranteed maximum latency, maximum permissible delay variation, maximum permissible packet loss rate, and availability are 'Directly Measurable Objectives'. Future specifications (such as IETF Network Slice service YANG models) may precisely define these SLOs, and other SLOs may be introduced as described in Section 4.1.1.2.

The definition of these objectives are as follows:

##### Guaranteed Minimum Bandwidth

Minimum guaranteed bandwidth between two endpoints at any time. The bandwidth is measured in data rate units of bits per second and is measured unidirectionally.

##### Guaranteed Maximum Latency

Upper bound of network latency when transmitting between two endpoints. The latency is measured in terms of network characteristics (excluding application-level latency). [RFC2681] and [RFC7679] discuss round trip times and one-way metrics, respectively.

##### Maximum Permissible Delay Variation

Packet delay variation (PDV) as defined by [RFC3393], is the difference in the one-way delay between sequential packets in a flow. This SLO sets a maximum value PDV for packets between two endpoints.

##### Maximum Permissible Packet Loss Rate

The ratio of packets dropped to packets transmitted between two endpoints over a period of time. See [RFC7680].

#### Availability

The ratio of uptime to the sum of uptime and downtime, where uptime is the time the IETF Network Slice is available in accordance with the SLOs associated with it.

##### 4.1.1.2. Other Service Level Objectives

Additional SLOs may be defined to provide additional description of the IETF Network Slice service that a customer requests. These would be specified in further documents.

If the IETF network slice service is traffic aware, other traffic specific characteristics may be valuable including MTU, traffic-type (e.g., IPv4, IPv6, Ethernet or unstructured), or a higher-level behavior to process traffic according to user-application (which may be realized using network functions).

##### 4.1.2. Service Level Expectations

SLEs define a set of network attributes and characteristics that describe an IETF Network Slice service, but which are not directly measurable by the customer. Even though the delivery of an SLE cannot usually be determined by the customer, the SLEs form an important part of the contract between customer and provider.

Quite often, an SLE will imply some details of how an IETF Network Slice service is realized by the provider, although most aspects of the implementation in the underlying network layers remain a free choice for the provider.

SLEs may be seen as aspirational on the part of the customer, and they are expressed as behaviors that the provider is expected to apply to the network resources used to deliver the IETF Network Slice service. An IETF network slice service can have one or more SLEs associated with it. The SLEs are combined with SLOs in an SLA.

An IETF Network Slice service may include multiple connection constructs that associate sets of endpoints. SLEs apply to sets of two or more endpoints and apply to specific directions of traffic flow. That is, they apply to a specific source endpoint and the connection to specific destination endpoints. However, being more general in nature, SLEs may commonly be applied to all connection constructs in an IETF Network Slice service.

#### 4.1.2.1. Some Common SLEs

SLEs can be described as 'Indirectly Measurable Objectives': they are not generally directly measurable by the customer.

Security, geographic restrictions, maximum occupancy level, and isolation are example SLEs as follows.

##### Security

A customer may request that the provider applies encryption or other security techniques to traffic flowing between endpoints of an IETF Network Slice service. For example, the customer could request that only network links that have MACsec [MACsec] enabled are used to realize the IETF Network Slice service.

This SLE may include the request for encryption (e.g., [RFC4303]) between the two endpoints explicitly to meet architecture recommendations as in [TS33.210] or for compliance with [HIPAA] or [PCI].

Whether or not the provider has met this SLE is generally not directly observable by the customer and cannot be measured as a quantifiable metric.

Please see further discussion on security in Section 9.

##### Geographic Restrictions

A customer may request that certain geographic limits are applied to how the provider routes traffic for the IETF Network Slice service. For example, the customer may have a preference that its traffic does not pass through a particular country for political or security reasons.

Whether or not the provider has met this SLE is generally not directly observable by the customer and cannot be measured as a quantifiable metric.

##### Maximal Occupancy Level

The maximal occupancy level specifies the number of flows to be admitted and optionally a maximum number of countable resource units (e.g., IP or MAC addresses) an IETF network slice service can consume. Since an IETF Network Slice service may include multiple connection constructs, this SLE should also say whether it applies for the entire IETF Network Service slice, for group of connections, or on a per connection basis.

Again, a customer may not be able to fully determine whether this SLE is being met by the provider.

#### Isolation

As described in Section 7, a customer may request that its traffic within its IETF Network Slice service is isolated from the effects of other network services supported by the same provider. That is, if another service exceeds capacity or has a burst of traffic, the customer's IETF Network Slice service should remain unaffected and there should be no noticeable change to the quality of traffic delivered.

In general, a customer cannot tell whether a service provider is meeting this SLE. They cannot tell whether the variation of an SLI is because of changes in the underlying network or because of interference from other services carried by the network. And if the service varies within the allowed bounds of the SLOs, there may be no noticeable indication that this SLE has been violated.

#### Diversity

A customer may request that traffic on the connection between one set of endpoints should use different network resources from the traffic between another set of endpoints. This might be done to enhance the availability of the IETF Network Slice service.

While availability is a measurable objective (see Section 4.1.1.1) this SLE requests a finer grade of control and is not directly measurable (although the customer might become suspicious if two connections fail at the same time).

### 4.2. IETF Network Slice Endpoints

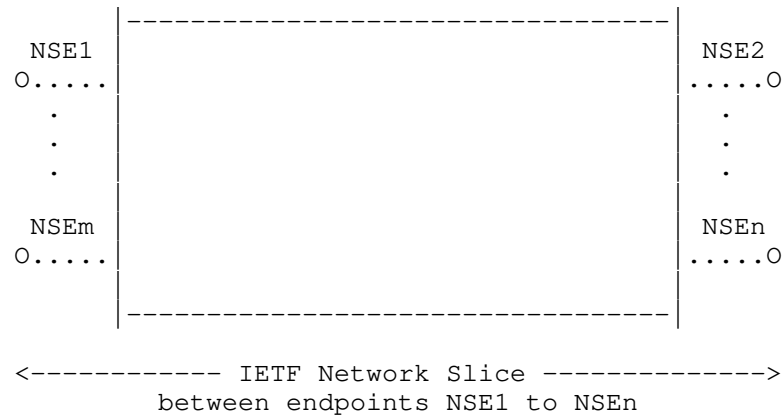
As noted in Section 3.1, an IETF Network Slice describes connectivity between multiple endpoints across the underlying network. These connectivity types are: point-to-point, point-to-multipoint, multipoint-to-point, or multipoint-to-multipoint.

Figure 1 shows an IETF Network Slice along with its Network Slice Endpoints (NSEs).

The characteristics of IETF NSEs are as follows:

- o The IETF NSEs are conceptual points of connection to IETF network slice. As such, they serve as the IETF Network Slice ingress/egress points.
- o Each endpoint could map to a device, application or a network function. A non-exhaustive list of devices, applications or network functions might include but not limited to: routers, switches, firewalls, WAN, 4G/5G RAN nodes, 4G/5G Core nodes, application acceleration, Deep Packet Inspection (DPI), server load balancers, NAT44 [RFC3022], NAT64 [RFC6146], HTTP header enrichment functions, and TCP optimizers.
- o An NSE should be identified by a unique ID in the context of an IETF Network Slice customer.
- o In addition to an identifier, each NSE should contain a subset of attributes such as IPv4/IPv6 addresses, encapsulation type (i.e., VLAN tag, MPLS Label etc.), interface/port numbers, node ID etc.
- o A combination of NSE unique ID and NSE attributes defines an NSE in the context of the IETF Network Slice Controller (NSC).
- o During the realization of the IETF Network Slice, in addition to SLOs, all or subset of IETF NSE attributes will be utilized by the IETF NSC to find the optimal realization in the IETF network.
- o Similarly to IETF Network Slices, the IETF Network Slice Endpoints are logical entities that are mapped to services/tunnels/paths endpoints in IETF Network Slice during its initialization and realization.

Note that there are various IETF TE terms such as access points (AP) defined in [RFC8453], Termination Point (TP) defined in [RFC8345], and Link Termination Point (LTP) defined in [RFC8795] which are tightly coupled with TE network type and various realization techniques. At the time of realization of the IETF Network Slice, the NSE could be mapped to one or more of these based on the network slice realization technique in use.



## Legend:

NSE: IETF Network Slice Endpoint

O: Represents IETF Network Slice Endpoints

Figure 1: An IETF Network Slice Endpoints (NSE)

## 4.2.1. IETF Network Slice Connectivity Types

The IETF Network Slice connection types can be point to point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), or multipoint-to-multipoint (MP2MP). They will be requested by the higher level operation system.

## 4.3. IETF Network Slice Decomposition

Operationally, an IETF Network Slice may be decomposed into two or more IETF Network Slices as specified below. Decomposed network slices are then independently realized and managed.

- o Hierarchical (i.e., recursive) composition: An IETF Network Slice can be further sliced into other network slices. Recursive composition allows an IETF Network Slice at one layer to be used by the other layers. This type of multi-layer vertical IETF Network Slice associates resources at different layers.
- o Sequential composition: Different IETF Network Slices can be placed into a sequence to provide an end-to-end service. In sequential composition, each IETF Network Slice would potentially support different dataplanes that need to be stitched together.

## 5. Framework

A number of IETF Network Slice services will typically be provided over a shared underlying network infrastructure. Each IETF Network Slice consists of both the overlay connectivity and a specific set of dedicated network resources and/or functions allocated in a shared underlay network to satisfy the needs of the IETF Network Slice customer. In at least some examples of underlying network technologies, the integration between the overlay and various underlay resources is needed to ensure the guaranteed performance requested for different IETF Network Slices.

### 5.1. IETF Network Slice Stakeholders

An IETF Network Slice and its realization involves the following stakeholders and it is relevant to define them for consistent terminology. The IETF Network Slice Customer and IETF network Slice provider (see Section 2.1) are also stakeholders.

**Orchestrator:** An orchestrator is an entity that composes different services, resource and network requirements. It interfaces with the IETF NSC.

**IETF Network Slice Controller (NSC):** It realizes an IETF Network Slice in the underlying network, maintains and monitors the run-time state of resources and topologies associated with it. A well-defined interface is needed between different types of IETF NSCs and different types of orchestrators. An IETF Network Slice operator (or slice operator for short) manages one or more IETF Network Slices using the IETF NSCs.

**Network Controller:** is a form of network infrastructure controller that offers network resources to the NSC to realize a particular network slice. These may be existing network controllers associated with one or more specific technologies that may be adapted to the function of realizing IETF Network Slices in a network.

### 5.2. Expressing Connectivity Intents

The NSC northbound interface (NBI) can be used to communicate between IETF Network Slice customers and the NSC.

An IETF Network Slice customer may be a network operator who, in turn, provides the IETF Network Slice to another IETF Network Slice customer.



Using the NBI, a customer expresses requirements for a particular slice by specifying what is required rather than how that is to be achieved. That is, the customer's view of a slice is an abstract one. Customers normally have limited (or no) visibility into the provider network's actual topology and resource availability information.

This should be true even if both the customer and provider are associated with a single administrative domain, in order to reduce the potential for adverse interactions between IETF Network Slice customers and other users of the underlay network infrastructure.

The benefits of this model can include:

- o Security: because the underlay network (or network operator) does not need to expose network details (topology, capacity, etc.) to IETF Network Slice customers the underlay network components are less exposed to attack;
- o Layered Implementation: the underlay network comprises network elements that belong to a different layer network than customer applications, and network information (advertisements, protocols, etc.) that a customer cannot interpret or respond to (note - a customer should not use network information not exposed via the NSC NBI, even if that information is available);
- o Scalability: customers do not need to know any information beyond that which is exposed via the NBI.

The general issues of abstraction in a TE network is described more fully in [RFC7926].

This framework document does not assume any particular layer at which IETF Network Slices operate as a number of layers (including virtual L2, Ethernet or IP connectivity) could be employed.

Data models and interfaces are of course needed to set up IETF Network Slices, and specific interfaces may have capabilities that allow creation of specific layers.

Layered virtual connections are comprehensively discussed in IETF documents and are widely supported. See, for instance, GMPLS-based networks [RFC5212] and [RFC4397], or Abstraction and Control of TE Networks (ACTN) [RFC8453] and [RFC8454]. The principles and mechanisms associated with layered networking are applicable to IETF Network Slices.

There are several IETF-defined mechanisms for expressing the need for a desired logical network. The NBI carries data either in a protocol-defined format, or in a formalism associated with a modeling language.

For instance:

- o Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440] and GMPLS User-Network Interface (UNI) using RSVP-TE [RFC4208] use a TLV-based binary encoding to transmit data.
- o Network Configuration Protocol (NETCONF) [RFC6241] and RESTCONF Protocol [RFC8040] use XML and JSON encoding.
- o gRPC/GNMI [I-D.openconfig-rtgwg-gnmi-spec] uses a binary encoded programmable interface;
- o For data modeling, YANG ([RFC6020] and [RFC7950]) may be used to model configuration and other data for NETCONF, RESTCONF, and GNMI - among others; ProtoBufs can be used to model gRPC and GNMI data.

While several generic formats and data models for specific purposes exist, it is expected that IETF Network Slice management may require enhancement or augmentation of existing data models.

### 5.3. IETF Network Slice Controller (NSC)

The IETF NSC takes abstract requests for IETF Network Slices and implements them using a suitable underlying technology. An IETF NSC is the key building block for control and management of the IETF Network Slice. It provides the creation/modification/deletion, monitoring and optimization of IETF Network Slices in a multi-domain, a multi-technology and multi-vendor environment.

The main task of the IETF NSC is to map abstract IETF Network Slice requirements to concrete technologies and establish required connectivity, and ensuring that required resources are allocated to the IETF Network Slice.

An NSC northbound interface (NBI) is needed for communicating details of a IETF Network Slice (configuration, selected policies, operational state, etc.), as well as providing information to a slice requester/customer about IETF Network Slice status and performance. The details for this NBI are not in scope for this document.

The controller provides the following functions:

- o Provides a technology-agnostic NBI for creation/modification/deletion of the IETF Network Slices. The API exposed by this NBI communicates the endpoints of the IETF network slice, IETF Network Slice SLO parameters (and possibly monitoring thresholds), applicable input selection (filtering) and various policies, and provides a way to monitor the slice.
- o Determines an abstract topology connecting the endpoints of the IETF Network Slice that meets criteria specified via the NBI. The NSC also retains information about the mapping of this abstract topology to underlying components of the IETF network slice as necessary to monitor IETF Network Slice status and performance.
- o Provides "Mapping Functions" for the realization of IETF Network Slices. In other words, it will use the mapping functions that:
  - \* map technology-agnostic NBI request to technology-specific SBIs
  - \* map filtering/selection information as necessary to entities in the underlay network.
- o Via an SBI, the controller collects telemetry data (e.g., OAM results, statistics, states, etc.) for all elements in the abstract topology used to realize the IETF Network Slice.
- o Using the telemetry data from the underlying realization of a IETF Network Slice (i.e., services/paths/tunnels), evaluates the current performance against IETF Network Slice SLO parameters and exposes them to the IETF Network Slice customer via the NBI. The NSC NBI may also include a capability to provide notification in case the IETF Network Slice performance reaches threshold values defined by the IETF Network Slice customer.

An IETF Network Slice customer is served by the IETF Network Slice Controller (NSC), as follows:

- o The NSC takes requests from a management system or other application, which are then communicated via an NBI. This interface carries data objects the IETF Network Slice customer provides, describing the needed IETF Network Slices in terms of topology, applicable service level objectives (SLO), and any monitoring and reporting requirements that may apply. Note that - in this context - "topology" means what the IETF Network Slice connectivity is meant to look like from the customer's perspective; it may be as simple as a list of mutually (and symmetrically) connected end points, or it may be complicated by details of connection asymmetry, per-connection SLO requirements, etc.

- o These requests are assumed to be translated by one or more underlying systems, which are used to establish specific IETF Network Slice instances on top of an underlying network infrastructure.
- o The NSC maintains a record of the mapping from customer requests to slice instantiations, as needed to allow for subsequent control functions (such as modification or deletion of the requested slices), and as needed for any requested monitoring and reporting functions.

#### 5.3.1. IETF Network Slice Controller Interfaces

The interworking and interoperability among the different stakeholders to provide common means of provisioning, operating and monitoring the IETF Network Slices is enabled by the following communication interfaces (see Figure 2).

**NSC Northbound Interface (NBI):** The NSC Northbound Interface is an interface between a customer's higher level operation system (e.g., a network slice orchestrator) and the NSC. It is a technology agnostic interface. The customer can use this interface to communicate the requested characteristics and other requirements (i.e., the SLOs) for the IETF Network Slice, and the NSC can use the interface to report the operational state of an IETF Network Slice to the customer.

**NSC Southbound Interface (SBI):** The NSC Southbound Interface is an interface between the NSC and network controllers. It is technology-specific and may be built around the many network models defined within the IETF.

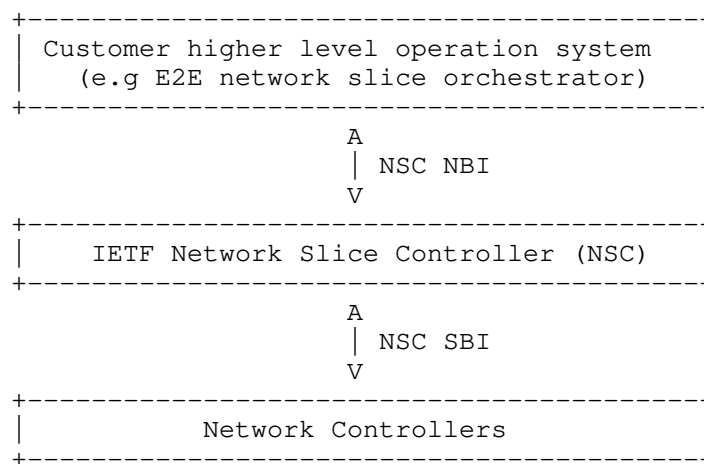


Figure 2: Interface of IETF Network Slice Controller

### 5.3.2. Northbound Interface (NBI)

The IETF Network Slice Controller provides a Northbound Interface (NBI) that allows customers of network slices to request and monitor IETF Network Slices. Customers operate on abstract IETF Network Slices, with details related to their realization hidden.

The NBI complements various IETF services, tunnels, path models by providing an abstract layer on top of these models.

The NBI is independent of type of network functions or services that need to be connected, i.e., it is independent of any specific storage, software, protocol, or platform used to realize physical or virtual network connectivity or functions in support of IETF Network Slices.

The NBI uses protocol mechanisms and information passed over those mechanisms to convey desired attributes for IETF Network Slices and their status. The information is expected to be represented as a well-defined data model, and should include at least endpoint and connectivity information, SLO specification, and status information.

To accomplish this, the NBI needs to convey information needed to support communication across the NBI, in terms of identifying the IETF Network Slices, as well providing the above model information.

#### 5.4. IETF Network Slice Structure

An IETF Network Slice is a set of connections among various endpoints to form a logical network that meets the SLOs agreed upon.

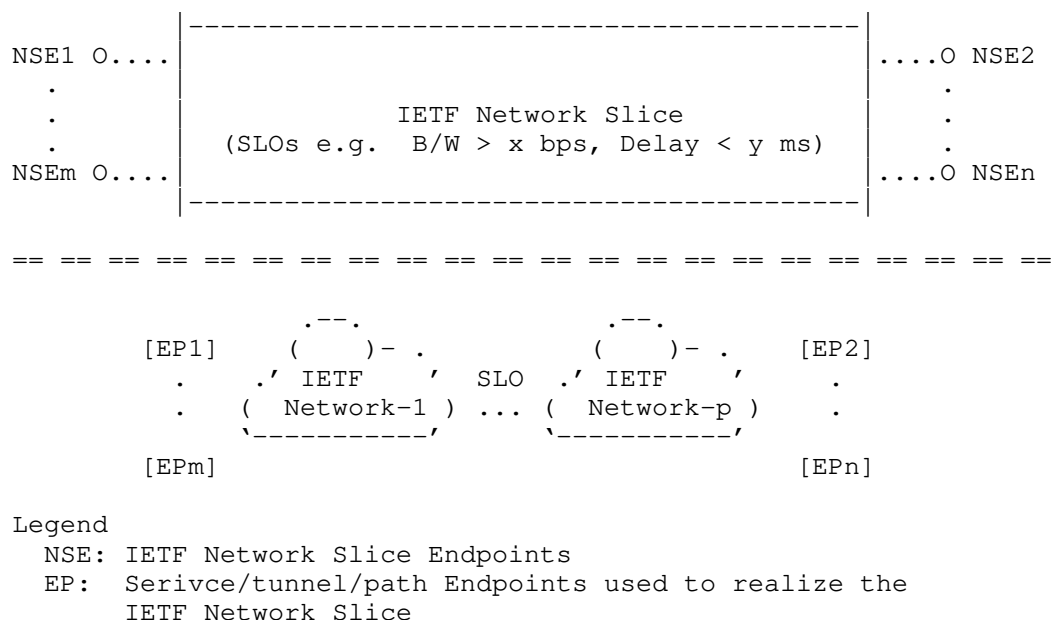


Figure 3: IETF Network Slice

Figure 3 illustrates a case where an IETF Network Slice provides connectivity between a set of IETF network slice endpoints (NSE) pairs with specific SLOs (e.g., guaranteed minimum bandwidth of  $x$  bps and guaranteed delay of no more than  $y$  ms). The IETF Network Slice endpoints are mapped to the service/tunnel/path Endpoints (EPs) in the underlay network. Also, the IETF NSEs in the same IETF network slice may belong to the same or different address spaces.

IETF Network Slice structure fits into a broader concept of end-to-end network slices. A network operator may be responsible for delivering services over a number of technologies (such as radio networks) and for providing specific and fine-grained services (such as CCTV feed or High definition realtime traffic data). That operator may need to combine slices of various networks to produce an end-to-end network service. Each of these networks may include multiple physical or virtual nodes and may also provide network functions beyond simply carrying of technology-specific protocol data

units. An end-to-end network slice is defined by the 3GPP as a complete logical network that provides a service in its entirety with a specific assurance to the customer [TS23501].

An end-to-end network slice may be composed from other network slices that include IETF Network Slices. This composition may include the hierarchical (or recursive) use of underlying network slices and the sequential (or stitched) combination of slices of different networks.

## 6. Realizing IETF Network Slices

Realization of IETF Network Slices is out of scope of this document. It is a mapping of the definition of the IETF Network Slice to the underlying infrastructure and is necessarily technology-specific and achieved by the NSC over the SBI.

The realization can be achieved in a form of either physical or logical connectivity using VPNs, virtual networks (VNs), or a variety of tunneling technologies such as Segment Routing, MPLS, etc. Accordingly, endpoints (NSEs) may be realized as physical or logical service or network functions.

### 6.1. Procedures to Realize IETF Network Slices

There are a number of different technologies that can be used in the underlay, including physical connections, MPLS, time-sensitive networking (TSN), Flex-E, etc.

An IETF Network Slice can be realized in a network, using specific underlying technology or technologies. The creation of a new IETF Network Slice will be initiated with following three steps:

- o Step 1: A higher level system requests connections with specific characteristics via the NBI.
- o Step 2: This request will be processed by an IETF NSC which specifies a mapping between northbound request to any IETF Services, Tunnels, and paths models.
- o Step 3: A series of requests for creation of services, tunnels and paths will be sent to the network to realize the transport slice.

It is very clear that, regardless of how IETF Network Slice is realized in the network (i.e., using tunnels of different types), the definition of the IETF Network Slice does not change at all. The only difference is how the slice is realized. The following sections briefly introduce some existing architectural approaches that can be applied to realize IETF Network Slices.

## 6.2. Applicability of ACTN to IETF Network Slices

Abstraction and Control of TE Networks (ACTN - [RFC8453]) is a management architecture and toolkit used to create virtual networks (VNs) on top of a traffic engineering (TE) underlay network. The VNs can be presented to customers for them to operate as private networks.

In many ways, the function of ACTN is similar to IETF network slicing. Customer requests for connectivity-based overlay services are mapped to dedicated or shared resources in the underlay network in a way that meets customer guarantees for service level objectives and for separation from other customers' traffic. [RFC8453] the function of ACTN as collecting resources to establish a logically dedicated virtual network over one or more TE networks. Thus, in the case of a TE-enabled underlying network, the ACTN VN can be used as a basis to realize an IETF network slicing.

While the ACTN framework is a generic VN framework that can be used for VN services beyond the IETF network slice, it also a suitable basis for delivering and realizing IETF network slices.

Further discussion of the applicability of ACTN to IETF network slices including a discussion of the relevant YANG models can be found in [I-D.king-teas-applicability-actn-slicing].

## 6.3. Applicability of Enhanced VPNs to IETF Network Slices

An enhanced VPN (VPN+) is designed to support the needs of new applications, particularly applications that are associated with 5G services, by utilizing an approach that is based on existing VPN and Traffic Engineering (TE) technologies and adds characteristics that specific services require over and above traditional VPNs.

An enhanced VPN can be used to provide enhanced connectivity services between customer sites (a concept similar to an IETF Network Slice) and can be used to create the infrastructure to underpin network slicing.

It is envisaged that enhanced VPNs will be delivered using a combination of existing, modified, and new networking technologies.

[I-D.ietf-teas-enhanced-vpn] describes the framework for Enhanced Virtual Private Network (VPN+) services.



#### 6.4. Network Slicing and Slice Aggregation in IP/MPLS Networks

Network slicing provides the ability to partition a physical network into multiple isolated logical networks of varying sizes, structures, and functions so that each slice can be dedicated to specific services or customers.

Many approaches are currently being worked on to support IETF Network Slices in IP and MPLS networks with or without the use of Segment Routing. Most of these approaches utilize a way of marking packets so that network nodes can apply specific routing and forwarding behaviors to packets that belong to different IETF Network Slices. Different mechanisms for marking packets have been proposed (including using MPLS labels and Segment Routing segment IDs) and those mechanisms are agnostic to the path control technology used within the underlay network.

These approaches are also sensitive to the scaling concerns of supporting a large number of IETF Network Slices within a single IP or MPLS network, and so offer ways to aggregate the slices so that the packet markings indicate an aggregate or grouping of IETF Network Slices where all of the packets are subject to the same routing and forwarding behavior.

At this stage, it is inappropriate to mention any of these proposed solutions that are currently work in progress and not yet adopted as IETF work.

### 7. Isolation in IETF Network Slices

#### 7.1. Isolation as a Service Requirement

An IETF network slice customer may request that the IETF network slice delivered to them is delivered such that changes to other IETF network slices or services do not have any negative impact on the delivery of the IETF Network Slice. The IETF Network Slice customer may specify the degree to which their IETF Network Slice is unaffected by changes in the provider network or by the behavior of other IETF Network Slice customers. The customer may express this via an SLE it agrees with the provider. This concept is termed 'isolation'

#### 7.2. Isolation in IETF Network Slice Realization

Isolation may be achieved in the underlying network by various forms of resource partitioning ranging from dedicated allocation of resources for a specific IETF Network Slice, to sharing of resources with safeguards. For example, traffic separation between different

IETF Network Slices may be achieved using VPN technologies, such as L3VPN, L2VPN, EVPN, etc. Interference avoidance may be achieved by network capacity planning, allocating dedicated network resources, traffic policing or shaping, prioritizing in using shared network resources, etc. Finally, service continuity may be ensured by reserving backup paths for critical traffic, dedicating specific network resources for a selected number of IETF Network Slices.

## 8. Management Considerations

IETF Network Slice realization needs to be instrumented in order to track how it is working, and it might be necessary to modify the IETF Network Slice as requirements change. Dynamic reconfiguration might be needed.

## 9. Security Considerations

This document specifies terminology and has no direct effect on the security of implementations or deployments. In this section, a few of the security aspects are identified.

- o Conformance to security constraints: Specific security requests from customer defined IETF Network Slices will be mapped to their realization in the underlay networks. It will be required by underlay networks to have capabilities to conform to customer's requests as some aspects of security may be expressed in SLEs.
- o IETF NSC authentication: Underlying networks need to be protected against the attacks from an adversary NSC as they can destabilize overall network operations. It is particularly critical since an IETF Network Slice may span across different networks, therefore, IETF NSC should have strong authentication with each those networks. Furthermore, both SBI and NBI need to be secured.
- o Specific isolation criteria: The nature of conformance to isolation requests means that it should not be possible to attack an IETF Network Slice service by varying the traffic on other services or slices carried by the same underlay network. In general, isolation is expected to strengthen the IETF Network Slice security.
- o Data Integrity of an IETF Network Slice: A customer wanting to secure their data and keep it private will be responsible for applying appropriate security measures to their traffic and not depending on the network operator that provides the IETF Network Slice. It is expected that for data integrity, a customer is responsible for end-to-end encryption of its own traffic.

Note: see NGMN document[NGMN\_SEC] on 5G network slice security for discussion relevant to this section.

IETF Network Slices might use underlying virtualized networking. All types of virtual networking require special consideration to be given to the separation of traffic between distinct virtual networks, as well as some degree of protection from effects of traffic use of underlying network (and other) resources from other virtual networks sharing those resources.

For example, if a service requires a specific upper bound of latency, then that service can be degraded by added delay in transmission of service packets through the activities of another service or application using the same resources.

Similarly, in a network with virtual functions, noticeably impeding access to a function used by another IETF Network Slice (for instance, compute resources) can be just as service degrading as delaying physical transmission of associated packet in the network.

While a IETF Network Slice might include encryption and other security features as part of the service, customers might be well advised to take responsibility for their own security needs, possibly by encrypting traffic before hand-off to a service provider.

## 10. Privacy Considerations

Privacy of IETF Network Slice service customers must be preserved. It should not be possible for one IETF Network Slice customer to discover the presence of other customers, nor should sites that are members of one IETF Network Slice be visible outside the context of that IETF Network Slice.

In this sense, it is of paramount importance that the system use the privacy protection mechanism defined for the specific underlying technologies used, including in particular those mechanisms designed to preclude acquiring identifying information associated with any IETF Network Slice customer.

## 11. IANA Considerations

This document makes no requests for IANA action.

## 12. Informative References

- [BBF-SD406] Broadband Forum, "End-to-end network slicing", BBF SD-406, <<https://wiki.broadband-forum.org/display/BBF/SD-406+End-to-End+Network+Slicing>>.
- [HIPAA] HHS, "Health Insurance Portability and Accountability Act - The Security Rule", February 2003, <<https://www.hhs.gov/hipaa/for-professionals/security/index.html>>.
- [I-D.ietf-teas-enhanced-vpn] Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", draft-ietf-teas-enhanced-vpn-07 (work in progress), February 2021.
- [I-D.king-teas-applicability-actn-slicing] King, D., Drake, J., Zheng, H., and A. Farrel, "Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing", draft-king-teas-applicability-actn-slicing-10 (work in progress), March 2021.
- [I-D.openconfig-rtgwg-gnmi-spec] Shakir, R., Shaikh, A., Borman, P., Hines, M., Lebsack, C., and C. Morrow, "gRPC Network Management Interface (gNMI)", draft-openconfig-rtgwg-gnmi-spec-01 (work in progress), March 2018.
- [MACsec] IEEE, "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Security", 2018, <<https://1.ieee802.org/security/802-1ae>>.
- [NGMN-NS-Concept] NGMN Alliance, "Description of Network Slicing Concept", [https://www.ngmn.org/uploads/media/161010\\_NGMN\\_Network\\_Slicing\\_framework\\_v1.0.8.pdf](https://www.ngmn.org/uploads/media/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf) , 2016.
- [NGMN\_SEC] NGMN Alliance, "NGMN 5G Security - Network Slicing", April 2016, <[https://www.ngmn.org/wp-content/uploads/Publication/s/2016/160429\\_NGMN\\_5G\\_Security\\_Network\\_Slicing\\_v1\\_0.pdf](https://www.ngmn.org/wp-content/uploads/Publication/s/2016/160429_NGMN_5G_Security_Network_Slicing_v1_0.pdf)>.
- [PCI] PCI Security Standards Council, "PCI DSS", May 2018, <<https://www.pcisecuritystandards.org>>.

- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <<https://www.rfc-editor.org/info/rfc2681>>.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<https://www.rfc-editor.org/info/rfc3022>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC4208] Swallow, G., Drake, J., Ishimatsu, H., and Y. Rekhter, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC 4208, DOI 10.17487/RFC4208, October 2005, <<https://www.rfc-editor.org/info/rfc4208>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4397] Bryskin, I. and A. Farrel, "A Lexicography for the Interpretation of Generalized Multiprotocol Label Switching (GMPLS) Terminology within the Context of the ITU-T's Automatically Switched Optical Network (ASON) Architecture", RFC 4397, DOI 10.17487/RFC4397, February 2006, <<https://www.rfc-editor.org/info/rfc4397>>.
- [RFC5212] Shiimoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", RFC 5212, DOI 10.17487/RFC5212, July 2008, <<https://www.rfc-editor.org/info/rfc5212>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<https://www.rfc-editor.org/info/rfc6146>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.
- [RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <<https://www.rfc-editor.org/info/rfc7926>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.

- [RFC8454] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., and B. Yoon, "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC 8454, DOI 10.17487/RFC8454, September 2018, <<https://www.rfc-editor.org/info/rfc8454>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.
- [TS23501] 3GPP, "System architecture for the 5G System (5GS)", 3GPP TS 23.501, 2019.
- [TS28530] 3GPP, "Management and orchestration; Concepts, use cases and requirements", 3GPP TS 28.530, 2019.
- [TS33.210] 3GPP, "3G security; Network Domain Security (NDS); IP network layer security (Release 14).", December 2016, <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2279>>.

#### Acknowledgments

The entire TEAS Network Slicing design team and everyone participating in related discussions has contributed to this document. Some text fragments in the document have been copied from the [I-D.ietf-teas-enhanced-vpn], for which we are grateful.

Significant contributions to this document were gratefully received from the contributing authors listed in the "Contributors" section. In addition we would like to also thank those others who have attended one or more of the design team meetings, including the following people not listed elsewhere:

- o Aihua Guo
- o Bo Wu
- o Greg Mirsky
- o Lou Berger
- o Rakesh Gandhi
- o Ran Chen

- o Sergio Belotti
- o Stewart Bryant
- o Tomonobu Niwa
- o Xuesong Geng

Further useful comments were received from Daniele Ceccarelli, Uma Chunduri, Pavan Beeram, Tarek Saad, Med Boucadair, Kenichi Okagi, Oscar Gonzalez de Dios, and Xiaobing Niu.

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).

#### Contributors

The following authors contributed significantly to this document:

Jari Arkko  
Ericsson  
Email: jari.arkko@piuha.net

Dhruv Dhody  
Huawei, India  
Email: dhruv.ietf@gmail.com

Jie Dong  
Huawei  
Email: jie.dong@huawei.com

Xufeng Liu  
Volta Networks  
Email: xufeng.liu.ietf@gmail.com

#### Authors' Addresses

Adrian Farrel (editor)  
Old Dog Consulting  
UK

Email: adrian@olddog.co.uk



Eric Gray  
Independent  
USA

Email: ewgray@graiymage.com

John Drake  
Juniper Networks  
USA

Email: jdrake@juniper.net

Reza Rokui  
Nokia

Email: reza.rokui@nokia.com

Shunsuke Homma  
NTT  
Japan

Email: shunsuke.homma.ietf@gmail.com

Kiran Makhijani  
Futurewei  
USA

Email: kiranm@futurewei.com

Luis M. Contreras  
Telefonica  
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Jeff Tantsura  
Juniper Networks

Email: jefftant.ietf@gmail.com



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 28 September 2022

A. Farrel, Ed.  
Old Dog Consulting  
J. Drake, Ed.  
Juniper Networks  
R. Rokui  
Ciena  
S. Homma  
NTT  
K. Makhijani  
Futurewei  
L.M. Contreras  
Telefonica  
J. Tantsura  
Microsoft  
27 March 2022

Framework for IETF Network Slices  
draft-ietf-teas-ietf-network-slices-10

Abstract

This document describes network slicing in the context of networks built from IETF technologies. It defines the term "IETF Network Slice" and establishes the general principles of network slicing in the IETF context.

The document discusses the general framework for requesting and operating IETF Network Slices, the characteristics of an IETF Network Slice, the necessary system components and interfaces, and how abstract requests can be mapped to more specific technologies. The document also discusses related considerations with monitoring and security.

This document also provides definitions of related terms to enable consistent usage in other IETF documents that describe or use aspects of IETF Network Slices.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 28 September 2022.

#### Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

#### Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 3  |
| 1.1. Background . . . . .  | 4  |
| 2. Terms and Abbreviations . . . . .                                 | 5  |
| 2.1. Core Terminology . . . . .                                      | 6  |
| 3. IETF Network Slice . . . . .                                      | 7  |
| 3.1. Definition and Scope of IETF Network Slice . . . . .            | 8  |
| 3.2. IETF Network Slice Service . . . . .                            | 8  |
| 3.2.1. Ancillary SDPs . . . . .                                      | 12 |
| 4. IETF Network Slice System Characteristics . . . . .               | 12 |
| 4.1. Objectives for IETF Network Slices . . . . .                    | 12 |
| 4.1.1. Service Level Objectives . . . . .                            | 13 |
| 4.1.2. Service Level Expectations . . . . .                          | 15 |
| 4.2. IETF Network Slice Service Demarcation Points . . . . .         | 17 |
| 4.3. IETF Network Slice Composition . . . . .                        | 19 |
| 5. Framework . . . . .   | 20 |
| 5.1. IETF Network Slice Stakeholders . . . . .                       | 20 |
| 5.2. Expressing Connectivity Intents . . . . .                       | 21 |
| 5.3. IETF Network Slice Controller (NSC) . . . . .                   | 22 |
| 5.3.1. IETF Network Slice Controller Interfaces . . . . .            | 24 |
| 5.3.2. Management Architecture . . . . .                             | 25 |
| 6. Realizing IETF Network Slices . . . . .                           | 26 |
| 6.1. Architecture to Realize IETF Network Slices . . . . .           | 27 |
| 6.2. Procedures to Realize IETF Network Slices . . . . .             | 30 |
| 6.3. Applicability of ACTN to IETF Network Slices . . . . .          | 31 |
| 6.4. Applicability of Enhanced VPNs to IETF Network Slices . . . . . | 31 |

|  |    |
|--|----|
| 6.5. Network Slicing and Aggregation in IP/MPLS Networks . . . | 32 |
| 6.6. Network Slicing and Service Function Chaining (SFC) . . . | 32 |
| 7. Isolation in IETF Network Slices . . . . .                  | 33 |
| 7.1. Isolation as a Service Requirement . . . . .              | 33 |
| 7.2. Isolation in IETF Network Slice Realization . . . . .     | 34 |
| 8. Management Considerations . . . . .                         | 34 |
| 9. Security Considerations . . . . .                           | 34 |
| 10. Privacy Considerations . . . . .                           | 35 |
| 11. IANA Considerations . . . . .                              | 36 |
| 12. Informative References . . . . .                           | 36 |
| Acknowledgments . . . . .                                      | 40 |
| Contributors . . . . .   | 41 |
| Authors' Addresses . . . . .                                   | 41 |

## 1. Introduction

A number of use cases benefit from network connections that, along with connectivity, provide assurance of meeting a specific set of objectives with respect to network resources use. This connectivity and resource commitment is referred to as a network slice and is expressed in terms of connectivity constructs (see Section 3) and service objectives (see Section 4). Since the term network slice is rather generic, the qualifying term "IETF" is used in this document to limit the scope of network slice to network technologies described and standardized by the IETF. This document defines the concept of IETF Network Slices that provide connectivity coupled with a set of specific commitments of network resources between a number of endpoints (known as Service Demarcation Points (SDPs) - see Section 2.1 and Section 4.2) over a shared underlay network. The term IETF Network Slice service is also introduced to describe the service requested by and provided to the service provider's customer.

Services that might benefit from IETF Network Slices include, but are not limited to:

- \* 5G services (e.g. eMBB, URLLC, mMTC) (See [TS23501])
- \* Network wholesale services
- \* Network infrastructure sharing among operators
- \* NFV connectivity and Data Center Interconnect

IETF Network Slices are created and managed within the scope of one or more network technologies (e.g., IP, MPLS, optical). They are intended to enable a diverse set of applications with different requirements to coexist over a shared underlay network. A request for an IETF Network Slice service is agnostic to the technology in

the underlay network so as to allow a customer to describe their network connectivity objectives in a common format, independent of the underlay technologies used.

This document also provides a framework for discussing IETF Network Slices. The framework is intended as a structure for discussing interfaces and technologies. It is not intended to specify a new set of concrete interfaces or technologies.

For example, virtual private networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated access to a common network. The common or base network that is used to support the VPNs is often referred to as an underlay network, and the VPN is often called an overlay network. An overlay network may, in turn, serve as an underlay network to support another overlay network.

Note that it is conceivable that extensions to IETF technologies are needed in order to fully support all the ideas that can be implemented with network slices. Evaluation of existing technologies, proposed extensions to existing protocols and interfaces, and the creation of new protocols or interfaces is outside the scope of this document.

### 1.1. Background

The concept of network slicing has gained traction driven largely by needs surfacing from 5G ([NGMN-NS-Concept], [TS23501], and [TS28530]). In [TS23501], a Network Slice is defined as "a logical network that provides specific network capabilities and network characteristics", and a Network Slice Instance is defined as "A set of Network Function instances and the required resources (e.g. compute, storage and networking resources) which form a deployed Network Slice." According to [TS28530], an end-to-end network slice consists of three major types of network segments: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN). An IETF Network Slice provides the required connectivity between different entities in RAN and CN segments of an end-to-end network slice, with a specific performance commitment (for example, serving as a TN slice). For each end-to-end network slice, the topology and performance requirement on a customer's use of an IETF Network Slice can be very different, which requires the underlay network to have the capability of supporting multiple different IETF Network Slices.

While network slices are commonly discussed in the context of 5G, it is important to note that IETF Network Slices are a narrower concept with a broader usage profile, and focus primarily on particular network connectivity aspects. Other systems, including 5G

deployments, may use IETF Network Slices as a component to create entire systems and concatenated constructs that match their needs, including end-to-end connectivity.

An IETF Network Slice could span multiple technologies and multiple administrative domains. Depending on the IETF Network Slice customer's requirements, an IETF Network Slice could be isolated from other, often concurrent IETF Network Slices in terms of data, control and management planes.

The customer expresses requirements for a particular IETF Network Slice service by specifying what is required rather than how the requirement is to be fulfilled. That is, the IETF Network Slice customer's view of an IETF Network Slice is an abstract one.

Thus, there is a need to create logical network structures with required characteristics. The customer of such a logical network can require a degree of isolation and performance that previously might not have been satisfied by overlay VPNs. Additionally, the IETF Network Slice customer might ask for some level of control of their virtual networks, e.g., to customize the service paths in a network slice.

This document specifies definitions and a framework for the provision of an IETF Network Slice service. Section 6 briefly indicates some candidate technologies for realizing IETF Network Slices.

## 2. Terms and Abbreviations

The following abbreviations are used in this document.

- \* NSC: Network Slice Controller
- \* SDP: Service Demarcation Point
- \* SLA: Service Level Agreement
- \* SLE: Service Level Expectation
- \* SLI: Service Level Indicator
- \* SLO: Service Level Objective

The meaning of these abbreviations is defined in greater details in the remainder of this document.

## 2.1. Core Terminology

The following terms are presented here to give context. Other terminology is defined in the remainder of this document.

**Customer:** A customer is the requester of an IETF Network Slice service. Customers may request monitoring of SLOs. A customer may be an entity such as an enterprise network or a network operator, an individual working at such an entity, a private individual contracting for a service, or an application or software component. A customer may be an external party (classically a paying customer) or a division of a network operator that uses the service provided by another division of the same operator. Other terms that have been applied to the customer role are "client" and "consumer".

**Provider:** A provider is the organization that delivers an IETF Network Slice service. A provider is the network operator that controls the network resources used to construct the network slice (that is, the network that is sliced). The provider's network maybe a physical network or may be a virtual network supplied by another service provider.

**Customer Edge (CE):** The customer device that provides connectivity to a service provider. Examples include routers, Ethernet switches, firewalls, 4G/5G RAN or Core nodes, application accelerators, server load balancers, HTTP header enrichment functions, and PEPs (Performance Enhancing Proxy). In some circumstances CEs are provided to the customer and managed by the provider.

**Provider Edge (PE):** The device within the provider network to which a CE is attached. A CE may be attached to multiple PEs, and multiple CEs may be attached to a given PE.

**Attachment Circuit (AC):** A channel connecting a CE and a PE over which packets that belong to an IETF Network Slice service are exchanged. An AC is, by definition, technology specific: that is, the AC defines how customer traffic is presented to the provider network. The customer and provider agree (through configuration) on which values in which combination of layer 2 and layer 3 header and payload fields within a packet identify to which {IETF Network Slice service, connectivity construct, and SLOs/SLEs} that packet is assigned. The customer and provider may agree on a per {IETF Network Slice service, connectivity construct, and SLOs/SLEs} basis to police or shape traffic on the AC in both the ingress (CE to PE) direction and egress (PE to CE) direction, This ensures that the traffic is within the capacity profile that is agreed in



an IETF Network Slice service. Excess traffic is dropped by default, unless specific out-of-profile policies are agreed between the customer and the provider. As described in Section 4.2 the AC may be part of the IETF Network Slice service or may be external to it.

**Service Demarcation Point (SDP):** The point at which an IETF Network Slice service is delivered by a service provider to a customer. Depending on the service delivery model (see Section 4.2) this may be a CE or a PE, and could be a device, a software component, or in the case of network functions virtualization (for example), be an abstract function supported within the provider's network. Each SDP must have a unique identifier (e.g., an IP address or MAC address) within a given IETF Network Slice service and may use the same identifier in multiple IETF Network Slice services.

An SDP may be abstracted as a Service Attachment Point (SAP) [I-D.ietf-opsawg-sap] for the purpose generalizing the concept across multiple service types and representing it in management and configuration systems.

**Connectivity Construct:** A set of SDPs together with a communication type that defines how traffic flows between the SDPs. An IETF Network Slice service is specified in terms of a set of SDPs, the associated connectivity constructs and the service objectives that the customer wishes to see fulfilled.

### 3. IETF Network Slice

IETF Network Slices are created to meet specific requirements, typically expressed as bandwidth, latency, latency variation, and other desired or required characteristics. Creation of an IETF Network Slice is initiated by a management system or other application used to specify network-related conditions for particular traffic flows in response to an actual or logical IETF Network Slice service request.

Once created, these slices can be monitored, modified, deleted, and otherwise managed.

Applications and components will be able to use these IETF Network Slices to move packets between the specified end-points of the service in accordance with specified characteristics.

A clear distinction should be made between the "IETF Network Slice service" which is the function delivered to the customer (see Section 3.2) and which is agnostic to the technologies and mechanisms used by the service provider, and the "IETF Network Slice" which is

the realization of the service in the provider's network achieved by partitioning network resources and by applying certain tools and techniques within the network (see Section 3.1 and Section 6).

### 3.1. Definition and Scope of IETF Network Slice

The term "Slice" refers to a set of characteristics and behaviors that differentiate one type of user-traffic from another within a network. An IETF Network Slice is a slice of a network that uses IETF technology. An IETF Network Slice assumes that an underlay network is capable of changing the configurations of the network devices on demand, through in-band signaling, or via controllers.

An IETF Network Slice enables connectivity between a set of Service Demarcation Points (SDPs) with specific Service Level Objectives (SLOs) and Service Level Expectations (SLEs) (see Section 4) over a common underlay network. Thus, an IETF Network Slice delivers a service to a customer by meeting connectivity resource requirements and associated network capabilities such as bandwidth, latency, jitter, and network functions with other resource behaviors such as compute and storage availability.

IETF Network Slices may be combined hierarchically, so that a network slice may itself be sliced. They may also be combined sequentially so that various different networks can each be sliced and the network slices placed into a sequence to provide an end-to-end service. This form of sequential combination is utilized in some services such as in 3GPP's 5G network [TS23501].

### 3.2. IETF Network Slice Service

A service provider delivers an IETF Network Slice service for a customer by realizing an IETF Network Slice. The IETF Network Slice service is agnostic to the technology of the underlay network, and its realization may be selected based upon multiple considerations including its service requirements and the capabilities of the underlay network. This allows an IETF Network Slice service customer to describe their network connectivity and relevant objectives in a common format, independent of the underlay technologies used.

The IETF Network Slice service is specified in terms of a set of SDPs, a set of one or more connectivity constructs between subsets of these SDPs, and a set of SLOs and SLEs (see Section 4) for each SDP sending to each connectivity construct. A communication type (point-to-point (P2P), point-to-multipoint (P2MP), or any-to-any (A2A)) is specified for each connectivity construct. That is, in a given IETF Network Slice service there may be one or more connectivity constructs of the same or different type, each connectivity construct

may be between a different subset of SDPs, for a given connectivity construct each sending SDP has its own set of SLOs and SLEs, and the SLOs and SLEs in each set may be different. Note that a service provider may decide how many connectivity constructs per IETF Network Slice service it wishes to support such that an IETF Network Slice service may be limited to one connectivity construct or may support many.

This approach results in the following possible connectivity constructs:

- \* For a P2P connectivity construct, there is one sending SDP and one receiving SDP. This construct is like a private wire or a tunnel. All traffic injected at the sending SDP is intended to be received by the receiving SDP. The SLOs and SLEs apply at the sender (and implicitly at the receiver).
- \* For a P2MP connectivity construct, there is only one sending SDP and more than one receiving SDP. This is like a P2MP tunnel or multi-access VLAN segment. All traffic from the sending SDP is intended to be received by all the receiving SDPs. There is one set of SLOs and SLEs that applies at the sending SDP (and implicitly at all receiving SDPs).
- \* With an A2A connectivity construct, any sending SDP may send to any one receiving SDP or any set of receiving SDPs in the construct. There is an implicit level of routing in this connectivity construct that is not present in the other connectivity constructs because the provider's network must determine to which receiving SDPs to deliver each packet. This construct may be used to support P2P traffic between any pair of SDPs, or to support multicast or broadcast traffic from one SDP to a set of other SDPs. In the latter case, whether the service is delivered using multicast within the provider's network or using "ingress replication" or some other means is out of scope of the specification of the service. A service provider may choose to support A2A constructs, but to limit the traffic to unicast.

The SLOs/SLEs in an A2A connectivity construct apply to individual sending SDPs regardless of the receiving SDPs, and there is no linkage between sender and receiver in the specification of the connectivity construct. A sending SDP may be "disappointed" if the receiver is over-subscribed. If a customer wants to be more specific about different behaviors from one SDP to another SDP, they should use P2P connectivity constructs.

A customer traffic flow may be unicast or multicast, and various network realizations are possible:

- \* Unicast traffic may be mapped to a P2P connectivity construct for direct delivery, or to an A2A connectivity construct for the service provider to perform routing to the destination SDP. It would be unusual to use a P2MP connectivity construct to deliver unicast traffic because all receiving SDPs would get a copy, but this can still be done if the receivers are capable of dropping the unwanted traffic.
- \* A bidirectional unicast service can be constructed by specifying two P2P connectivity constructs. An additional SLE may specify fate-sharing in this case.
- \* Multicast traffic may be mapped to a set of P2P connectivity constructs, a single P2MP connectivity construct, or a mixture of P2P and P2MP connectivity constructs. Multicast may also be supported by an A2A connectivity construct. The choice clearly influences how and where traffic is replicated in the network. With a P2MP or A2A connectivity construct, it is the operator's choice whether to realize the construct with ingress replication, multicast in the core, P2MP tunnels, or hub-and-spoke. This choice should not change how the customer perceives the service.
- \* The concept of a multipoint-to-point (MP2P) service can be realized with multiple P2P connectivity constructs. Note that, in this case, the egress may simultaneously receive traffic from all ingresses. The SLOs at the sending SDPs must be set with this in mind because the provider's network is not capable of coordinating the policing of traffic across multiple distinct source SDPs. It is assumed that the customer, requesting SLOs for the various P2P connectivity constructs, is aware of the capabilities of the receiving SDP. If the receiver receives more traffic than it can handle, it may drop some and introduce queuing delays.
- \* The concept of a multipoint-to-multipoint (MP2MP) service can best be realized using a set of P2MP connectivity constructs, but could be delivered over an A2A connectivity construct if each sender is using multicast. As with MP2P, the customer is assumed to be familiar with the capabilities of all receivers. A customer may wish to achieve an MP2MP service using a hub-and-spoke architecture where they control the hub: that is, the hub may be an SDP or an ancillary SDP (see Section 3.2.1) and the service may be achieved by using a set of P2P connectivity constructs to the hub, and a single P2MP connectivity construct from the hub.

From the above, it can be seen that the SLOs of the senders define the SLOs for the receivers on any connectivity construct. That is, and in particular, the network may be expected to handle the traffic volume from a sender to all destinations. This extends to all connectivity constructs in an IETF Network Slice service.

Note that the realization of an IETF Network Slice service does not need to map the connectivity constructs one-to-one onto underlying network constructs (such as tunnels, etc.). The service provided to the customer is distinct from how the provider decides to deliver that service.

If a CE has multiple attachment circuits to a PE within a given IETF Network Slice service and they are operating in single-active mode, then all traffic between the CE and its attached PEs transits a single attachment circuit; if they are operating in all-active mode, then traffic between the CE and its attached PEs is distributed across all of the active attachment circuits.

A given sending SDP may be part of multiple connectivity constructs within a single IETF Network Slice service, and the SDP may have different SLOs and SLEs for each connectivity construct to which it is sending. Note that a given sending SDP's SLOs and SLEs for a given connectivity construct apply between it and each of the receiving SDPs for that connectivity construct.

An IETF Network Slice service provider may freely make a deployment choice as to whether to offer a 1:1 relationship between IETF Network Slice service and connectivity construct, or to support multiple connectivity constructs in a single IETF Network Slice service. In the former case, the provider might need to deliver multiple IETF Network Slice services to achieve the function of the second case.

It should be noted that per Section 9 of [RFC4364] an IETF Network Slice service customer may actually provide IETF Network Slice services to other customers in a mode sometimes referred to as "carrier's carrier". In this case, the underlying IETF Network Slice service provider may be owned and operated by the same or a different provider network. As noted in Section 4.3, network slices may be composed hierarchically or serially.

Section 4.2 provides a description of endpoints in the context of IETF network slicing. These are known as Service Demarcation Points (SDPs). For a given IETF Network Slice service, the customer and provider agree, on a per-SDP basis which end of the attachment circuit provides the SDP (i.e., whether the attachment circuit is inside or outside the IETF Network Slice service). This determines whether the attachment circuit is subject to the set of SLOs and SLEs at the specific SDP.

#### 3.2.1. Ancillary SDPs

It may be the case that the set of SDPs needs to be supplemented with additional senders or receivers. An additional sender could be, for example, an IPTV or DNS server either within the provider's network or attached to it, while an extra receiver could be, for example, a node reachable via the Internet. This is modelled as a set of ancillary SDPs which supplement the other SDPs in one or more connectivity constructs, or which have their own connectivity constructs. Note that an ancillary SDP can either have a resolvable address, e.g., an IP address or MAC address, or the SDP may be a placeholder, e.g., IPTV or DNS server, which is resolved within the provider's network when the IETF Network Slice service is instantiated.

### 4. IETF Network Slice System Characteristics

The following subsections describe the characteristics of IETF Network Slices in addition to the list of SDPs, the connectivity constructs, and the technology of the ACs.

#### 4.1. Objectives for IETF Network Slices

An IETF Network Slice service is defined in terms of quantifiable characteristics known as Service Level Objectives (SLOs) and unquantifiable characteristics known as Service Level Expectations (SLEs). SLOs are expressed in terms Service Level Indicators (SLIs), and together with the SLEs form the contractual agreement between service customer and service provider known as a Service Level Agreement (SLA).

The terms are defined as follows:

- \* A Service Level Indicator (SLI) is a quantifiable measure of an aspect of the performance of a network. For example, it may be a measure of throughput in bits per second, or it may be a measure of latency in milliseconds.

- \* A Service Level Objective (SLO) is a target value or range for the measurements returned by observation of an SLI. For example, an SLO may be expressed as "SLI <= target", or "lower bound <= SLI <= upper bound". A customer can determine whether the provider is meeting the SLOs by performing measurements on the traffic.
- \* A Service Level Expectation (SLE) is an expression of an unmeasurable service-related request that a customer of an IETF Network Slice makes of the provider. An SLE is distinct from an SLO because the customer may have little or no way of determining whether the SLE is being met, but they still contract with the provider for a service that meets the expectation.
- \* A Service Level Agreement (SLA) is an explicit or implicit contract between the customer of an IETF Network Slice service and the provider of the slice. The SLA is expressed in terms of a set of SLOs and SLEs that are to be applied for a given connectivity construct between a sending SDP and the set of receiving SDPs, and may describe the extent to which divergence from individual SLOs and SLEs can be tolerated, and commercial terms as well as any consequences for violating these SLOs and SLEs.

#### 4.1.1. Service Level Objectives

SLOs define a set of measurable network attributes and characteristics that describe an IETF Network Slice service. SLOs do not describe how an IETF Network Slice service is implemented or realized in the underlying network layers. Instead, they are defined in terms of dimensions of operation (time, capacity, etc.), availability, and other attributes.

An IETF Network Slice service may include multiple connectivity constructs that associate sets of endpoints (SDPs). SLOs apply to a given connectivity construct and apply to a specific direction of traffic flow. That is, they apply to a specific sending SDP and the connection to the specific set of receiving SDPs.

The SLOs are combined with Service Level Expectations in an SLA.

##### 4.1.1.1. Some Common SLOs

SLOs can be described as 'Directly Measurable Objectives': they are always measurable. See Section 4.1.2 for the description of Service Level Expectations which are unmeasurable service-related requests sometimes known as 'Indirectly Measurable Objectives'.

Objectives such as guaranteed minimum bandwidth, guaranteed maximum latency, maximum permissible delay variation, maximum permissible packet loss rate, and availability are 'Directly Measurable Objectives'. Future specifications (such as IETF Network Slice service YANG models) may precisely define these SLOs, and other SLOs may be introduced as described in Section 4.1.1.2.

The definition of these objectives are as follows:

**Guaranteed Minimum Bandwidth:** Minimum guaranteed bandwidth between two endpoints at any time. The bandwidth is measured in data rate units of bits per second and is measured unidirectionally.

**Guaranteed Maximum Latency:** Upper bound of network latency when transmitting between two endpoints. The latency is measured in terms of network characteristics (excluding application-level latency). [RFC7679] discusses one-way metrics.

**Maximum Permissible Delay Variation:** Packet delay variation (PDV) as defined by [RFC3393], is the difference in the one-way delay between sequential packets in a flow. This SLO sets a maximum value PDV for packets between two endpoints.

**Maximum Permissible Packet Loss Rate:** The ratio of packets dropped to packets transmitted between two endpoints over a period of time. See [RFC7680].

**Availability:** The ratio of uptime to the sum of uptime and downtime, where uptime is the time the connectivity construct is available in accordance with all of the SLOs associated with it. Availability will often be expressed along with the time period over which the availability is measured, and specifying the maximum allowed single period of downtime.

#### 4.1.1.2. Other Service Level Objectives

Additional SLOs may be defined to provide additional description of the IETF Network Slice service that a customer requests. These would be specified in further documents.

If the IETF Network Slice service is traffic aware, other traffic specific characteristics may be valuable including MTU, traffic-type (e.g., IPv4, IPv6, Ethernet or unstructured), or a higher-level behavior to process traffic according to user-application (which may be realized using network functions).



#### 4.1.2. Service Level Expectations

SLEs define a set of network attributes and characteristics that describe an IETF Network Slice service, but which are not directly measurable by the customer (e.g. diversity, isolation, and geographical restrictions). Even though the delivery of an SLE cannot usually be determined by the customer, the SLEs form an important part of the contract between customer and provider.

Quite often, an SLE will imply some details of how an IETF Network Slice service is realized by the provider, although most aspects of the implementation in the underlying network layers remain a free choice for the provider. For example, activating unicast or multicast capabilities to deliver an IETF Network Slice service could be explicitly requested by a customer or could be left as an engineering decision for the service provider based on capabilities of the network and operational choices.

SLEs may be seen as aspirational on the part of the customer, and they are expressed as behaviors that the provider is expected to apply to the network resources used to deliver the IETF Network Slice service. Of course, over time, it is possible that mechanisms will be developed that enable a customer to verify the provision of an SLE, at which point it effectively becomes an SLO. The SLEs are combined with SLOs in an SLA.

An IETF Network Slice service may include multiple connectivity constructs that associate sets of endpoints (SDPs). SLEs apply to a given connectivity construct and apply to specific directions of traffic flow. That is, they apply to a specific sending SDP and the connection to the specific set of receiving SDPs. However, being more general in nature than SLOs, SLEs may commonly be applied to all connectivity constructs in an IETF Network Slice service.

##### 4.1.2.1. Some Common SLEs

SLEs can be described as 'Indirectly Measurable Objectives': they are not generally directly measurable by the customer.

Security, geographic restrictions, maximum occupancy level, and isolation are example SLEs as follows.

Security: A customer may request that the provider applies encryption or other security techniques to traffic flowing between SDPs of a connectivity construct within an IETF Network Slice service. For example, the customer could request that only network links that have MACsec [MACsec] enabled are used to realize the connectivity construct.

This SLE may include a request for encryption (e.g., [RFC4303]) between the two SDPs explicitly to meet the architectural recommendations in [TS33.210] or for compliance with [HIPAA] or [PCI].

Whether or not the provider has met this SLE is generally not directly observable by the customer and cannot be measured as a quantifiable metric.

Please see further discussion on security in Section 9.

**Geographic Restrictions:** A customer may request that certain geographic limits are applied to how the provider routes traffic for the IETF Network Slice service. For example, the customer may have a preference that its traffic does not pass through a particular country for political or security reasons.

Whether or not the provider has met this SLE is generally not directly observable by the customer and cannot be measured as a quantifiable metric.

**Maximal Occupancy Level:** The maximal occupancy level specifies the number of flows to be admitted and optionally a maximum number of countable resource units (e.g., IP or MAC addresses) an IETF Network Slice service can consume. Since an IETF Network Slice service may include multiple connectivity constructs, this SLE should also say whether it applies for the entire IETF Network Slice service, for group of connections, or on a per connection basis.

Again, a customer may not be able to fully determine whether this SLE is being met by the provider.

**Isolation:** As described in Section 7, a customer may request that its traffic within its IETF Network Slice service is isolated from the effects of other network services supported by the same provider. That is, if another service exceeds capacity or has a burst of traffic, the customer's IETF Network Slice service should remain unaffected and there should be no noticeable change to the quality of traffic delivered.

In general, a customer cannot tell whether a service provider is meeting this SLE. They cannot tell whether the variation of an SLI is because of changes in the underlay network or because of interference from other services carried by the network. If the service varies within the allowed bounds of the SLOs, there may be no noticeable indication that this SLE has been violated.

Diversity: A customer may request that different connectivity constructs use different underlay network resources. This might be done to enhance the availability of the connectivity constructs within an IETF Network Slice service.

While availability is a measurable objective (see Section 4.1.1.1) this SLE requests a finer grade of control and is not directly measurable (although the customer might become suspicious if two connectivity constructs fail at the same time).

#### 4.2. IETF Network Slice Service Demarcation Points

As noted in Section 3.1, an IETF Network Slice provides connectivity between sets of SDPs with specific SLOs and SLEs. Section 3.2 goes on to describe how the IETF Network Slice service is composed of a set of one or more connectivity constructs that describe connectivity between the Service Demarcation Points (SDPs) across the underlay network.

The characteristics of IETF Network Slice SDPs are as follows.

- \* SDPs are conceptual points of connection to an IETF Network Slice. As such, they serve as the IETF Network Slice ingress/egress points.
- \* Each SDP maps to a device, application, or a network function, such as (but not limited to) routers, switches, interfaces/ports, firewalls, WAN, 4G/5G RAN nodes, 4G/5G Core nodes, application accelerators, server load balancers, NAT44 [RFC3022], NAT64 [RFC6146], HTTP header enrichment functions, and Performance Enhancing Proxies (PEPs) [RFC3135].
- \* An SDP is identified by a unique identifier in the context of an IETF Network Slice customer.
- \* The provider associates each SDP with a set of provider-scope identifiers such as IP addresses, encapsulation-specific identifiers (e.g., VLAN tag, MPLS Label), interface/port numbers, node ID, etc.
- \* SDPs are mapped to endpoints of services/tunnels/paths within the IETF Network Slice during its initialization and realization.
  - A combination of the SDP identifier and SDP provider-network-scope identifiers define an SDP in the context of the Network Slice Controller (NSC) (see Section 5.3).

- The NSC will use the SDP provider-network-scope identifiers as part of the process of realizing the IETF Network Slice.

For a given IETF Network Slice service, the IETF Network Slice customer and provider agree where the endpoint (i.e., the service demarcation point) is located. This determines what resources at the edge of the network form part of the IETF Network Slice and are subject to the set of SLOs and SLEs for a specific endpoint.

Figure 1 shows different potential scopes of an IETF Network Slice that are consistent with the different SDP locations. For the purpose of this discussion and without loss of generality, the figure shows customer edge (CE) and provider edge (PE) nodes connected by attachment circuits (ACs). Notes after the figure give some explanations.

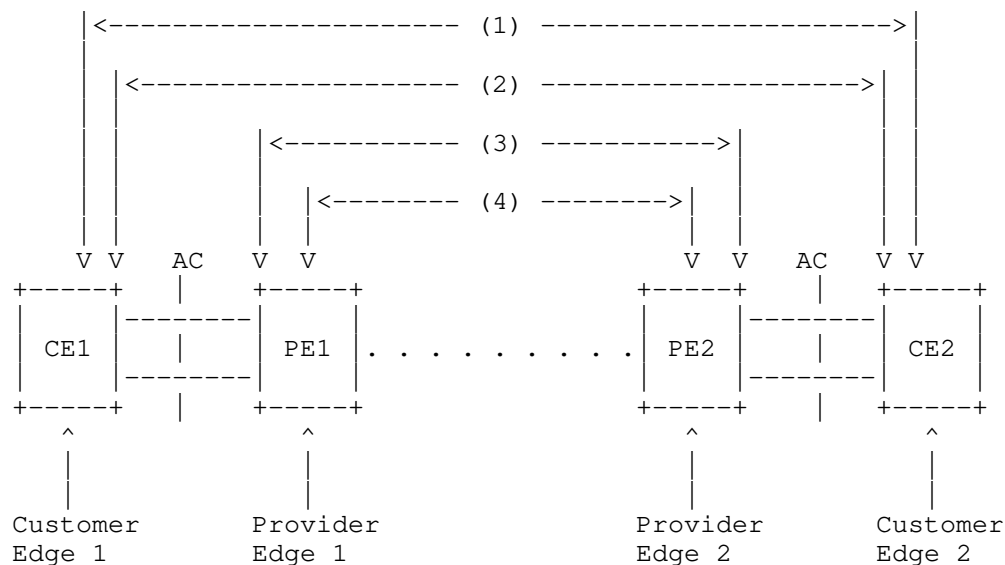


Figure 1: Positioning IETF Service Demarcation Points

Explanatory notes for Figure 1 are as follows:

1. If the CE is operated by the IETF Network Slice service provider, then the edge of the IETF Network Slice may be within the CE. In this case the slicing process may utilize resources from within the CE such as buffers and queues on the outgoing interfaces.

2. The IETF Network Slice may be extended as far as the CE, to include the AC, but not to include any part of the CE. In this case, the CE may be operated by the customer or the provider. Slicing the resources on the AC may require the use of traffic tagging (such as through Ethernet VLAN tags) or may require traffic policing at the AC link ends.
3. In another model, the SDPs of the IETF Network Slice are the customer-facing ports on the PEs. This case can be managed in a way that is similar to a port-based VPN: each port (AC) or virtual port (e.g., VLAN tag) identifies the IETF Network Slice and maps to an IETF Network Slice SDP.
4. Finally, the SDP may be within the PE. In this mode, the PE classifies the traffic coming from the AC according to information (such as the source and destination IP addresses, payload protocol and port numbers, etc.) in order to place it onto an IETF Network Slice.

The choice of which of these options to apply is entirely up to the network operator. It may limit or enable the provisioning of particular managed services and the operator will want to consider how they want to manage CEs and what control they wish to offer the customer over AC resources.

Note that Figure 1 shows a symmetrical positioning of SDPs, but this decision can be taken on a per-SDP basis through agreement between the customer and provider.

In practice, it may be necessary to map traffic not only onto an IETF Network Slice, but also onto a specific connectivity construct if the IETF Network Slice supports more than one with a source at the specific SDP. The mechanism used will be one of the mechanisms described above, dependent on how the SDP is realized.

Finally, note (as described in Section 2.1) that an SDP is an abstract endpoint of an IETF Network Slice service and as such may be a device, interface, or software component and may, in the case of network functions virtualization (for example), be an abstract function supported within the provider's network.

#### 4.3. IETF Network Slice Composition

Operationally, an IETF Network Slice may be composed of two or more IETF Network Slices as specified below. Decomposed network slices are independently realized and managed.

- \* Hierarchical (i.e., recursive) composition: An IETF Network Slice can be further sliced into other network slices. Recursive composition allows an IETF Network Slice at one layer to be used by the other layers. This type of multi-layer vertical IETF Network Slice associates resources at different layers.
- \* Sequential composition: Different IETF Network Slices can be placed into a sequence to provide an end-to-end service. In sequential composition, each IETF Network Slice would potentially support different dataplanes that need to be stitched together.

## 5. Framework

A number of IETF Network Slice services will typically be provided over a shared underlay network infrastructure. Each IETF Network Slice consists of both the overlay connectivity and a specific set of dedicated network resources and/or functions allocated in a shared underlay network to satisfy the needs of the IETF Network Slice customer. In at least some examples of underlay network technologies, the integration between the overlay and various underlay resources is needed to ensure the guaranteed performance requested for different IETF Network Slices.

### 5.1. IETF Network Slice Stakeholders

An IETF Network Slice and its realization involves the following stakeholders. The IETF Network Slice customer and IETF Network Slice provider (see Section 2.1) are also stakeholders.

**Orchestrator:** An orchestrator is an entity that composes different services, resource, and network requirements. It interfaces with the IETF NSC when composing a complex service such as an end-to-end network slice.

**IETF Network Slice Controller (NSC):** The NSC realizes an IETF Network Slice in the underlay network, and maintains and monitors the run-time state of resources and topologies associated with it. A well-defined interface is needed to support interworking between different NSC implementations and different orchestrator implementations.

**Network Controller:** The Network Controller is a form of network infrastructure controller that offers network resources to the NSC to realize a particular network slice. This may be an existing network controller associated with one or more specific technologies that may be adapted to the function of realizing IETF Network Slices in a network.

## 5.2. Expressing Connectivity Intents

An IETF Network Slice customer communicates with the NSC using the IETF Network Slice Service Interface.

An IETF Network Slice customer may be a network operator who, in turn, uses the IETF Network Slice to provide a service for another IETF Network Slice customer.

Using the IETF Network Slice Service Interface, a customer expresses requirements for a particular slice by specifying what is required rather than how that is to be achieved. That is, the customer's view of a slice is an abstract one. Customers normally have limited (or no) visibility into the provider network's actual topology and resource availability information.

This should be true even if both the customer and provider are associated with a single administrative domain, in order to reduce the potential for adverse interactions between IETF Network Slice customers and other users of the underlay network infrastructure.

The benefits of this model can include the following.

- \* **Security:** The underlay network components are less exposed to attack because the underlay network (or network operator) does not need to expose network details (topology, capacity, etc.) to the IETF Network Slice customers.
- \* **Layered Implementation:** The underlay network comprises network elements that belong to a different layer network than customer applications. Network information (advertisements, protocols, etc.) that a customer cannot interpret or respond to is not exposed to the customer. (Note - a customer should not use network information not exposed via the IETF Network Slice Service Interface, even if that information is available.)
- \* **Scalability:** Customers do not need to know any information concerning Network topology, capabilities, or state beyond that which is exposed via the IETF Network Slice Service Interface.

The general issues of abstraction in a TE network are described more fully in [RFC7926].

This framework document does not assume any particular technology layer at which IETF Network Slices operate. A number of layers (including virtual L2, Ethernet or, IP connectivity) could be employed.

Data models and interfaces are needed to set up IETF Network Slices, and specific interfaces may have capabilities that allow creation of slices within specific technology layers.

Layered virtual connections are comprehensively discussed in other IETF documents. See, for instance, GMPLS-based networks [RFC5212] and [RFC4397], or Abstraction and Control of TE Networks (ACTN) [RFC8453] and [RFC8454]. The principles and mechanisms associated with layered networking are applicable to IETF Network Slices.

There are several IETF-defined mechanisms for expressing the need for a desired logical network. The IETF Network Slice Service Interface carries data either in a protocol-defined format, or in a formalism associated with a modeling language.

For instance:

- \* The Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440] and GMPLS User-Network Interface (UNI) using RSVP-TE [RFC4208] use a TLV-based binary encoding to transmit data.
- \* The Network Configuration Protocol (NETCONF) [RFC6241] and RESTCONF Protocol [RFC8040] use XML and JSON encoding.
- \* gRPC/GNMI [I-D.openconfig-rtgwg-gnmi-spec] uses a binary encoded programmable interface. ProtoBufs can be used to model gRPC and GNMI data.
- \* For data modeling, YANG ([RFC6020] and [RFC7950]) may be used to model configuration and other data for NETCONF, RESTCONF, and GNMI, among others.

While several generic formats and data models for specific purposes exist, it is expected that IETF Network Slice management may require enhancement or augmentation of existing data models. Further, it is possible that mechanisms will be needed to determine the feasibility of service requests before they are actually made.

### 5.3. IETF Network Slice Controller (NSC)

The IETF NSC takes abstract requests for IETF Network Slices and implements them using a suitable underlay technology. An IETF NSC is the key component for control and management of the IETF Network Slice. It provides the creation/modification/deletion, monitoring and optimization of IETF Network Slices in a multi-domain, a multi-technology and multi-vendor environment.



The main task of the IETF NSC is to map abstract IETF Network Slice requirements to concrete technologies and establish required connectivity ensuring that resources are allocated to the IETF Network Slice as necessary.

The IETF Network Slice Service Interface is used for communicating details of an IETF Network Slice (configuration, selected policies, operational state, etc.), as well as information about status and performance of the IETF Network Slice. The details for this IETF Network Slice Service Interface are not in scope for this document.

The controller provides the following functions.

- \* Provides an IETF Network Slice Service Interface for creation/modification/deletion of the IETF Network Slices that is agnostic to the technology of the underlay network. The API exposed by this interface communicates the Service Demarcation Points of the IETF Network Slice, IETF Network Slice SLO/SLE parameters (and possibly monitoring thresholds), applicable input selection (filtering) and various policies, and provides a way to monitor the slice.
- \* Determines an abstract topology connecting the SDPs of the IETF Network Slice that meets criteria specified via the IETF Network Slice Service Interface. The NSC also retains information about the mapping of this abstract topology to underlay components of the IETF Network Slice as necessary to monitor IETF Network Slice status and performance.
- \* Provides "Mapping Functions" for the realization of IETF Network Slices. In other words, it will use the mapping functions that:
  - map IETF Network Slice Service Interface requests that are agnostic to the technology of the underlay network to technology-specific network configuration interfaces.
  - map filtering/selection information as necessary to entities in the underlay network so that those entities are able to identify what traffic is associated with which connectivity construct and IETF network slice and necessary according to the realization solution, and how traffic should be treated to meet the SLOs and SLEs of the connectivity construct.
- \* The controller collects telemetry data (e.g., OAM results, statistics, states, etc.) via a network configuration interface for all elements in the abstract topology used to realize the IETF Network Slice.

- \* Evaluates the current performance against IETF Network Slice SLO parameters using the telemetry data from the underlying realization of an IETF Network Slice (i.e., services/paths/tunnels). Exposes this performance to the IETF Network Slice customer via the IETF Network Slice Service Interface. The IETF Network Slice Service Interface may also include the capability to provide notifications if the IETF Network Slice performance reaches threshold values defined by the IETF Network Slice customer.

#### 5.3.1. IETF Network Slice Controller Interfaces

The interworking and interoperability among the different stakeholders to provide common means of provisioning, operating and monitoring the IETF Network Slices is enabled by the following communication interfaces (see Figure 2).

**IETF Network Slice Service Interface:** The IETF Network Slice Service Interface is an interface between a customer's higher level operation system (e.g., a network slice orchestrator or a customer network management system) and the NSC. It is agnostic to the technology of the underlay network. The customer can use this interface to communicate the requested characteristics and other requirements for the IETF Network Slice, and the NSC can use the interface to report the operational state of an IETF Network Slice to the customer.

**Network Configuration Interface:** The Network Configuration Interface is an interface between the NSC and network controllers. It is technology-specific and may be built around the many network models already defined within the IETF.

These interfaces can be considered in the context of the Service Model and Network Model described in [RFC8309] and, together with the Device Configuration Interface used by the Network Controllers, provides a consistent view of service delivery and realization.

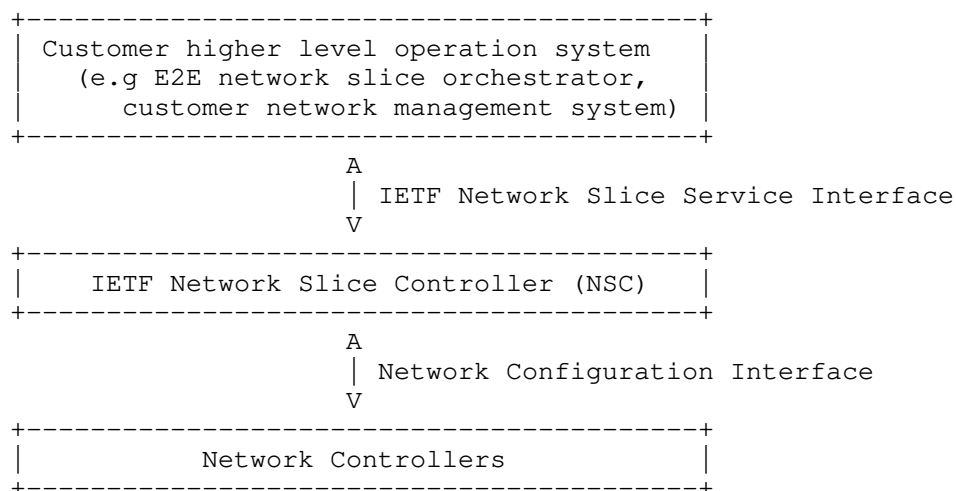


Figure 2: Interfaces of the IETF Network Slice Controller

#### 5.3.1.1. IETF Network Slice Service Interface

The IETF Network Slice Controller provides an IETF Network Slice Service Interface that allows customers to request and monitor IETF Network Slices. Customers operate on abstract IETF Network Slices, with details related to their realization hidden.

The IETF Network Slice Service Interface is also independent of the type of network functions or services that need to be connected, i.e., it is independent of any specific storage, software, protocol, or platform used to realize physical or virtual network connectivity or functions in support of IETF Network Slices.

The IETF Network Slice Service Interface uses protocol mechanisms and information passed over those mechanisms to convey desired attributes for IETF Network Slices and their status. The information is expected to be represented as a well-defined data model, and should include at least SDP and connectivity information, SLO/SLE specification, and status information.

#### 5.3.2. Management Architecture

The management architecture described in Figure 2 may be further decomposed as shown in Figure 3. This should also be seen in the context of the component architecture shown in Figure 4 and corresponds to the architecture in [RFC8309].

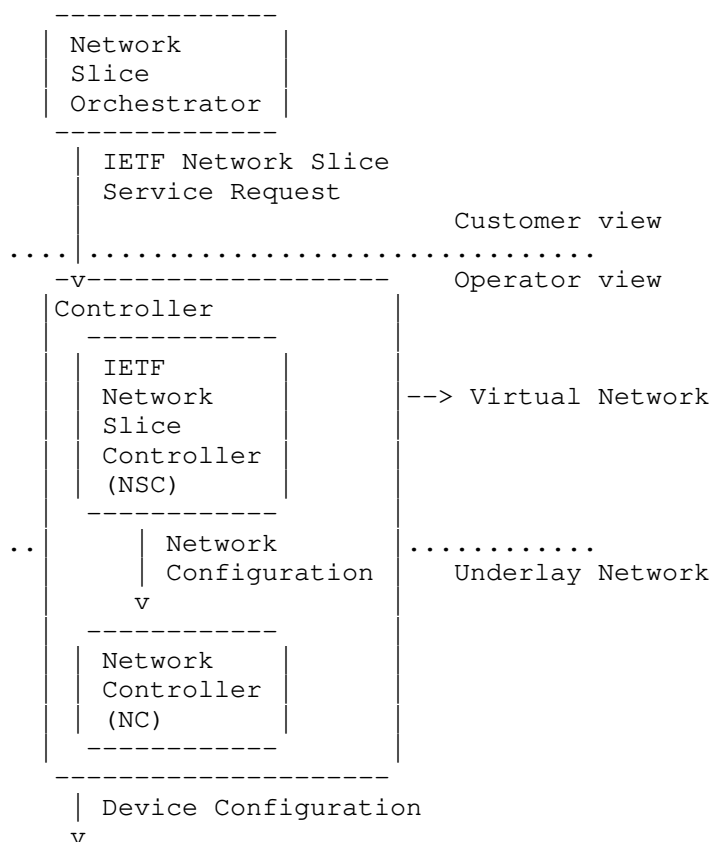


Figure 3: Interface of IETF Network Slice Management Architecture

## 6. Realizing IETF Network Slices

Realization of IETF Network Slices is out of scope of this document. It is a mapping of the definition of the IETF Network Slice to the underlying infrastructure and is necessarily technology-specific and achieved by the NSC over the Network Configuration Interface. However, this section provides an overview of the components and processes involved in realizing an IETF Network Slice.

The realization can be achieved in a form of either physical or logical connectivity using VPNs, virtual networks (VNs), or a variety of tunneling technologies such as Segment Routing, MPLS, etc. Accordingly, SDPs may be realized as physical or logical service or network functions.

### 6.1. Architecture to Realize IETF Network Slices

The architecture described in this section is deliberately at a high level. It is not intended to be prescriptive: implementations and technical solutions may vary freely. However, this approach provides a common framework that other documents may reference in order to facilitate a shared understanding of the work.

Figure 4 shows the architectural components of a network managed to provide IETF Network Slices. The customer's view is of individual IETF Network Slices with their SDPs, and connectivity constructs. Requests for IETF Network Slices are delivered to the NSC.

The figure shows, without loss of generality, the CEs, ACs, and PEs, that exist in the network. The SDPs are not shown and can be placed in any of the ways described in Section 4.2.

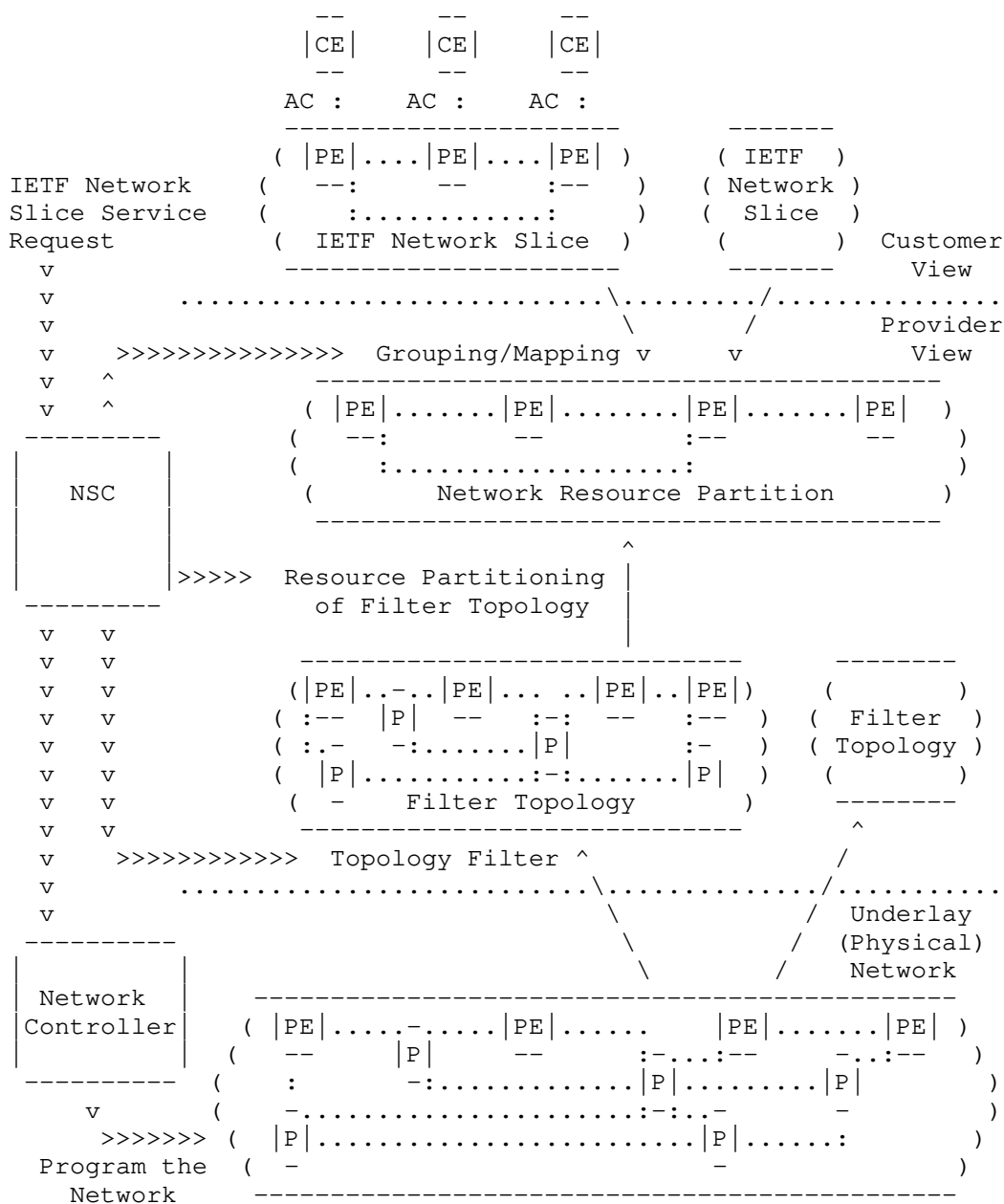


Figure 4: Architecture of an IETF Network Slice

The network itself (at the bottom of the figure) comprises an underlay network. This could be a physical network, but may be a virtual network. The underlay network is provisioned through network controllers that may utilize device controllers [RFC8309].

The underlay network may optionally be filtered or customized by the network operator to produce a number of network topologies that we call Filter Topologies. Customization is just a way of selecting specific resources (e.g., nodes and links) from the underlay network according to their capabilities and connectivity in the underlay network. These actions are configuration options or operator policies. The resulting topologies can be used as candidates to host IETF Network Slices and provide a useful way for the network operator to know in advance that all of the resources they are using to plan an IETF Network Slice would be able to meet specific SLOs and SLEs. The creation of a Filter Topology could be an offline planning activity or could be performed dynamically as new demands arise. The use of Filter Topologies is entirely optional in the architecture, and IETF Network Slices could be hosted directly on the underlay network.

Recall that an IETF Network Slice is a service requested by / provided for the customer. The IETF Network Slice service is expressed in terms of one or more connectivity constructs. An implementation or operator is free to limit the number of connectivity constructs in a slice to exactly one. Each connectivity construct is associated within the IETF Network Slice service request with a set of SLOs and SLEs. The set of SLOs and SLEs does not need to be the same for every connectivity construct in the slice, but an implementation or operator is free to require that all connectivity constructs in a slice have the same set of SLOs and SLEs.

One or more connectivity constructs from one or more slices are mapped to a set of network resources called a Network Resource Partition (NRP). A single connectivity construct is mapped to only one NRP (that is, the relationship is many to one). An NRP may be chosen to support a specific connectivity construct because of its ability to support a specific set of SLOs and SLEs, or its ability to support particular connectivity types, or for any administrative or operational reason. An implementation or operator is free to map each connectivity construct to a separate NRP, although there may be scaling implications depending on the solution implemented. Thus, the connectivity constructs from one slice may be mapped to one or more NRPs. By implication from the above, an implementation or operator is free to map all the connectivity constructs in a slice to a single NRP, and to not share that NRP with connectivity constructs from another slice.

An NRP is simply a collection of resources identified in the underlay network. Thus, the NRP is a scoped view of a topology and may be considered as a topology in its own right. The process of determining the NRP may be made easier if the underlay network topology is first filtered into a Filter Topology in order to be aware of the subset of network resources that are suitable for specific NRPs, but this is optional.

The steps described here can be applied in a variety of orders according to implementation and deployment preferences. Furthermore, the steps may be iterative so that the components are continually refined and modified as network conditions change and as service requests are received or relinquished, and even the underlay network could be extended if necessary to meet the customers' demands.

## 6.2. Procedures to Realize IETF Network Slices

There are a number of different technologies that can be used in the underlay, including physical connections, MPLS, time-sensitive networking (TSN), Flex-E, etc.

An IETF Network Slice can be realized in a network, using specific underlay technology or technologies. The creation of a new IETF Network Slice will be realized with following steps:

- \* The NSC exposes the network slicing capabilities that it offers for the network it manages so that the customer can determine whether to request services and what features are in scope.
- \* The customer may issue a request to determine whether a specific IETF Network Slice could be supported by the network. The NSC may respond indicating a simple yes or no, and may supplement a negative response with information about what it could support were the customer to change some requirements.
- \* The customer requests an IETF Network Slice. The NSC may respond that the slice has or has not been created, and may supplement a negative response with information about what it could support were the customer to change some requirements.
- \* When processing a customer request for an IETF Network Slice, the NSC maps the request to the network capabilities and applies provider policies before creating or supplementing the NRP.



Regardless of how IETF Network Slice is realized in the network (i.e., using tunnels of different types), the definition of the IETF Network Slice service does not change at all. The only difference is how the slice is realized. The following sections briefly introduce how some existing architectural approaches can be applied to realize IETF Network Slices.

### 6.3. Applicability of ACTN to IETF Network Slices

Abstraction and Control of TE Networks (ACTN - [RFC8453]) is a management architecture and toolkit used to create virtual networks (VNs) on top of a TE underlay network. The VNs can be presented to customers for them to operate as private networks.

In many ways, the function of ACTN is similar to IETF network slicing. Customer requests for connectivity-based overlay services are mapped to dedicated or shared resources in the underlay network in a way that meets customer guarantees for service level objectives and for separation from other customers' traffic. [RFC8453] describes the function of ACTN as collecting resources to establish a logically dedicated virtual network over one or more TE networks. Thus, in the case of a TE-enabled underlay network, the ACTN VN can be used as a basis to realize IETF network slicing.

While the ACTN framework is a generic VN framework that can be used for VN services beyond the IETF Network Slice, it also a suitable basis for delivering and realizing IETF Network Slices.

Further discussion of the applicability of ACTN to IETF Network Slices including a discussion of the relevant YANG models can be found in [I-D.ietf-teas-applicability-actn-slicing].

### 6.4. Applicability of Enhanced VPNs to IETF Network Slices

An enhanced VPN (VPN+) is designed to support the needs of new applications, particularly applications that are associated with 5G services, by utilizing an approach that is based on existing VPN and TE technologies and adds characteristics that specific services require over and above VPNs as they have previously been specified.

An enhanced VPN can be used to provide enhanced connectivity services between customer sites and can be used to create the infrastructure to underpin a IETF Network Slice service.

It is envisaged that enhanced VPNs will be delivered using a combination of existing, modified, and new networking technologies.

[I-D.ietf-teas-enhanced-vpn] describes the framework for Enhanced Virtual Private Network (VPN+) services.

#### 6.5. Network Slicing and Aggregation in IP/MPLS Networks

Network slicing provides the ability to partition a physical network into multiple isolated logical networks of varying sizes, structures, and functions so that each slice can be dedicated to specific services or customers.

Many approaches are currently being worked on to support IETF Network Slices in IP and MPLS networks with or without the use of Segment Routing. Most of these approaches utilize a way of marking packets so that network nodes can apply specific routing and forwarding behaviors to packets that belong to different IETF Network Slices. Different mechanisms for marking packets have been proposed (including using MPLS labels and Segment Routing segment IDs) and those mechanisms are agnostic to the path control technology used within the underlay network.

These approaches are also sensitive to the scaling concerns of supporting a large number of IETF Network Slices within a single IP or MPLS network, and so offer ways to aggregate the connectivity constructs of slices (or whole slices) so that the packet markings indicate an aggregate or grouping where all of the packets are subject to the same routing and forwarding behavior.

At this stage, it is inappropriate to mention any of these proposed solutions that are currently work in progress and not yet adopted as IETF work.

#### 6.6. Network Slicing and Service Function Chaining (SFC)

A customer may request an IETF Network Slice service that involves a set of service functions (SFs) together with the order in which these SFs are invoked. Also, the customer can specify the service objectives to be met by the underly network (e.g., one-way delay to cross a service function path, one-way delay to reach a specific SF). These SFs are considered as ancillary SDPs and are possibly placeholders (i.e., the SFs are identified, but not their locators).

Service Function Chaining (SFC) [RFC7665] techniques can be used by a provider to instantiate such an IETF Network Service Slice. The NSC may proceed as follows.

- \* Expose a set of ancillary SDPs that are hosted in the underlay network.

- \* Capture the SFC requirements (including, traffic performance metrics) from the customer. One or more service chains may be associated with the same IETF Network Slice service as connectivity constructs.
- \* Execute an SF placement algorithm to decide where to locate the ancillary SDPs in order to fulfil the service objectives.
- \* Generate SFC classification rules to identify (part of) the slice traffic that will be bound to an SFC. These classification rules may be the same as or distinct from the identification rules used to bind incoming traffic to the associated IETF Network Slice.

The NSC also generates a set of SFC forwarding policies that govern how the traffic will be forwarded along a service function path (SFP).

- \* Identify the appropriate Classifiers in the underlay network and provision them with the classification rules. Likewise, the NSC communicates the SFC forwarding policies to the appropriate Service Function Forwarders (SFF).

The provider can enable an SFC data plane mechanism, such as [RFC8300], [RFC8596], or [I-D.ietf-spring-nsh-sr].

## 7. Isolation in IETF Network Slices

### 7.1. Isolation as a Service Requirement

An IETF Network Slice customer may request that the IETF Network Slice delivered to them is such that changes to other IETF Network Slices or to other services do not have any negative impact on the delivery of the IETF Network Slice. The IETF Network Slice customer may specify the degree to which their IETF Network Slice is unaffected by changes in the provider network or by the behavior of other IETF Network Slice customers. The customer may express this via an SLE it agrees with the provider. This concept is termed 'isolation'.

In general, a customer cannot tell whether a service provider is meeting an isolation SLE. If the service varies such that an SLO is breached then the customer will become aware of the problem, and if the service varies within the allowed bounds of the SLOs, there may be no noticeable indication that this SLE has been violated.

## 7.2. Isolation in IETF Network Slice Realization

Isolation may be achieved in the underlay network by various forms of resource partitioning ranging from dedicated allocation of resources for a specific IETF Network Slice, to sharing of resources with safeguards. For example, traffic separation between different IETF Network Slices may be achieved using VPN technologies, such as L3VPN, L2VPN, EVPN, etc. Interference avoidance may be achieved by network capacity planning, allocating dedicated network resources, traffic policing or shaping, prioritizing in using shared network resources, etc. Finally, service continuity may be ensured by reserving backup paths for critical traffic, dedicating specific network resources for a selected number of IETF Network Slices.

## 8. Management Considerations

IETF Network Slice realization needs to be instrumented in order to track how it is working, and it might be necessary to modify the IETF Network Slice as requirements change. Dynamic reconfiguration might be needed.

The various management interfaces and components are discussed in Section 5.

## 9. Security Considerations

This document specifies terminology and has no direct effect on the security of implementations or deployments. In this section, a few of the security aspects are identified.

Conformance to security constraints: Specific security requests from customer-defined IETF Network Slices will be mapped to their realization in the underlay networks. Underlay networks will require capabilities to conform to customer's requests as some aspects of security may be expressed in SLEs.

IETF NSC authentication: Underlay networks need to be protected against the attacks from an adversary NSC as this could destabilize overall network operations. An IETF Network Slice may span across different networks, therefore, the NSC should have strong authentication with each of these networks. Furthermore, both the IETF Network Slice Service Interface and the Network Configuration Interface need to be secured.

Specific isolation criteria: The nature of conformance to isolation

requests means that it should not be possible to attack an IETF Network Slice service by varying the traffic on other services or slices carried by the same underlay network. In general, isolation is expected to strengthen the IETF Network Slice security.

**Data Integrity of an IETF Network Slice:** A customer wanting to secure their data and keep it private will be responsible for applying appropriate security measures to their traffic and not depending on the network operator that provides the IETF Network Slice. It is expected that for data integrity, a customer is responsible for end-to-end encryption of its own traffic. While an IETF Network Slice might include encryption and other security features as part of the service (for example as SLEs), customers might be well advised to take responsibility for their own security needs.

Note: See [NGMN\_SEC] on 5G network slice security for discussion relevant to this section.

IETF Network Slices might use underlying virtualized networking. All types of virtual networking require special consideration to be given to the separation of traffic between distinct virtual networks, as well as some degree of protection from effects of traffic use of underlay network (and other) resources from other virtual networks sharing those resources.

For example, if a service requires a specific upper bound of latency, then that service can be degraded by added delay in transmission of service packets caused by the activities of another service or application using the same resources.

Similarly, in a network with virtual functions, noticeably impeding access to a function used by another IETF Network Slice (for instance, compute resources) can be just as service-degrading as delaying physical transmission of associated packet in the network.

## 10. Privacy Considerations

Privacy of IETF Network Slice service customers must be preserved. It should not be possible for one IETF Network Slice customer to discover the presence of other customers, nor should sites that are members of one IETF Network Slice be visible outside the context of that IETF Network Slice.

In this sense, it is of paramount importance that the system use the privacy protection mechanism defined for the specific underlay technologies that support the slice, including in particular those mechanisms designed to preclude acquiring identifying information associated with any IETF Network Slice customer.

## 11. IANA Considerations

This document makes no requests for IANA action.

## 12. Informative References

- [HIPAA] HHS, "Health Insurance Portability and Accountability Act - The Security Rule", February 2003, <<https://www.hhs.gov/hipaa/for-professionals/security/index.html>>.
- [I-D.ietf-opsawg-sap] Boucadair, M., Dios, O. G. D., Barguil, S., Wu, Q., and V. Lopez, "A Network YANG Model for Service Attachment Points (SAPs)", Work in Progress, Internet-Draft, draft-ietf-opsawg-sap-03, 21 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-sap-03>>.
- [I-D.ietf-spring-nsh-sr] Guichard, J. N. and J. Tantsura, "Integration of Network Service Header (NSH) and Segment Routing for Service Function Chaining (SFC)", Work in Progress, Internet-Draft, draft-ietf-spring-nsh-sr-10, 13 December 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-spring-nsh-sr-10>>.
- [I-D.ietf-teas-applicability-actn-slicing] King, D., Drake, J., Zheng, H., and A. Farrel, "Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing", Work in Progress, Internet-Draft, draft-ietf-teas-applicability-actn-slicing-01, 7 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-applicability-actn-slicing-01>>.

- [I-D.ietf-teas-enhanced-vpn]  
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", Work in Progress, Internet-Draft, draft-ietf-teas-enhanced-vpn-10, 6 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-enhanced-vpn-10>>.
- [I-D.openconfig-rtgwg-gnmi-spec]  
Shakir, R., Shaikh, A., Borman, P., Hines, M., Lebsack, C., and C. Morrow, "gRPC Network Management Interface (gNMI)", Work in Progress, Internet-Draft, draft-openconfig-rtgwg-gnmi-spec-01, 5 March 2018, <<https://datatracker.ietf.org/doc/html/draft-openconfig-rtgwg-gnmi-spec-01>>.
- [MACsec] IEEE, "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Security", 2018, <<https://1.ieee802.org/security/802-lae>>.
- [NGMN-NS-Concept]  
NGMN Alliance, "Description of Network Slicing Concept", [https://www.ngmn.org/uploads/media/161010\\_NGMN\\_Network\\_Slicing\\_framework\\_v1.0.8.pdf](https://www.ngmn.org/uploads/media/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf), 2016.
- [NGMN\_SEC] NGMN Alliance, "NGMN 5G Security - Network Slicing", April 2016, <[https://www.ngmn.org/wp-content/uploads/Publication\\_s/2016/160429\\_NGMN\\_5G\\_Security\\_Network\\_Slicing\\_v1\\_0.pdf](https://www.ngmn.org/wp-content/uploads/Publication_s/2016/160429_NGMN_5G_Security_Network_Slicing_v1_0.pdf)>.
- [PCI] PCI Security Standards Council, "PCI DSS", May 2018, <<https://www.pcisecuritystandards.org>>.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<https://www.rfc-editor.org/info/rfc3022>>.
- [RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, DOI 10.17487/RFC3135, June 2001, <<https://www.rfc-editor.org/info/rfc3135>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <<https://www.rfc-editor.org/info/rfc3393>>.

- [RFC4208] Swallow, G., Drake, J., Ishimatsu, H., and Y. Rekhter, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC 4208, DOI 10.17487/RFC4208, October 2005, <<https://www.rfc-editor.org/info/rfc4208>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC4397] Bryskin, I. and A. Farrel, "A Lexicography for the Interpretation of Generalized Multiprotocol Label Switching (GMPLS) Terminology within the Context of the ITU-T's Automatically Switched Optical Network (ASON) Architecture", RFC 4397, DOI 10.17487/RFC4397, February 2006, <<https://www.rfc-editor.org/info/rfc4397>>.
- [RFC5212] Shiimoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", RFC 5212, DOI 10.17487/RFC5212, July 2008, <<https://www.rfc-editor.org/info/rfc5212>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<https://www.rfc-editor.org/info/rfc6146>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.



- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.
- [RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <<https://www.rfc-editor.org/info/rfc7926>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.
- [RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8454] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., and B. Yoon, "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC 8454, DOI 10.17487/RFC8454, September 2018, <<https://www.rfc-editor.org/info/rfc8454>>.

- [RFC8596] Malis, A., Bryant, S., Halpern, J., and W. Henderickx, "MPLS Transport Encapsulation for the Service Function Chaining (SFC) Network Service Header (NSH)", RFC 8596, DOI 10.17487/RFC8596, June 2019, <<https://www.rfc-editor.org/info/rfc8596>>.
- [TS23501] 3GPP, "System architecture for the 5G System (5GS)", 3GPP TS 23.501, 2019.
- [TS28530] 3GPP, "Management and orchestration; Concepts, use cases and requirements", 3GPP TS 28.530, 2019.
- [TS33.210] 3GPP, "3G security; Network Domain Security (NDS); IP network layer security (Release 14).", December 2016, <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2279>>.

#### Acknowledgments

The entire TEAS Network Slicing design team and everyone participating in related discussions has contributed to this document. Some text fragments in the document have been copied from the [I-D.ietf-teas-enhanced-vpn], for which we are grateful.

Significant contributions to this document were gratefully received from the contributing authors listed in the "Contributors" section. In addition we would like to also thank those others who have attended one or more of the design team meetings, including the following people not listed elsewhere:

- \* Aihua Guo
- \* Bo Wu
- \* Greg Mirsky
- \* Lou Berger
- \* Rakesh Gandhi
- \* Ran Chen
- \* Sergio Belotti
- \* Stewart Bryant
- \* Tomonobu Niwa

\* Xuesong Geng

Further useful comments were received from Daniele Ceccarelli, Uma Chunduri, Pavan Beeram, Tarek Saad, Kenichi Ogaki, Oscar Gonzalez de Dios, Xiaobing Niu, Dan Voyer, Igor Bryskin, Luay Jalil, Joel Halpern, John Scudder, John Mullooly, and Krzysztof Szarkowicz.

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).

#### Contributors

The following authors contributed significantly to this document:

Eric Gray  
(The original editor of the foundation documents)  
Independent  
Email: ewgray@graiymage.com

Jari Arkko  
Ericsson  
Email: jari.arkko@piuha.net

Mohamed Boucadair  
Orange  
Email: mohamed.boucadair@orange.com

Dhruv Dhody  
Huawei, India  
Email: dhruv.ietf@gmail.com

Jie Dong  
Huawei  
Email: jie.dong@huawei.com

Xufeng Liu  
Volta Networks  
Email: xufeng.liu.ietf@gmail.com

#### Authors' Addresses

Adrian Farrel (editor)  
Old Dog Consulting  
United Kingdom  
Email: adrian@olddog.co.uk

John Drake (editor)  
Juniper Networks  
United States of America  
Email: jdrake@juniper.net

Reza Rokui  
Ciena  
Email: rrokui@ciena.com

Shunsuke Homma  
NTT  
Japan  
Email: shunsuke.homma.ietf@gmail.com

Kiran Makhijani  
Futurewei  
United States of America  
Email: kiranm@futurewei.com

Luis M. Contreras  
Telefonica  
Spain  
Email: luismiguel.contrerasmurillo@telefonica.com

Jeff Tantsura  
Microsoft Inc.  
Email: jefftant.ietf@gmail.com

TEAS Working Group  
Internet-Draft  
Obsoletes: 3272 (if approved)  
Intended status: Informational  
Expires: November 16, 2021

A. Farrel, Ed.  
Old Dog Consulting  
May 15, 2021

Overview and Principles of Internet Traffic Engineering  
draft-ietf-teas-rfc3272bis-12

Abstract

This document describes the principles of traffic engineering (TE) in the Internet. The document is intended to promote better understanding of the issues surrounding traffic engineering in IP networks and the networks that support IP networking, and to provide a common basis for the development of traffic engineering capabilities for the Internet. The principles, architectures, and methodologies for performance evaluation and performance optimization of operational networks are also discussed.

This work was first published as RFC 3272 in May 2002. This document obsoletes RFC 3272 by making a complete update to bring the text in line with best current practices for Internet traffic engineering and to include references to the latest relevant work in the IETF.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 16, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .   | 4  |
| 1.1. What is Internet Traffic Engineering? . . . . .                        | 4  |
| 1.2. Components of Traffic Engineering . . . . .                            | 6  |
| 1.3. Scope . . . . .  | 8  |
| 1.4. Terminology . . . . .  | 8  |
| 2. Background . . . . .   | 11 |
| 2.1. Context of Internet Traffic Engineering . . . . .                      | 11 |
| 2.2. Network Domain Context . . . . .                                       | 12 |
| 2.3. Problem Context . . . . .  | 14 |
| 2.3.1. Congestion and its Ramifications . . . . .                           | 15 |
| 2.4. Solution Context . . . . .   | 16 |
| 2.4.1. Combating the Congestion Problem . . . . .                           | 18 |
| 2.5. Implementation and Operational Context . . . . .                       | 21 |
| 3. Traffic Engineering Process Models . . . . .                             | 21 |
| 3.1. Components of the Traffic Engineering Process Model . . . . .          | 22 |
| 4. Taxonomy of Traffic Engineering Systems . . . . .                        | 22 |
| 4.1. Time-Dependent Versus State-Dependent Versus Event-Dependent . . . . . | 23 |
| 4.2. Offline Versus Online . . . . .  | 24 |
| 4.3. Centralized Versus Distributed . . . . .                               | 25 |
| 4.3.1. Hybrid Systems . . . . .   | 25 |
| 4.3.2. Considerations for Software Defined Networking . . . . .             | 26 |
| 4.4. Local Versus Global . . . . .  | 26 |
| 4.5. Prescriptive Versus Descriptive . . . . .                              | 27 |
| 4.5.1. Intent-Based Networking . . . . .                                    | 27 |
| 4.6. Open-Loop Versus Closed-Loop . . . . .                                 | 28 |
| 4.7. Tactical versus Strategic . . . . .                                    | 28 |
| 5. Review of TE Techniques . . . . .  | 28 |
| 5.1. Overview of IETF Projects Related to Traffic Engineering . . . . .     | 28 |
| 5.1.1. Constraint-Based Routing . . . . .                                   | 29 |
| 5.1.2. Integrated Services . . . . .  | 31 |
| 5.1.3. RSVP . . . . .   | 31 |
| 5.1.4. Differentiated Services . . . . .                                    | 32 |
| 5.1.5. QUIC . . . . .   | 33 |
| 5.1.6. Multiprotocol Label Switching (MPLS) . . . . .                       | 34 |
| 5.1.7. Generalized MPLS (GMPLS) . . . . .                                   | 34 |

|             |   |    |
|-------------|---|----|
| 5.1.8.      | IP Performance Metrics . . . . .  | 35 |
| 5.1.9.      | Flow Measurement . . . . .  | 35 |
| 5.1.10.     | Endpoint Congestion Management . . . . .                                      | 36 |
| 5.1.11.     | TE Extensions to the IGPs . . . . .   | 36 |
| 5.1.12.     | Link-State BGP . . . . .  | 37 |
| 5.1.13.     | Path Computation Element . . . . .  | 37 |
| 5.1.14.     | Application-Layer Traffic Optimization . . . . .                              | 38 |
| 5.1.15.     | Segment Routing with MPLS Encapsulation (SR-MPLS) . . . . .                   | 39 |
| 5.1.16.     | Segment Routing Policy . . . . .  | 40 |
| 5.1.17.     | Network Virtualization and Abstraction . . . . .                              | 41 |
| 5.1.18.     | Network Slicing . . . . .   | 41 |
| 5.1.19.     | Deterministic Networking . . . . .  | 42 |
| 5.1.20.     | Network TE State Definition and Presentation . . . . .                        | 43 |
| 5.1.21.     | System Management and Control Interfaces . . . . .                            | 43 |
| 5.2.        | Content Distribution . . . . .  | 43 |
| 6.          | Recommendations for Internet Traffic Engineering . . . . .                    | 44 |
| 6.1.        | Generic Non-functional Recommendations . . . . .                              | 44 |
| 6.2.        | Routing Recommendations . . . . .   | 46 |
| 6.3.        | Traffic Mapping Recommendations . . . . .                                     | 49 |
| 6.4.        | Measurement Recommendations . . . . .   | 50 |
| 6.5.        | Policing, Planning, and Access Control . . . . .                              | 50 |
| 6.6.        | Network Survivability . . . . .   | 51 |
| 6.6.1.      | Survivability in MPLS Based Networks . . . . .                                | 53 |
| 6.6.2.      | Protection Options . . . . .  | 54 |
| 6.7.        | Multi-Layer Traffic Engineering . . . . .                                     | 55 |
| 6.8.        | Traffic Engineering in Diffserv Environments . . . . .                        | 56 |
| 6.9.        | Network Controllability . . . . .   | 57 |
| 7.          | Inter-Domain Considerations . . . . .   | 58 |
| 8.          | Overview of Contemporary TE Practices in Operational IP<br>Networks . . . . . | 60 |
| 9.          | Security Considerations . . . . .   | 62 |
| 10.         | IANA Considerations . . . . .   | 63 |
| 11.         | Acknowledgments . . . . .   | 63 |
| 12.         | Contributors . . . . .  | 64 |
| 13.         | Informative References . . . . .  | 65 |
| Appendix A. | Historic Overview . . . . .   | 78 |
| A.1.        | Traffic Engineering in Classical Telephone Networks . . . . .                 | 78 |
| A.2.        | Evolution of Traffic Engineering in Packet Networks . . . . .                 | 79 |
| A.2.1.      | Adaptive Routing in the ARPANET . . . . .                                     | 80 |
| A.2.2.      | Dynamic Routing in the Internet . . . . .                                     | 80 |
| A.2.3.      | ToS Routing . . . . .   | 81 |
| A.2.4.      | Equal Cost Multi-Path . . . . .   | 81 |
| A.2.5.      | Nimrod . . . . .  | 82 |
| A.3.        | Development of Internet Traffic Engineering . . . . .                         | 82 |
| A.3.1.      | Overlay Model . . . . .   | 82 |
| Appendix B. | Overview of Traffic Engineering Related Work in<br>Other SDOs . . . . .       | 83 |
| B.1.        | Overview of ITU Activities Related to Traffic Engineering . . . . .           | 83 |

|   |    |
|---|----|
| Appendix C. Summary of Changes Since RFC 3272 . . . . . | 84 |
| C.1. RFC 3272 . . . . .                                 | 84 |
| C.2. This Document . . . . .                            | 87 |
| Author's Address . . . . .                              | 90 |

## 1. Introduction

This document describes the principles of Internet traffic engineering (TE). The objective of the document is to articulate the general issues and principles for Internet traffic engineering, and where appropriate to provide recommendations, guidelines, and options for the development of preplanned (offline) and dynamic (online) Internet traffic engineering capabilities and support systems.

This document provides a terminology and taxonomy for describing and understanding common Internet traffic engineering concepts.

Even though Internet traffic engineering is most effective when applied end-to-end, the focus of this document is traffic engineering within a given domain (such as an autonomous system). However, because a preponderance of Internet traffic tends to originate in one autonomous system and terminate in another, this document also provides an overview of aspects pertaining to inter-domain traffic engineering.

This work was first published as [RFC3272] in May 2002. This document obsoletes [RFC3272] by making a complete update to bring the text in line with best current practices for Internet traffic engineering and to include references to the latest relevant work in the IETF. It is worth noting around three fifths of the RFCs referenced in this document post-date the publication of RFC 3272. Appendix C provides a summary of changes between RFC 3272 and this document.

### 1.1. What is Internet Traffic Engineering?

One of the most significant functions performed in the Internet is the routing and forwarding of traffic from ingress nodes to egress nodes. Therefore, one of the most distinctive functions performed by Internet traffic engineering is the control and optimization of these routing and forwarding functions, to steer traffic through the network.

Internet traffic engineering is defined as that aspect of Internet network engineering dealing with the issues of performance evaluation and performance optimization of operational IP networks. Traffic engineering encompasses the application of technology and scientific



principles to the measurement, characterization, modeling, and control of Internet traffic [RFC2702], [AWD2].

It is the performance of the network as seen by end users of network services that is paramount. The characteristics visible to end users are the emergent properties of the network, which are the characteristics of the network when viewed as a whole. A central goal of the service provider, therefore, is to enhance the emergent properties of the network while taking economic considerations into account. This is accomplished by addressing traffic oriented performance requirements while utilizing network resources economically and reliably. Traffic oriented performance measures include delay, delay variation, packet loss, and throughput.

Internet traffic engineering responds to network events. Aspects of capacity management respond at intervals ranging from days to years. Routing control functions operate at intervals ranging from milliseconds to days. Packet level processing functions operate at very fine levels of temporal resolution, ranging from picoseconds to milliseconds while reacting to the real-time statistical behavior of traffic.

Thus, the optimization aspects of traffic engineering can be viewed from a control perspective, and can be both pro-active and reactive. In the pro-active case, the traffic engineering control system takes preventive action to protect against predicted unfavorable future network states, for example, by engineering backup paths. It may also take action that will lead to a more desirable future network state. In the reactive case, the control system responds to correct issues and adapt to network events, such as routing after failure.

Another important objective of Internet traffic engineering is to facilitate reliable network operations [RFC2702]. Reliable network operations can be facilitated by providing mechanisms that enhance network integrity and by embracing policies emphasizing network survivability. This reduces the vulnerability of services to outages arising from errors, faults, and failures occurring within the network infrastructure.

The optimization aspects of traffic engineering can be achieved through capacity management and traffic management. In this document, capacity management includes capacity planning, routing control, and resource management. Network resources of particular interest include link bandwidth, buffer space, and computational resources. In this document, traffic management includes:

1. Nodal traffic control functions such as traffic conditioning, queue management, and scheduling

2. Other functions that regulate the flow of traffic through the network or that arbitrate access to network resources between different packets or between different traffic flows.

One major challenge of Internet traffic engineering is the realization of automated control capabilities that adapt quickly and cost effectively to significant changes in network state, while still maintaining stability of the network. Performance evaluation can assess the effectiveness of traffic engineering methods, and the results of this evaluation can be used to identify existing problems, guide network re-optimization, and aid in the prediction of potential future problems. However, this process can also be time consuming and may not be suitable to act on short-lived changes in the network.

Performance evaluation can be achieved in many different ways. The most notable techniques include analytical methods, simulation, and empirical methods based on measurements.

Traffic engineering comes in two flavors: either a background process that constantly monitors traffic and optimizes the use of resources to improve performance; or a form of a pre-planned optimized traffic distribution that is considered optimal. In the later case, any deviation from the optimum distribution (e.g., caused by a fiber cut) is reverted upon repair without further optimization. However, this form of traffic engineering relies upon the notion that the planned state of the network is optimal. Hence, in such a mode there are two levels of traffic engineering: the TE-planning task to enable optimum traffic distribution, and the routing and forwarding tasks that keep traffic flows attached to the pre-planned distribution.

As a general rule, traffic engineering concepts and mechanisms must be sufficiently specific and well-defined to address known requirements, but simultaneously flexible and extensible to accommodate unforeseen future demands (see Section 6.1).

## 1.2. Components of Traffic Engineering

As mentioned in Section 1.1, Internet traffic engineering provides performance optimization of IP networks while utilizing network resources economically and reliably. Such optimization is supported at the control/controller level and within the data/forwarding plane.

The key elements required in any TE solution are as follows:

1. Policy
2. Path steering

### 3. Resource management

Some TE solutions rely on these elements to a lesser or greater extent. Debate remains about whether a solution can truly be called traffic engineering if it does not include all of these elements. For the sake of this document, we assert that all TE solutions must include some aspects of all of these elements. Other solutions can be classed as "partial TE" and also fall in scope of this document.

Policy allows for the selection of paths (including next hops) based on information beyond basic reachability. Early definitions of routing policy, e.g., [RFC1102] and [RFC1104], discuss routing policy being applied to restrict access to network resources at an aggregate level. BGP is an example of a commonly used mechanism for applying such policies, see [RFC4271] and [RFC8955]. In the traffic engineering context, policy decisions are made within the control plane or by controllers, and govern the selection of paths. Examples can be found in [RFC4655] and [RFC5394]. Standard TE solutions may cover the mechanisms to distribute and/or enforce policies, but specific policy definition is generally unspecified.

Path steering is the ability to forward packets using more information than just knowledge of the next hop. Examples of path steering include IPv4 source routes [RFC0791], RSVP-TE explicit routes [RFC3209], Segment Routing [RFC8402], and Service Function Chaining [RFC7665]. Path steering for TE can be supported via control plane protocols, by encoding in the data plane headers, or by a combination of the two. This includes when control is provided by a controller using a network-facing control protocol.

Resource management provides resource-aware control and forwarding. Examples of resources are bandwidth, buffers, and queues, all of which can be managed to control loss and latency.

Resource reservation is the control aspect of resource management. It provides for domain-wide consensus about which network resources are used by a particular flow. This determination may be made at a very coarse or very fine level. Note that this consensus exists at the network control or controller level, not within the data plane. It may be composed purely of accounting/bookkeeping, but it typically includes an ability to admit, reject, or reclassify a flow based on policy. Such accounting can be done based on any combination of a static understanding of resource requirements, and the use of dynamic mechanisms to collect requirements (e.g., via [RFC3209]) and resource availability (e.g., via [RFC4203]).

Resource allocation is the data plane aspect of resource management. It provides for the allocation of specific node and link resources to specific flows. Example resources include buffers, policing, and rate-shaping mechanisms that are typically supported via queuing. It also includes the matching of a flow (i.e., flow classification) to a particular set of allocated resources. The method of flow classification and granularity of resource management is technology specific. Examples include Diffserv with dropping and remarking [RFC4594], MPLS-TE [RFC3209], and GMPLS based label switched paths [RFC3945], as well as controller-based solutions [RFC8453]. This level of resource control, while optional, is important in networks that wish to support congestion management policies to control or regulate the offered traffic to deliver different levels of service and alleviate congestion problems, or those networks that wish to control latencies experienced by specific traffic flows.

### 1.3. Scope

The scope of this document is intra-domain traffic engineering. That is, traffic engineering within a given autonomous system in the Internet. This document discusses concepts pertaining to intra-domain traffic control, including such issues as routing control, micro and macro resource allocation, and the control coordination problems that arise consequently.

This document describes and characterizes techniques already in use or in advanced development for Internet traffic engineering. The way these techniques fit together is discussed and scenarios in which they are useful will be identified.

Although the emphasis in this document is on intra-domain traffic engineering, an overview of the high level considerations pertaining to inter-domain traffic engineering is provided in Section 7. Inter-domain Internet traffic engineering is crucial to the performance enhancement of the global Internet infrastructure.

Whenever possible, relevant requirements from existing IETF documents and other sources are incorporated by reference.

### 1.4. Terminology

This section provides terminology which is useful for Internet traffic engineering. The definitions presented apply to this document. These terms may have other meanings elsewhere.

**Busy hour:** A one hour period within a specified interval of time (typically 24 hours) in which the traffic load in a network or sub-network is greatest.

**Congestion:** A state of a network resource in which the traffic incident on the resource exceeds its output capacity over an interval of time.

**Congestion avoidance:** An approach to congestion management that attempts to obviate the occurrence of congestion.

**Congestion control:** An approach to congestion management that attempts to remedy congestion problems that have already occurred.

**Constraint-based routing:** A class of routing protocols that take specified traffic attributes, network constraints, and policy constraints into account when making routing decisions. Constraint-based routing is applicable to traffic aggregates as well as flows. It is a generalization of QoS-based routing.

**Demand side congestion management:** A congestion management scheme that addresses congestion problems by regulating or conditioning offered load.

**Effective bandwidth:** The minimum amount of bandwidth that can be assigned to a flow or traffic aggregate in order to deliver 'acceptable service quality' to the flow or traffic aggregate.

**Hot-spot:** A network element or subsystem which is in a state of congestion.

**Inter-domain traffic:** Traffic that originates in one Autonomous system and terminates in another.

**Metric:** A parameter defined in terms of standard units of measurement.

**Measurement methodology:** A repeatable measurement technique used to derive one or more metrics of interest.

**Network survivability:** The capability to provide a prescribed level of QoS for existing services after a given number of failures occur within the network.

**Offline traffic engineering:** A traffic engineering system that exists outside of the network.

**Online traffic engineering:** A traffic engineering system that exists within the network, typically implemented on or as adjuncts to operational network elements.

**Performance measures:** Metrics that provide quantitative or qualitative measures of the performance of systems or subsystems of interest.

**Performance metric:** A performance parameter defined in terms of standard units of measurement.

**Provisioning:** The process of assigning or configuring network resources to meet certain requests.

**QoS routing:** Class of routing systems that selects paths to be used by a flow based on the QoS requirements of the flow.

**Service Level Agreement (SLA):** A contract between a provider and a customer that guarantees specific levels of performance and reliability at a certain cost.

**Service Level Objective (SLO):** A key element of an SLA between a provider and a customer. SLOs are agreed upon as a means of measuring the performance of the Service Provider and are outlined as a way of avoiding disputes between the two parties based on misunderstanding.

**Stability:** An operational state in which a network does not oscillate in a disruptive manner from one mode to another mode.

**Supply-side congestion management:** A congestion management scheme that provisions additional network resources to address existing and/or anticipated congestion problems.

**Traffic characteristic:** A description of the temporal behavior or a description of the attributes of a given traffic flow or traffic aggregate.

**Traffic engineering system:** A collection of objects, mechanisms, and protocols that are used together to accomplish traffic engineering objectives.

**Traffic flow:** A stream of packets between two end-points that can be characterized in a certain way. A common classification for a traffic flow selects packets with the "five-tuple" of source and destination addresses, source and destination ports, and protocol ID.

**Traffic matrix:** A representation of the traffic demand between a set of origin and destination abstract nodes. An abstract node can consist of one or more network elements.

**Traffic monitoring:** The process of observing traffic characteristics at a given point in a network and collecting the traffic information for analysis and further action.

**Traffic trunk:** An aggregation of traffic flows belonging to the same class which are forwarded through a common path. A traffic trunk may be characterized by an ingress and egress node, and a set of attributes which determine its behavioral characteristics and requirements from the network.

## 2. Background

The Internet aims to convey IP packets from ingress nodes to egress nodes efficiently, expeditiously, and economically. Furthermore, in a multiclass service environment (e.g., Diffserv capable networks - see Section 5.1.4), the resource sharing parameters of the network must be appropriately determined and configured according to prevailing policies and service models to resolve resource contention issues arising from mutual interference between packets traversing through the network. Thus, consideration must be given to resolving competition for network resources between traffic flows belonging to the same service class (intra-class contention resolution) and traffic flows belonging to different classes (inter-class contention resolution).

### 2.1. Context of Internet Traffic Engineering

The context of Internet traffic engineering includes:

1. A network domain context that defines the scope under consideration, and in particular the situations in which the traffic engineering problems occur. The network domain context includes network structure, network policies, network characteristics, network constraints, network quality attributes, and network optimization criteria.
2. A problem context defining the general and concrete issues that traffic engineering addresses. The problem context includes identification, abstraction of relevant features, representation, formulation, specification of the requirements on the solution space, and specification of the desirable features of acceptable solutions.

3. A solution context suggesting how to address the issues identified by the problem context. The solution context includes analysis, evaluation of alternatives, prescription, and resolution.
4. An implementation and operational context in which the solutions are instantiated. The implementation and operational context includes planning, organization, and execution.

The context of Internet traffic engineering and the different problem scenarios are discussed in the following subsections.

## 2.2. Network Domain Context

IP networks range in size from small clusters of routers situated within a given location, to thousands of interconnected routers, switches, and other components distributed all over the world.

At the most basic level of abstraction, an IP network can be represented as a distributed dynamic system consisting of:

- o a set of interconnected resources which provide transport services for IP traffic subject to certain constraints
- o a demand system representing the offered load to be transported through the network
- o a response system consisting of network processes, protocols, and related mechanisms which facilitate the movement of traffic through the network (see also [AWD2]).

The network elements and resources may have specific characteristics restricting the manner in which the traffic demand is handled. Additionally, network resources may be equipped with traffic control mechanisms managing the way in which the demand is serviced. Traffic control mechanisms may be used to:

- o control packet processing activities within a given resource
- o arbitrate contention for access to the resource by different packets
- o regulate traffic behavior through the resource.

A configuration management and provisioning system may allow the settings of the traffic control mechanisms to be manipulated by external or internal entities in order to exercise control over the



way in which the network elements respond to internal and external stimuli.

The details of how the network carries packets are specified in the policies of the network administrators and are installed through network configuration management and policy based provisioning systems. Generally, the types of service provided by the network also depend upon the technology and characteristics of the network elements and protocols, the prevailing service and utility models, and the ability of the network administrators to translate policies into network configurations.

Internet networks have two significant characteristics:

- o they provide real-time services
- o their operating environments are very dynamic.

The dynamic characteristics of IP and IP/MPLS networks can be attributed in part to fluctuations in demand, to the interaction between various network protocols and processes, to the rapid evolution of the infrastructure which demands the constant inclusion of new technologies and new network elements, and to transient and persistent faults which occur within the system.

Packets contend for the use of network resources as they are conveyed through the network. A network resource is considered to be congested if, for an interval of time, the arrival rate of packets exceed the output capacity of the resource. Congestion may result in some of the arriving packets being delayed or even dropped.

Congestion increases transit delay, delay variation, may lead to packet loss, and reduces the predictability of network services. Clearly, congestion is highly undesirable. Combating congestion at a reasonable cost is a major objective of Internet traffic engineering.

Efficient sharing of network resources by multiple traffic flows is a basic operational premise for the Internet. A fundamental challenge in network operation is to increase resource utilization while minimizing the possibility of congestion.

The Internet has to function in the presence of different classes of traffic with different service requirements. This requirement is clarified in [RFC2475] which also provides an architecture for Differentiated Services (Diffserv). That document describes how packets can be grouped into behavior aggregates such that each aggregate has a common set of behavioral characteristics or a common set of delivery requirements. Delivery requirements of a specific

set of packets may be specified explicitly or implicitly. Two of the most important traffic delivery requirements are:

- o Capacity constraints can be expressed statistically as peak rates, mean rates, burst sizes, or as some deterministic notion of effective bandwidth.
- o QoS requirements can be expressed in terms of:
  - \* integrity constraints such as packet loss
  - \* temporal constraints such as timing restrictions for the delivery of each packet (delay) and timing restrictions for the delivery of consecutive packets belonging to the same traffic stream (delay variation).

### 2.3. Problem Context

There are several problems associated with operating a network described in the previous section. This section analyzes the problem context in relation to traffic engineering. The identification, abstraction, representation, and measurement of network features relevant to traffic engineering are significant issues.

A particular challenge is to formulate the problems that traffic engineering attempts to solve. For example:

- o how to identify the requirements on the solution space
- o how to specify the desirable features of solutions
- o how to actually solve the problems
- o how to measure and characterize the effectiveness of solutions.

Another class of problems is how to measure and estimate relevant network state parameters. Effective traffic engineering relies on a good estimate of the offered traffic load as well as a view of the underlying topology and associated resource constraints. A network-wide view of the topology is also a must for offline planning.

Still another class of problem is how to characterize the state of the network and how to evaluate its performance. The performance evaluation problem is two-fold: one aspect relates to the evaluation of the system-level performance of the network; the other aspect relates to the evaluation of resource-level performance, which restricts attention to the performance analysis of individual network resources.

In this document, we refer to the system-level characteristics of the network as the "macro-states" and the resource-level characteristics as the "micro-states." The system-level characteristics are also known as the emergent properties of the network. Correspondingly, we refer to the traffic engineering schemes dealing with network performance optimization at the systems level as "macro-TE" and the schemes that optimize at the individual resource level as "micro-TE." Under certain circumstances, the system-level performance can be derived from the resource-level performance using appropriate rules of composition, depending upon the particular performance measures of interest.

Another fundamental class of problem concerns how to effectively optimize network performance. Performance optimization may entail translating solutions for specific traffic engineering problems into network configurations. Optimization may also entail some degree of resource management control, routing control, and capacity augmentation.

#### 2.3.1. Congestion and its Ramifications

Congestion is one of the most significant problems in an operational IP context. A network element is said to be congested if it experiences sustained overload over an interval of time. Congestion almost always results in degradation of service quality to end users. Congestion control schemes can include demand-side policies and supply-side policies. Demand-side policies may restrict access to congested resources or dynamically regulate the demand to alleviate the overload situation. Supply-side policies may expand or augment network capacity to better accommodate offered traffic. Supply-side policies may also re-allocate network resources by redistributing traffic over the infrastructure. Traffic redistribution and resource re-allocation serve to increase the 'effective capacity' of the network.

The emphasis of this document is primarily on congestion management schemes falling within the scope of the network, rather than on congestion management systems dependent upon sensitivity and adaptivity from end-systems. That is, the aspects that are considered in this document with respect to congestion management are those solutions that can be provided by control entities operating on the network and by the actions of network administrators and network operations systems.

## 2.4. Solution Context

The solution context for Internet traffic engineering involves analysis, evaluation of alternatives, and choice between alternative courses of action. Generally the solution context is based on making inferences about the current or future state of the network, and making decisions that may involve a preference between alternative sets of action. More specifically, the solution context demands reasonable estimates of traffic workload, characterization of network state, derivation of solutions which may be implicitly or explicitly formulated, and possibly instantiating a set of control actions. Control actions may involve the manipulation of parameters associated with routing, control over tactical capacity acquisition, and control over the traffic management functions.

The following list of instruments may be applicable to the solution context of Internet traffic engineering.

- o A set of policies, objectives, and requirements (which may be context dependent) for network performance evaluation and performance optimization.
- o A collection of online and possibly offline tools and mechanisms for measurement, characterization, modeling, and control traffic, and control over the placement and allocation of network resources, as well as control over the mapping or distribution of traffic onto the infrastructure.
- o A set of constraints on the operating environment, the network protocols, and the traffic engineering system itself.
- o A set of quantitative and qualitative techniques and methodologies for abstracting, formulating, and solving traffic engineering problems.
- o A set of administrative control parameters which may be manipulated through a configuration management system. Such system itself may include a configuration control subsystem, a configuration repository, a configuration accounting subsystem, and a configuration auditing subsystem.
- o A set of guidelines for network performance evaluation, performance optimization, and performance improvement.

Determining traffic characteristics through measurement or estimation is very useful within the realm the traffic engineering solution space. Traffic estimates can be derived from customer subscription information, traffic projections, traffic models, and from actual

measurements. The measurements may be performed at different levels, e.g., at the traffic-aggregate level or at the flow level. Measurements at the flow level or on small traffic aggregates may be performed at edge nodes, when traffic enters and leaves the network. Measurements for large traffic-aggregates may be performed within the core of the network.

To conduct performance studies and to support planning of existing and future networks, a routing analysis may be performed to determine the paths the routing protocols will choose for various traffic demands, and to ascertain the utilization of network resources as traffic is routed through the network. Routing analysis captures the selection of paths through the network, the assignment of traffic across multiple feasible routes, and the multiplexing of IP traffic over traffic trunks (if such constructs exist) and over the underlying network infrastructure. A model of network topology is necessary to perform routing analysis. A network topology model may be extracted from:

- o network architecture documents
- o network designs
- o information contained in router configuration files
- o routing databases
- o routing tables
- o automated tools that discover and collate network topology information.

Topology information may also be derived from servers that monitor network state, and from servers that perform provisioning functions.

Routing in operational IP networks can be administratively controlled at various levels of abstraction including the manipulation of BGP attributes and interior gateway protocol (IGP) metrics. For path oriented technologies such as MPLS, routing can be further controlled by the manipulation of relevant traffic engineering parameters, resource parameters, and administrative policy constraints. Within the context of MPLS, the path of an explicitly routed label switched path (LSP) can be computed and established in various ways including:

- o manually
- o automatically, online using constraint-based routing processes implemented on label switching routers

- o automatically, offline using constraint-based routing entities implemented on external traffic engineering support systems.

#### 2.4.1. Combating the Congestion Problem

Minimizing congestion is a significant aspect of Internet traffic engineering. This subsection gives an overview of the general approaches that have been used or proposed to combat congestion.

Congestion management policies can be categorized based upon the following criteria (see [YARE95] for a more detailed taxonomy of congestion control schemes):

##### 1. Congestion Management Based on Response Time Scales

- \* Long (weeks to months): Expanding network capacity by adding new equipment, routers, and links takes time and is comparatively costly. Capacity planning needs to take this into consideration. Network capacity is expanded based on estimates or forecasts of future traffic development and traffic distribution. These upgrades are typically carried out over weeks or months, or maybe even years.
- \* Medium (minutes to days): Several control policies fall within the medium timescale category. Examples include:
  - a. Adjusting routing protocol parameters to route traffic away or towards certain segments of the network.
  - b. Setting up or adjusting explicitly routed LSPs in MPLS networks to route traffic trunks away from possibly congested resources or toward possibly more favorable routes.
  - c. Re-configuring the logical topology of the network to make it correlate more closely with the spatial traffic distribution using, for example, an underlying path-oriented technology such as MPLS LSPs or optical channel trails.

Many of these adaptive schemes rely on measurement systems. A measurement system monitors changes in traffic distribution, traffic loads, and network resource utilization and then provides feedback to the online or offline traffic engineering mechanisms and tools so that they can trigger control actions within the network. The traffic engineering mechanisms and tools can be implemented in a distributed or centralized fashion. A centralized scheme may have global visibility into

the network state and may produce more optimal solutions. However, centralized schemes are prone to single points of failure and may not scale as well as distributed schemes. Moreover, the information utilized by a centralized scheme may be stale and might not reflect the actual state of the network. It is not an objective of this document to make a recommendation between distributed and centralized schemes: that is a choice that network administrators must make based on their specific needs.

- \* Short (picoseconds to minutes): This category includes packet level processing functions and events that are recorded on the order of several round trip times. It also includes router mechanisms such as passive and active buffer management. All of these mechanisms are used to control congestion or signal congestion to end systems so that they can adaptively regulate the rate at which traffic is injected into the network. One of the most popular active queue management schemes, especially for TCP traffic, is Random Early Detection (RED) [FLJA93]. During congestion (but before the queue is filled), the RED scheme chooses arriving packets to "mark" according to a probabilistic algorithm which takes into account the average queue size. A router that does not utilize explicit congestion notification (ECN) [FLOY94] can simply drop marked packets to alleviate congestion and implicitly notify the receiver about the congestion. On the other hand, if the router supports ECN, it can set the ECN field in the packet header. Several variations of RED have been proposed to support different drop precedence levels in multi-class environments [RFC2597]. RED provides congestion avoidance which is not worse than traditional Tail-Drop (TD) queue management (drop arriving packets only when the queue is full). Importantly, RED reduces the possibility of global synchronization where retransmission burst become synchronized across the whole network, and improves fairness among different TCP sessions. However, RED by itself cannot prevent congestion and unfairness caused by sources unresponsive to RED, e.g., UDP traffic and some misbehaved greedy connections. Other schemes have been proposed to improve the performance and fairness in the presence of unresponsive traffic. Some of those schemes (such as Longest Queue Drop (LQD) and Dynamic Soft Partitioning with Random Drop (RND) [SLDC98]) were proposed as theoretical frameworks and are typically not available in existing commercial products.

## 2. Reactive Versus Preventive Congestion Management Schemes

- \* Reactive (recovery) congestion management policies react to existing congestion problems. All the policies described above for the long and medium time scales can be categorized as being reactive. They are based on monitoring and identifying congestion problems that exist in the network, and on the initiation of relevant actions to ease a situation.
- \* Preventive (predictive/avoidance) policies take proactive action to prevent congestion based on estimates and predictions of future congestion problems (e.g., traffic matrix forecasts). Some of the policies described for the long and medium time scales fall into this category. Preventive policies do not necessarily respond immediately to existing congestion problems. Instead, forecasts of traffic demand and workload distribution are considered, and action may be taken to prevent potential future congestion problems. The schemes described for the short time scale can also be used for congestion avoidance because dropping or marking packets before queues actually overflow would trigger corresponding TCP sources to slow down.

### 3. Supply-Side Versus Demand-Side Congestion Management Schemes

- \* Supply-side congestion management policies increase the effective capacity available to traffic in order to control or reduce congestion. This can be accomplished by increasing capacity or by balancing distribution of traffic over the network. Capacity planning aims to provide a physical topology and associated link bandwidths that match or exceed estimated traffic workload and traffic distribution subject to traffic forecasts and budgetary or other constraints. If the actual traffic distribution does not fit the topology derived from capacity planning, then the traffic can be mapped onto the topology by using routing control mechanisms, by applying path oriented technologies (e.g., MPLS LSPs and optical channel trails) to modify the logical topology, or by employing some other load redistribution mechanisms.
- \* Demand-side congestion management policies control or regulate the offered traffic to alleviate congestion problems. For example, some of the short time scale mechanisms described earlier as well as policing and rate-shaping mechanisms attempt to regulate the offered load in various ways.



## 2.5. Implementation and Operational Context

The operational context of Internet traffic engineering is characterized by constant changes that occur at multiple levels of abstraction. The implementation context demands effective planning, organization, and execution. The planning aspects may involve determining prior sets of actions to achieve desired objectives. Organizing involves arranging and assigning responsibility to the various components of the traffic engineering system and coordinating the activities to accomplish the desired TE objectives. Execution involves measuring and applying corrective or perfective actions to attain and maintain desired TE goals.

## 3. Traffic Engineering Process Models

This section describes a generic process model that captures the high-level practical aspects of Internet traffic engineering in an operational context. The process model is described as a sequence of actions that must be carried out to optimize the performance of an operational network (see also [RFC2702], [AWD2]). This process model may be enacted explicitly or implicitly, by a software process or by a human.

The traffic engineering process model is iterative [AWD2]. The four phases of the process model described below are repeated as a continual sequence.

- o Define the relevant control policies that govern the operation of the network.
- o Acquire measurement data from the operational network.
- o Analyze the network state and characterize the traffic workload. Proactive analysis identifies potential problems that could manifest in the future. Reactive analysis identifies existing problems and determines their causes.
- o Optimize the performance of the network. This involves a decision process which selects and implements a set of actions from a set of alternatives given the results of the three previous steps. Optimization actions may include the use of techniques to control the offered traffic and to control the distribution of traffic across the network.

### 3.1. Components of the Traffic Engineering Process Model

The key components of the traffic engineering process model are as follows.

1. Measurement is crucial to the traffic engineering function. The operational state of a network can only be conclusively determined through measurement. Measurement is also critical to the optimization function because it provides feedback data which is used by traffic engineering control subsystems. This data is used to adaptively optimize network performance in response to events and stimuli originating within and outside the network. Measurement in support of the TE function can occur at different levels of abstraction. For example, measurement can be used to derive packet level characteristics, flow level characteristics, user or customer level characteristics, traffic aggregate characteristics, component level characteristics, and network wide characteristics.
2. Modeling, analysis, and simulation are important aspects of Internet traffic engineering. Modeling involves constructing an abstract or physical representation which depicts relevant traffic characteristics and network attributes. A network model is an abstract representation of the network which captures relevant network features, attributes, and characteristic. Network simulation tools are extremely useful for traffic engineering. Because of the complexity of realistic quantitative analysis of network behavior, certain aspects of network performance studies can only be conducted effectively using simulation.
3. Network performance optimization involves resolving network issues by transforming such issues into concepts that enable a solution, identification of a solution, and implementation of the solution. Network performance optimization can be corrective or perfective. In corrective optimization, the goal is to remedy a problem that has occurred or that is incipient. In perfective optimization, the goal is to improve network performance even when explicit problems do not exist and are not anticipated.

### 4. Taxonomy of Traffic Engineering Systems

This section presents a short taxonomy of traffic engineering systems constructed based on traffic engineering styles and views as listed below and described in greater detail in the following subsections of this document.

- o Time-dependent versus State-dependent versus Event-dependent

- o Offline versus Online
- o Centralized versus Distributed
- o Local versus Global Information
- o Prescriptive versus Descriptive
- o Open Loop versus Closed Loop
- o Tactical versus Strategic

#### 4.1. Time-Dependent Versus State-Dependent Versus Event-Dependent

Traffic engineering methodologies can be classified as time-dependent, state-dependent, or event-dependent. All TE schemes are considered to be dynamic in this document. Static TE implies that no traffic engineering methodology or algorithm is being applied - it is a feature of network planning, but lacks the reactive and flexible nature of traffic engineering.

In time-dependent TE, historical information based on periodic variations in traffic (such as time of day) is used to pre-program routing and other TE control mechanisms. Additionally, customer subscription or traffic projection may be used. Pre-programmed routing plans typically change on a relatively long time scale (e.g., daily). Time-dependent algorithms do not attempt to adapt to short-term variations in traffic or changing network conditions. An example of a time-dependent algorithm is a global centralized optimizer where the input to the system is a traffic matrix and multi-class QoS requirements as described [MR99]. Another example of such a methodology is the application of data mining to Internet traffic [AJ19] which enables the use of various machine learning algorithms to identify patterns within historically collected datasets about Internet traffic, and to extract information in order to guide decision-making, and to improve efficiency and productivity of operational processes.

State-dependent TE adapts the routing plans based on the current state of the network which provides additional information on variations in actual traffic (i.e., perturbations from regular variations) that could not be predicted using historical information. Constraint-based routing is an example of state-dependent TE operating in a relatively long time scale. An example operating in a relatively short timescale is a load-balancing algorithm described in [MATE]. The state of the network can be based on parameters flooded by the routers. Another approach is for a particular router performing adaptive TE to send probe packets along a path to gather

the state of that path. [RFC6374] defines protocol extensions to collect performance measurements from MPLS networks. Another approach is for a management system to gather the relevant information directly from network elements using telemetry data collection "publication/subscription" techniques [RFC7923]. Timely gathering and distribution of state information is critical for adaptive TE. While time-dependent algorithms are suitable for predictable traffic variations, state-dependent algorithms may be applied to increase network efficiency and resilience to adapt to the prevailing network state.

Event-dependent TE methods can also be used for TE path selection. Event-dependent TE methods are distinct from time-dependent and state-dependent TE methods in the manner in which paths are selected. These algorithms are adaptive and distributed in nature and typically use learning models to find good paths for TE in a network. While state-dependent TE models typically use available-link-bandwidth (ALB) flooding for TE path selection, event-dependent TE methods do not require ALB flooding. Rather, event-dependent TE methods typically search out capacity by learning models, as in the success-to-the-top (STT) method. ALB flooding can be resource intensive, since it requires link bandwidth to carry LSAs, processor capacity to process LSAs, and the overhead can limit area/Autonomous System (AS) size. Modeling results suggest that event-dependent TE methods could lead to a reduction in ALB flooding overhead without loss of network throughput performance [I-D.ietf-tewg-qos-routing].

#### 4.2. Offline Versus Online

Traffic engineering requires the computation of routing plans. The computation may be performed offline or online. The computation can be done offline for scenarios where routing plans need not be executed in real-time. For example, routing plans computed from forecast information may be computed offline. Typically, offline computation is also used to perform extensive searches on multi-dimensional solution spaces.

Online computation is required when the routing plans must adapt to changing network conditions as in state-dependent algorithms. Unlike offline computation (which can be computationally demanding), online computation is geared toward relative simple and fast calculations to select routes, fine-tune the allocations of resources, and perform load balancing.

#### 4.3. Centralized Versus Distributed

Under centralized control there is a central authority which determines routing plans and perhaps other TE control parameters on behalf of each router. The central authority periodically collects network-state information from all routers, and sends routing information to the routers. The update cycle for information exchange in both directions is a critical parameter directly impacting the performance of the network being controlled. Centralized control may need high processing power and high bandwidth control channels.

Distributed control determines route selection by each router autonomously based on the router's view of the state of the network. The network state information may be obtained by the router using a probing method or distributed by other routers on a periodic basis using link state advertisements. Network state information may also be disseminated under exception conditions. Examples of protocol extensions used to advertise network link state information are defined in [RFC5305], [RFC6119], [RFC7471], [RFC8570], and [RFC8571]. See also Section 5.1.11.

##### 4.3.1. Hybrid Systems

In practice, most TE systems will be a hybrid of central and distributed control. For example, a popular MPLS approach to TE is to use a central controller based on an active, stateful PCE, but to use routing and signaling protocols to make local decisions at routers within the network. Local decisions may be able to respond more quickly to network events, but may result in conflicts with decisions made by other routers.

Network operations for TE systems may also use a hybrid of offline and online computation. TE paths may be precomputed based on stable-state network information and planned traffic demands, but may then be modified in the active network depending on variations in network state and traffic load. Furthermore, responses to network events may be precomputed offline to allow rapid reactions without further computation, or may be derived online depending on the nature of the events.

Lastly, note that a fully functional TE system is likely to use all aspects of time-dependent, state-dependent, and event-dependent methodologies as described in Section 4.1.

#### 4.3.2. Considerations for Software Defined Networking

As discussed in Section 5.1.17, one of the main drivers for SDN is a decoupling of the network control plane from the data plane [RFC7149]. However, SDN may also combine centralized control of resources, and facilitate application-to-network interaction via an application programming interface (API) such as [RFC8040]. Combining these features provides a flexible network architecture that can adapt to network requirements of a variety of higher-layer applications, a concept often referred to as the "programmable network" [RFC7426].

The centralized control aspect of SDN helps improve global network resource utilization compared with distributed network control, where local policy may often override global optimization goals. In an SDN environment, the data plane forwards traffic to its desired destination. However, before traffic reaches the data plane, the logically centralized SDN control plane often determines the end-to-end path the application traffic will take in the network. Therefore, the SDN control plane needs to be aware of the underlying network topology, capabilities and current node and link resource state.

Using a PCE-based SDN control framework [RFC7491], the available network topology may be discovered by running a passive instance of OSPF or IS-IS, or via BGP-LS [RFC7752], to generate a TED (see Section 5.1.20). The PCE is used to compute a path (see Section 5.1.13) based on the TED and available bandwidth, and further path optimization may be based on requested objective functions [RFC5541]. When a suitable path has been computed the programming of the explicit network path may be performed using either end-to-end signaling protocol [RFC3209] or per-hop with each node being directly programmed [RFC8283] by the SDN controller.

By utilizing a centralized approach to network control, additional network benefits are also available, including Global Concurrent Optimization (GCO) [RFC5557]. A GCO path computation request will simultaneously use the network topology and set of new end-to-end path requests, along with their respective constraints, for optimal placement in the network. Correspondingly, a GCO-based computation may be applied to recompute existing network paths to groom traffic and to mitigate congestion.

#### 4.4. Local Versus Global

Traffic engineering algorithms may require local and global network-state information.

Local information is the state of a portion of the domain. Examples include the bandwidth and packet loss rate of a particular path, or the state and capabilities of a network link. Local state information may be sufficient for certain instances of distributed control TE.

Global information is the state of the entire TE domain. Examples include a global traffic matrix, and loading information on each link throughout the domain of interest. Global state information is typically required with centralized control. Distributed TE systems may also need global information in some cases.

#### 4.5. Prescriptive Versus Descriptive

TE systems may also be classified as prescriptive or descriptive.

Prescriptive traffic engineering evaluates alternatives and recommends a course of action. Prescriptive traffic engineering can be further categorized as either corrective or perfective. Corrective TE prescribes a course of action to address an existing or predicted anomaly. Perfective TE prescribes a course of action to evolve and improve network performance even when no anomalies are evident.

Descriptive traffic engineering, on the other hand, characterizes the state of the network and assesses the impact of various policies without recommending any particular course of action.

##### 4.5.1. Intent-Based Networking

One way to express a service request is through "intent". Intent-Based Networking aims to produce networks that are simpler to manage and operate, requiring only minimal intervention. Intent is defined in [I-D.irtf-nmrg-ibn-concepts-definitions] as a set of operational goals (that a network should meet) and outcomes (that a network is supposed to deliver), defined in a declarative manner without specifying how to achieve or implement them.

Intent provides data and functional abstraction so that users and operators do not need to be concerned with low-level device configuration or the mechanisms used to achieve a given intent. This approach can be conceptually easier for a user, but may be less expressive in terms of constraints and guidelines.

Intent-Based Networking is applicable to traffic engineering because many of the high-level objectives may be expressed as "intent." For example, load balancing, delivery of services, and robustness against

failures. The intent is converted by the management system into traffic engineering actions within the network.

#### 4.6. Open-Loop Versus Closed-Loop

Open-loop traffic engineering control is where control action does not use feedback information from the current network state. The control action may use its own local information for accounting purposes, however.

Closed-loop traffic engineering control is where control action utilizes feedback information from the network state. The feedback information may be in the form of historical information or current measurement.

#### 4.7. Tactical versus Strategic

Tactical traffic engineering aims to address specific performance problems (such as hot-spots) that occur in the network from a tactical perspective, without consideration of overall strategic imperatives. Without proper planning and insights, tactical TE tends to be ad hoc in nature.

Strategic traffic engineering approaches the TE problem from a more organized and systematic perspective, taking into consideration the immediate and longer term consequences of specific policies and actions.

### 5. Review of TE Techniques

This section briefly reviews different traffic engineering approaches proposed and implemented in telecommunications and computer networks using IETF protocols and architectures. The discussion is not intended to be comprehensive. It is primarily intended to illuminate existing approaches to traffic engineering in the Internet. A historic overview of traffic engineering in telecommunications networks is provided in Appendix A, while Appendix B describes approaches in other standards bodies.

#### 5.1. Overview of IETF Projects Related to Traffic Engineering

This subsection reviews a number of IETF activities pertinent to Internet traffic engineering.



#### 5.1.1.1. Constraint-Based Routing

Constraint-based routing refers to a class of routing systems that compute routes through a network subject to the satisfaction of a set of constraints and requirements. In the most general case, constraint-based routing may also seek to optimize overall network performance while minimizing costs.

The constraints and requirements may be imposed by the network itself or by administrative policies. Constraints may include bandwidth, hop count, delay, and policy instruments such as resource class attributes. Constraints may also include domain specific attributes of certain network technologies and contexts which impose restrictions on the solution space of the routing function. Path oriented technologies such as MPLS have made constraint-based routing feasible and attractive in public IP networks.

The concept of constraint-based routing within the context of MPLS traffic engineering requirements in IP networks was first described in [RFC2702] and led to developments such as MPLS-TE [RFC3209] as described in Section 5.1.6.

Unlike QoS-based routing (for example, see [RFC2386], [MA], and [I-D.ietf-idr-performance-routing]) which generally addresses the issue of routing individual traffic flows to satisfy prescribed flow-based QoS requirements subject to network resource availability, constraint-based routing is applicable to traffic aggregates as well as flows and may be subject to a wide variety of constraints which may include policy restrictions.

##### 5.1.1.1.1. IGP Flexible Algorithms (Flex-Algos)

The traditional approach to routing in an IGP network relies on the IGP's deriving "shortest paths" over the network based solely on the IGP metric assigned to the links. Such an approach is often limited: traffic may tend to converge toward the destination, possibly causing congestion; and it is not possible to steer traffic onto paths depending on the end-to-end qualities demanded by the applications.

To overcome this limitation, various sorts of traffic engineering have been widely deployed (as described in this document), where the TE component is responsible for computing the path based on additional metrics and/or constraints. Such paths (or tunnels) need to be installed in the routers' forwarding tables in addition to, or as a replacement for the original paths computed by IGP's. The main drawback of these TE approaches is the additional complexity of protocols and management, and the state that may need to be maintained within the network.

IGP flexible algorithms (flex-algos) [I-D.ietf-lsr-flex-algo] allow IGPs to construct constraint-based paths over the network by computing constraint-based next hops. The intent of flex-algos is to reduce TE complexity by letting an IGP perform some basic TE computation capabilities. Flex-algo includes a set of extensions to the IGPs that enable a router to send TLVs that:

- o describe a set of constraints on the topology
- o identify calculation-type
- o describe a metric-type that is to be used to compute the best paths through the constrained topology.

A given combination of calculation-type, metric-type, and constraints is known as a "Flexible Algorithm Definition" (or FAD). A router that sends such a set of TLVs also assigns a specific identifier (the Flexible Algorithm) to the specified combination of calculation-type, metric-type, and constraints.

There are two use cases for flex-algo: in IP networks [I-D.ietf-lsr-ip-flexalgo] and in segment routing networks [I-D.ietf-lsr-flex-algo]. In the first case, flex-algo computes paths to an IPv4 or IPv6 address, in the second case, flex-algo computes paths to a prefix SID (see Section 5.1.15).

There are many use cases where flex-algo can bring big value, such as:

- o Expansion of functionality of IP Performance metrics [RFC5664] when points of interest could instantiate specific constraint-based routing (flex-algo) based on the measurement results.
- o Nested usage of flex-algo and TE extensions for IGP (see Section 5.1.11) when we can form 'underlay' by means of flex-algo and 'overlay' by TE.
- o Flex-algo in SR-MPLS (Section 5.1.15) is a base use case when we can easily benefit from TE-like topology that will be built without external TE component on routers or PCE (see Section 5.1.13).
- o Building of network slices [I-D.ietf-teas-ietf-network-slices] where particular IETF network slice SLO can be guaranteed by flex-algo.

### 5.1.2. Integrated Services

The IETF developed the Integrated Services (Intserv) model that requires resources, such as bandwidth and buffers, to be reserved a priori for a given traffic flow to ensure that the quality of service requested by the traffic flow is satisfied. The Integrated Services model includes additional components beyond those used in the best-effort model such as packet classifiers, packet schedulers, and admission control. A packet classifier is used to identify flows that are to receive a certain level of service. A packet scheduler handles the scheduling of service to different packet flows to ensure that QoS commitments are met. Admission control is used to determine whether a router has the necessary resources to accept a new flow.

The main issue with the Integrated Services model has been scalability [RFC2998], especially in large public IP networks which may potentially have millions of active traffic flows in transit concurrently.

A notable feature of the Integrated Services model is that it requires explicit signaling of QoS requirements from end systems to routers [RFC2753]. The Resource Reservation Protocol (RSVP) performs this signaling function and is a critical component of the Integrated Services model. RSVP is described in Section 5.1.3.

### 5.1.3. RSVP

RSVP is a soft state signaling protocol [RFC2205]. It supports receiver initiated establishment of resource reservations for both multicast and unicast flows. RSVP was originally developed as a signaling protocol within the Integrated Services framework (see Section 5.1.2) for applications to communicate QoS requirements to the network and for the network to reserve relevant resources to satisfy the QoS requirements [RFC2205].

In RSVP, the traffic sender or source node sends a PATH message to the traffic receiver with the same source and destination addresses as the traffic which the sender will generate. The PATH message contains: (1) a sender traffic specification describing the characteristics of the traffic, (2) a sender template specifying the format of the traffic, and (3) an optional advertisement specification which is used to support the concept of One Pass With Advertising (OPWA) [RFC2205]. Every intermediate router along the path forwards the PATH message to the next hop determined by the routing protocol. Upon receiving a PATH message, the receiver responds with a RESV message which includes a flow descriptor used to request resource reservations. The RESV message travels to the sender or source node in the opposite direction along the path that

the PATH message traversed. Every intermediate router along the path can reject or accept the reservation request of the RESV message. If the request is rejected, the rejecting router will send an error message to the receiver and the signaling process will terminate. If the request is accepted, link bandwidth and buffer space are allocated for the flow and the related flow state information is installed in the router.

One of the issues with the original RSVP specification was Scalability. This is because reservations were required for micro-flows, so that the amount of state maintained by network elements tends to increase linearly with the number of traffic flows. These issues are described in [RFC2961] which also modifies and extends RSVP to mitigate the scaling problems to make RSVP a versatile signaling protocol for the Internet. For example, RSVP has been extended to reserve resources for aggregation of flows, to set up MPLS explicit label switched paths (see Section 5.1.6), and to perform other signaling functions within the Internet. [RFC2961] also describes a mechanism to reduce the amount of Refresh messages required to maintain established RSVP sessions.

#### 5.1.4. Differentiated Services

The goal of Differentiated Services (Diffserv) within the IETF was to devise scalable mechanisms for categorization of traffic into behavior aggregates, which ultimately allows each behavior aggregate to be treated differently, especially when there is a shortage of resources such as link bandwidth and buffer space [RFC2475]. One of the primary motivations for Diffserv was to devise alternative mechanisms for service differentiation in the Internet that mitigate the scalability issues encountered with the Intserv model.

Diffserv uses the Differentiated Services field in the IP header (the DS field) consisting of six bits in what was formerly known as the Type of Service (TOS) octet. The DS field is used to indicate the forwarding treatment that a packet should receive at a transit node [RFC2474]. Diffserv includes the concept of Per-Hop Behavior (PHB) groups. Using the PHBs, several classes of services can be defined using different classification, policing, shaping, and scheduling rules.

For an end-user of network services to utilize Differentiated Services provided by its Internet Service Provider (ISP), it may be necessary for the user to have an SLA with the ISP. An SLA may explicitly or implicitly specify a Traffic Conditioning Agreement (TCA) which defines classifier rules as well as metering, marking, discarding, and shaping rules.

Packets are classified, and possibly policed and shaped at the ingress to a Diffserv network. When a packet traverses the boundary between different Diffserv domains, the DS field of the packet may be re-marked according to existing agreements between the domains.

Differentiated Services allows only a finite number of service classes to be specified by the DS field. The main advantage of the Diffserv approach relative to the Intserv model is scalability. Resources are allocated on a per-class basis and the amount of state information is proportional to the number of classes rather than to the number of application flows.

The Diffserv model deals with traffic management issues on a per hop basis. The Diffserv control model consists of a collection of micro-TE control mechanisms. Other traffic engineering capabilities, such as capacity management (including routing control), are also required in order to deliver acceptable service quality in Diffserv networks. The concept of Per Domain Behaviors has been introduced to better capture the notion of Differentiated Services across a complete domain [RFC3086].

Diffserv procedures can also be applied in an MPLS context. See Section 6.8 for more information.

#### 5.1.5. QUIC

QUIC [I-D.ietf-quic-transport] is a UDP-based multiplexed and secure transport protocol. QUIC provides applications with flow-controlled streams for structured communication, low-latency connection establishment, and network path migration.

QUIC is a connection-oriented protocol that creates a stateful interaction between a client and server. QUIC uses a handshake procedure that combines negotiation of cryptographic and transport parameters. This is a key differentiation from other transport protocols.

Endpoints communicate in QUIC by exchanging QUIC packets that use a customized framing for protection. Most QUIC packets contain frames, which carry control information and application data between endpoints. QUIC authenticates all packets and encrypts as much as is practical. QUIC packets are carried in UDP datagrams to better facilitate deployment within existing systems and networks.

Application protocols exchange information over a QUIC connection via streams, which are ordered sequences of bytes. Two types of stream can be created: bidirectional streams, which allow both endpoints to send data; and unidirectional streams, which allow a single endpoint

to send data. A credit-based scheme is used to limit stream creation and to bound the amount of data that can be sent.

QUIC provides the necessary feedback to implement reliable delivery and congestion control to avoid network congestion.

#### 5.1.6. Multiprotocol Label Switching (MPLS)

MPLS is an advanced forwarding scheme which also includes extensions to conventional IP control plane protocols. MPLS extends the Internet routing model and enhances packet forwarding and path control [RFC3031].

At the ingress to an MPLS domain, Label Switching Routers (LSRs) classify IP packets into Forwarding Equivalence Classes (FECs) based on a variety of factors, including, e.g., a combination of the information carried in the IP header of the packets and the local routing information maintained by the LSRs. An MPLS label stack entry is then prepended to each packet according to their forwarding equivalence classes. The MPLS label stack entry is 32 bits long and contains a 20-bit label field.

An LSR makes forwarding decisions by using the label prepended to packets as the index into a local next hop label forwarding entry (NHLFE). The packet is then processed as specified in the NHLFE. The incoming label may be replaced by an outgoing label (label swap), and the packet may be forwarded to the next LSR. Before a packet leaves an MPLS domain, its MPLS label may be removed (label pop). A Label Switched Path (LSP) is the path between an ingress LSRs and an egress LSRs through which a labeled packet traverses. The path of an explicit LSP is defined at the originating (ingress) node of the LSP. MPLS can use a signaling protocol such as RSVP or LDP to set up LSPs.

MPLS is a very powerful technology for Internet traffic engineering because it supports explicit LSPs which allow constraint-based routing to be implemented efficiently in IP networks [AWD2]. The requirements for traffic engineering over MPLS are described in [RFC2702]. Extensions to RSVP to support instantiation of explicit LSP are discussed in [RFC3209].

#### 5.1.7. Generalized MPLS (GMPLS)

GMPLS extends MPLS control protocols to encompass time-division (e.g., Synchronous Optical Network / Synchronous Digital Hierarchy (SONET/SDH), Plesiochronous Digital Hierarchy (PDH), Optical Transport Network (OTN)), wavelength ( $\lambda$ s), and spatial switching (e.g., incoming port or fiber to outgoing port or fiber) as well as continuing to support packet switching. GMPLS provides a common set

of control protocols for all of these layers (including some technology-specific extensions) each of which has a diverse data or forwarding plane. GMPLS covers both the signaling and the routing part of that control plane and is based on the Traffic Engineering extensions to MPLS (see Section 5.1.6).

In GMPLS, the original MPLS architecture is extended to include LSRs whose forwarding planes rely on circuit switching, and therefore cannot forward data based on the information carried in either packet or cell headers. Specifically, such LSRs include devices where the switching is based on time slots, wavelengths, or physical ports. These additions impact basic LSP properties: how labels are requested and communicated, the unidirectional nature of MPLS LSPs, how errors are propagated, and information provided for synchronizing the ingress and egress LSRs.

#### 5.1.8. IP Performance Metrics

The IETF IP Performance Metrics (IPPM) working group has developed a set of standard metrics that can be used to monitor the quality, performance, and reliability of Internet services. These metrics can be applied by network operators, end-users, and independent testing groups to provide users and service providers with a common understanding of the performance and reliability of the Internet component 'clouds' they use/provide [RFC2330]. The criteria for performance metrics developed by the IPPM working group are described in [RFC2330]. Examples of performance metrics include one-way packet loss [RFC7680], one-way delay [RFC7679], and connectivity measures between two nodes [RFC2678]. Other metrics include second-order measures of packet loss and delay.

Some of the performance metrics specified by the IPPM working group are useful for specifying SLAs. SLAs are sets of service level objectives negotiated between users and service providers, wherein each objective is a combination of one or more performance metrics, possibly subject to certain constraints.

#### 5.1.9. Flow Measurement

The IETF Real Time Flow Measurement (RTFM) working group produced an architecture that defines a method to specify traffic flows as well as a number of components for flow measurement (meters, meter readers, manager) [RFC2722]. A flow measurement system enables network traffic flows to be measured and analyzed at the flow level for a variety of purposes. As noted in RFC 2722, a flow measurement system can be very useful in the following contexts:

- o understanding the behavior of existing networks

- o planning for network development and expansion
- o quantification of network performance
- o verifying the quality of network service
- o attribution of network usage to users.

A flow measurement system consists of meters, meter readers, and managers. A meter observes packets passing through a measurement point, classifies them into groups, accumulates usage data (such as the number of packets and bytes for each group), and stores the usage data in a flow table. A group may represent any collection of user applications, hosts, networks, etc. A meter reader gathers usage data from various meters so it can be made available for analysis. A manager is responsible for configuring and controlling meters and meter readers. The instructions received by a meter from a manager include flow specifications, meter control parameters, and sampling techniques. The instructions received by a meter reader from a manager include the address of the meter whose data is to be collected, the frequency of data collection, and the types of flows to be collected.

#### 5.1.10. Endpoint Congestion Management

[RFC3124] provides a set of congestion control mechanisms for the use of transport protocols. It also allows the development of mechanisms for unifying congestion control across a subset of an endpoint's active unicast connections (called a congestion group). A congestion manager continuously monitors the state of the path for each congestion group under its control. The manager uses that information to instruct a scheduler on how to partition bandwidth among the connections of that congestion group.

#### 5.1.11. TE Extensions to the IGPs

[RFC5305] describes the extensions to the Intermediate System to Intermediate System (IS-IS) protocol to support TE, similarly [RFC3630] specifies TE extensions for OSPFv2 ([RFC5329] has the same description for OSPFv3).

The idea of redistribution TE extensions such as link type and ID, local and remote IP addresses, TE metric, maximum bandwidth, maximum reservable bandwidth and unreserved bandwidth, admin group in IGP is a common for both IS-IS and OSPF.

The difference is in the details of their transmission: IS-IS uses the Extended IS Reachability TLV (type 22) and Sub-TLVs for those TE



parameters, OSPFv2 uses Opaque LSA [RFC5250] type 10 (OSPFv3 uses Intra-Area-TE-LSA) with two top-level TLV (Router Address and Link) also with Sub-TLVs for that purpose.

IS-IS also uses the Extended IP Reachability TLV (type 135, which have the new 32 bit metric) and the TE Router ID TLV (type 134). Those Sub-TLV details are described in [RFC8570] for IS-IS and in [RFC7471] for OSPFv2 ([RFC5329] for OSPFv3).

#### 5.1.12. Link-State BGP

In a number of environments, a component external to a network is called upon to perform computations based on the network topology and current state of the connections within the network, including traffic engineering information. This is information typically distributed by IGP routing protocols within the network (see Section 5.1.11).

The Border Gateway Protocol (BGP) (see also Section 7) is one of the essential routing protocols that glue the Internet together. BGP Link State (BGP-LS) [RFC7752] is a mechanism by which link-state and traffic engineering information can be collected from networks and shared with external components using the BGP routing protocol. The mechanism is applicable to physical and virtual IGP links, and is subject to policy control.

Information collected by BGP-LS can be used to construct the Traffic Engineering Database (TED, see Section 5.1.20) for use by the Path Computation Element (PCE, see Section 5.1.13), or may be used by Application-Layer Traffic Optimization (ALTO) servers (see Section 5.1.14).

#### 5.1.13. Path Computation Element

Constraint-based path computation is a fundamental building block for traffic engineering in MPLS and GMPLS networks. Path computation in large, multi-domain networks is complex and may require special computational components and cooperation between the elements in different domains. The Path Computation Element (PCE) [RFC4655] is an entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.

Thus, a PCE can provide a central component in a traffic engineering system operating on the Traffic Engineering Database (TED, see Section 5.1.20) with delegated responsibility for determining paths in MPLS, GMPLS, or Segment Routing networks. The PCE uses the Path Computation Element Communication Protocol (PCEP) [RFC5440] to

communicate with Path Computation Clients (PCCs), such as MPLS LSRs, to answer their requests for computed paths or to instruct them to initiate new paths [RFC8281] and maintain state about paths already installed in the network [RFC8231].

PCEs form key components of a number of traffic engineering systems. More information about the applicability of PCE can be found in [RFC8051], while [RFC6805] describes the application of PCE to determining paths across multiple domains. PCE also has potential use in Abstraction and Control of TE Networks (ACTN) (see Section 5.1.17), Centralized Network Control [RFC8283], and Software Defined Networking (SDN) (see Section 4.3.2).

#### 5.1.14. Application-Layer Traffic Optimization

This document describes various TE mechanisms available in the network. However, distributed applications in general and, in particular, bandwidth-greedy P2P applications that are used, for example, for file sharing, cannot directly use those techniques. As per [RFC5693], applications could greatly improve traffic distribution and quality by cooperating with external services that are aware of the network topology. Addressing the Application-Layer Traffic Optimization (ALTO) problem means, on the one hand, deploying an ALTO service to provide applications with information regarding the underlying network (e.g., basic network location structure and preferences of network paths) and, on the other hand, enhancing applications in order to use such information to perform better-than-random selection of the endpoints with which they establish connections.

The basic function of ALTO is based on abstract maps of a network. These maps provide a simplified view, yet enough information about a network for applications to effectively utilize them. Additional services are built on top of the maps. [RFC7285] describes a protocol implementing the ALTO services as an information-publishing interface that allows a network to publish its network information such as network locations, costs between them at configurable granularities, and end-host properties to network applications. The information published by the ALTO Protocol should benefit both the network and the applications. The ALTO Protocol uses a REST-ful design and encodes its requests and responses using JSON [RFC8259] with a modular design by dividing ALTO information publication into multiple ALTO services (e.g., the Map service, the Map-Filtering Service, the Endpoint Property Service, and the Endpoint Cost Service).

[RFC8189] defines a new service that allows an ALTO Client to retrieve several cost metrics in a single request for an ALTO

filtered cost map and endpoint cost map. [RFC8896] extends the ALTO cost information service so that applications decide not only 'where' to connect, but also 'when'. This is useful for applications that need to perform bulk data transfer and would like to schedule these transfers during an off-peak hour, for example.

[I-D.ietf-alto-performance-metrics] introducing network performance metrics, including network delay, jitter, packet loss rate, hop count, and bandwidth. The ALTO server may derive and aggregate such performance metrics from BGP-LS (see Section 5.1.12) or IGP-TE (see Section 5.1.11), or management tools, and then expose the information to allow applications to determine 'where' to connect based on network performance criteria. ALTO WG is evaluating the use of network TE properties while making application decisions for new use-cases such as Edge computing and Datacenter interconnect.

#### 5.1.15. Segment Routing with MPLS Encapsulation (SR-MPLS)

Segment Routing (SR) [RFC8402] leverages the source routing and tunneling paradigms. The path a packet takes is defined at the ingress and the packet is tunneled to the egress. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with an SR header: a label stack in MPLS case.

A segment can represent any instruction, topological or service-based, thanks to the MPLS architecture [RFC3031]. Labels can be looked up in a global context (platform wide) as well as in some other context (see "context labels" in Section 3 of [RFC5331]).

Segments are identified by Segment Identifiers (SIDs). There are four types of SID that are relevant for traffic engineering.

Prefix SID: Unique within the routing domain used to identify a prefix.

Node SID: A Prefix SID with the 'N' bit set to identify a node.

Adjacency SID: Identifies a unidirectional adjacency.

Binding SID: A Binding SID has two purposes:

1. Used to advertise the mappings of prefixes to SIDs/Labels.
2. Used to advertise a path available for a Forwarding Equivalence Class.

#### 5.1.16. Segment Routing Policy

SR Policy [I-D.ietf-spring-segment-routing-policy] is an evolution of Segment Routing to enhance the TE capabilities. It is a framework that enables instantiation of an ordered list of segments on a node for implementing a source routing policy with a specific intent for traffic steering from that node.

An SR Policy is identified through the tuple <headend, color, endpoint>. The headend is the IP address of the node where the policy is instantiated. The endpoint is the IP address of the destination of the policy. The color is an index that associates the SR Policy with an intent (e.g., low-latency).

The headend node is notified of SR Policies and associated SR paths via configuration or by extensions to protocols such as PCEP [RFC8664] or BGP [I-D.ietf-idr-segment-routing-te-policy]. Each SR path consists of a Segment-List (an SR source-routed path), and the headend uses the endpoint and color parameters to classify packets to match the SR policy and so determine along which path to forward them. If an SR Policy is associated with a set of SR paths, each is associated with a weight for weighted load balancing. Furthermore, multiple SR Policies may be associated with a set of SR paths to allow multiple traffic flows to be placed on the same paths.

An SR Binding SID (BSID) are also be associated with each candidate path associated with an SR Policy, or with the SR Policy itself. The headend node installs a BSID-keyed entry in the forwarding plane and assigns it the action of steering packets that match the entry to the selected path of the SR Policy. This steering can be done in various ways:

- o SID Steering: Incoming packets have an active SID matching a local BSID at the headend.
- o Per-destination Steering: Incoming packets match a BGP/Service route which indicates an SR Policy.
- o Per-flow Steering: Incoming packets match a forwarding array (for example, the classic 5-tuple) which indicates an SR Policies.
- o Policy-based Steering: Incoming packets match a routing policy which directs them to an SR Policy.

#### 5.1.17. Network Virtualization and Abstraction

One of the main drivers for Software Defined Networking (SDN) [RFC7149] is a decoupling of the network control plane from the data plane. This separation has been achieved for TE networks with the development of MPLS/GMPLS (see Section 5.1.6 and Section 5.1.7) and the Path Computation Element (PCE) (Section 5.1.13). One of the advantages of SDN is its logically centralized control regime that allows a global view of the underlying networks. Centralized control in SDN helps improve network resource utilization compared with distributed network control.

Abstraction and Control of TE Networks (ACTN) [RFC8453] defines a hierarchical SDN architecture which describes the functional entities and methods for the coordination of resources across multiple domains, to provide end-to-end traffic engineered services. ACTN facilitates end-to-end connections and provides them to the user. ACTN is focused on:

- o Abstraction of the underlying network resources and how they are provided to higher-layer applications and customers.
- o Virtualization of underlying resources for use by the customer, application, or service. The creation of a virtualized environment allows operators to view and control multi-domain networks as a single virtualized network.
- o Presentation to customers of networks as a virtual network via open and programmable interfaces.

The ACTN managed infrastructure is built from traffic engineered network resources, which may include statistical packet bandwidth, physical forwarding plane sources (such as wavelengths and time slots), forwarding and cross-connect capabilities. The type of network virtualization seen in ACTN allows customers and applications (tenants) to utilize and independently control allocated virtual network resources as if resources as if they were physically their own resource. The ACTN network is "sliced", with tenants being given a different partial and abstracted topology view of the physical underlying network.

#### 5.1.18. Network Slicing

An IETF Network Slice is a logical network topology connecting a number of endpoints using a set of shared or dedicated network resources [I-D.ietf-teas-ietf-network-slices]. The resources are used to satisfy specific Service Level Objectives (SLOs) specified by the consumer.

IETF network slices are not, of themselves, TE constructs. However, a network operator that offers IETF network slices is likely to use many TE tools in order to manage their network and provide the services.

IETF Network Slices are defined such that they are independent of the underlying infrastructure connectivity and technologies used. From a customer's perspective an IETF Network Slice looks like a VPN connectivity matrix with additional information about the level of service required between endpoints. From an operator's perspective the IETF Network Slice looks like a set of routing or tunneling instructions with the network resource reservations necessary to provide the required service levels as specified by the SLOs. The concept of an IETF network slice is consistent with an enhanced VPN (VPN+) [I-D.ietf-teas-enhanced-vpn].

#### 5.1.19. Deterministic Networking

Deterministic Networking (DetNet) [RFC8655] is an architecture for applications with critical timing and reliability requirements. The layered architecture particularly focuses on developing DetNet service capabilities in the data plane [RFC8938]. The DetNet service sub-layer provides a set of Packet Replication, Elimination, and Ordering Functions (PREOF) functions to provide end-to-end service assurance. The DetNet forwarding sub-layer provides corresponding forwarding assurance (low packet loss, bounded latency, and in-order delivery) functions using resource allocations and explicit route mechanisms.

The separation into two sub-layers allows a greater flexibility to adapt Detnet capability over a number of TE data plane mechanisms such as IP, MPLS, and Segment Routing. More importantly it interconnects IEEE 802.1 Time Sensitive Networking (TSN) [I-D.ietf-detnet-ip-over-tsn] deployed in Industry Control and Automation Systems (ICAS).

DetNet can be seen as a specialized branch of TE, since it sets up explicit optimized paths with allocation of resources as requested. A DetNet application can express its QoS attributes or traffic behavior using any combination of DetNet functions described in sub-layers. They are then distributed and provisioned using well-established control and provisioning mechanisms adopted for traffic-engineering.

In DetNet, a considerable state information is required to maintain per flow queuing disciplines and resource reservation for a large number of individual flows. This can be quite challenging for network operations during network events such as faults, change in

traffic volume or re-provisioning. Therefore, DetNet recommends support for aggregated flows, however, it still requires large amount of control signaling to establish and maintain DetNet flows.

#### 5.1.20. Network TE State Definition and Presentation

The network states that are relevant to the traffic engineering need to be stored in the system and presented to the user. The Traffic Engineering Database (TED) is a collection of all TE information about all TE nodes and TE links in the network, which is an essential component of a TE system, such as MPLS-TE [RFC2702] and GMPLS [RFC3945]. In order to formally define the data in the TED and to present the data to the user with high usability, the data modeling language YANG [RFC7950] can be used as described in [RFC8795].

#### 5.1.21. System Management and Control Interfaces

The traffic engineering control system needs to have a management interface that is human-friendly and a control interfaces that is programmable for automation. The Network Configuration Protocol (NETCONF) [RFC6241] or the RESTCONF Protocol [RFC8040] provide programmable interfaces that are also human-friendly. These protocols use XML or JSON encoded messages. When message compactness or protocol bandwidth consumption needs to be optimized for the control interface, other protocols, such as Group Communication for the Constrained Application Protocol (CoAP) [RFC7390] or gRPC, are available, especially when the protocol messages are encoded in a binary format. Along with any of these protocols, the data modeling language YANG [RFC7950] can be used to formally and precisely define the interface data.

The Path Computation Element Communication Protocol (PCEP) [RFC5440] is another protocol that has evolved to be an option for the TE system control interface. The messages of PCEP are TLV-based, not defined by a data modeling language such as YANG.

#### 5.2. Content Distribution

The Internet is dominated by client-server interactions, principally Web traffic although in the future, more sophisticated media servers may become dominant. The location and performance of major information servers has a significant impact on the traffic patterns within the Internet as well as on the perception of service quality by end users.

A number of dynamic load balancing techniques have been devised to improve the performance of replicated information servers. These techniques can cause spatial traffic characteristics to become more

dynamic in the Internet because information servers can be dynamically picked based upon the location of the clients, the location of the servers, the relative utilization of the servers, the relative performance of different networks, and the relative performance of different parts of a network. This process of assignment of distributed servers to clients is called traffic directing. It is an application layer function.

Traffic directing schemes that allocate servers in multiple geographically dispersed locations to clients may require empirical network performance statistics to make more effective decisions. In the future, network measurement systems may need to provide this type of information.

When congestion exists in the network, traffic directing and traffic engineering systems should act in a coordinated manner. This topic is for further study.

The issues related to location and replication of information servers, particularly web servers, are important for Internet traffic engineering because these servers contribute a substantial proportion of Internet traffic.

## 6. Recommendations for Internet Traffic Engineering

This section describes high-level recommendations for traffic engineering in the Internet in general terms.

The recommendations describe the capabilities needed to solve a traffic engineering problem or to achieve a traffic engineering objective. Broadly speaking, these recommendations can be categorized as either functional or non-functional recommendations.

- o Functional recommendations describe the functions that a traffic engineering system should perform. These functions are needed to realize traffic engineering objectives by addressing traffic engineering problems.
- o Non-functional recommendations relate to the quality attributes or state characteristics of a traffic engineering system. These recommendations may contain conflicting assertions and may sometimes be difficult to quantify precisely.

### 6.1. Generic Non-functional Recommendations

The generic non-functional recommendations for Internet traffic engineering are listed in the paragraphs that follow. In a given context, some of these recommendations may be critical while others



may be optional. Therefore, prioritization may be required during the development phase of a traffic engineering system to tailor it to a specific operational context.

**Usability:** Usability is a human aspect of traffic engineering systems. It refers to the ease with which a traffic engineering system can be deployed and operated. In general, it is desirable to have a TE system that can be readily deployed in an existing network. It is also desirable to have a TE system that is easy to operate and maintain.

**Automation:** Whenever feasible, a TE system should automate as many TE functions as possible to minimize the amount of human effort needed to analyze and control operational networks. Automation is particularly important in large-scale public networks because of the high cost of the human aspects of network operations and the high risk of network problems caused by human errors. Automation may entail the incorporation of automatic feedback and intelligence into some components of the TE system.

**Scalability:** Public networks continue to grow rapidly with respect to network size and traffic volume. Therefore, to remain applicable as the network evolves, a TE system should be scalable. In particular, a TE system should remain functional as the network expands with regard to the number of routers and links, and with respect to the traffic volume. A TE system should have a scalable architecture, should not adversely impair other functions and processes in a network element, and should not consume too many network resources when collecting and distributing state information, or when exerting control.

**Stability:** Stability is a very important consideration in TE systems that respond to changes in the state of the network. State-dependent TE methodologies typically include a trade-off between responsiveness and stability. It is strongly recommended that when a trade-off between responsiveness and stability is needed, it should be made in favor of stability (especially in public IP backbone networks).

**Flexibility:** A TE system should allow for changes in optimization policy. In particular, a TE system should provide sufficient configuration options so that a network administrator can tailor the system to a particular environment. It may also be desirable to have both online and offline TE subsystems which can be independently enabled and disabled. TE systems that are used in multi-class networks should also have options to support class based performance evaluation and optimization.

**Visibility:** Mechanisms should exist as part of the TE system to collect statistics from the network and to analyze these statistics to determine how well the network is functioning. Derived statistics such as traffic matrices, link utilization, latency, packet loss, and other performance measures of interest which are determined from network measurements can be used as indicators of prevailing network conditions. The capabilities of the various components of the routing system are other examples of status information which should be observable.

**Simplicity:** A TE system should be as simple as possible. Simplicity in user interface does not necessarily imply that the TE system will use naive algorithms. When complex algorithms and internal structures are used, the user interface should hide such complexities from the network administrator as much as possible.

**Interoperability:** Whenever feasible, TE systems and their components should be developed with open standards-based interfaces to allow interoperation with other systems and components.

**Security:** Security is a critical consideration in TE systems. Such systems typically exert control over functional aspects of the network to achieve the desired performance objectives. Therefore, adequate measures must be taken to safeguard the integrity of the TE system. Adequate measures must also be taken to protect the network from vulnerabilities that originate from security breaches and other impairments within the TE system.

The remaining subsections of this section focus on some of the high-level functional recommendations for traffic engineering.

## 6.2. Routing Recommendations

Routing control is a significant aspect of Internet traffic engineering. Routing impacts many of the key performance measures associated with networks, such as throughput, delay, and utilization. Generally, it is very difficult to provide good service quality in a wide area network without effective routing control. A desirable TE routing system is one that takes traffic characteristics and network constraints into account during route selection while maintaining stability.

Shortest path first (SPF) IGPs are based on shortest path algorithms and have limited control capabilities for TE [RFC2702], [AWD2]. These limitations include:

1. Pure SPF protocols do not take network constraints and traffic characteristics into account during route selection. For

example, IGP always select the shortest paths based on link metrics assigned by administrators) so load sharing cannot be performed across paths of different costs. Using shortest paths to forward traffic may cause the following problems:

- \* If traffic from a source to a destination exceeds the capacity of a link along the shortest path, the link (and hence the shortest path) becomes congested while a longer path between these two nodes may be under-utilized
  - \* The shortest paths from different sources can overlap at some links. If the total traffic from the sources exceeds the capacity of any of these links, congestion will occur.
  - \* Problems can also occur because traffic demand changes over time, but network topology and routing configuration cannot be changed as rapidly. This causes the network topology and routing configuration to become sub-optimal over time, which may result in persistent congestion problems.
2. The Equal-Cost Multi-Path (ECMP) capability of SPF IGPs supports sharing of traffic among equal cost paths between two nodes. However, ECMP attempts to divide the traffic as equally as possible among the equal cost shortest paths. Generally, ECMP does not support configurable load sharing ratios among equal cost paths. The result is that one of the paths may carry significantly more traffic than other paths because it may also carry traffic from other sources. This situation can result in congestion along the path that carries more traffic. Weighted ECMP (WECMP) (see, for example, [I-D.ietf-bess-evpn-unequal-lb]) provides some mitigation.
  3. Modifying IGP metrics to control traffic routing tends to have network-wide effects. Consequently, undesirable and unanticipated traffic shifts can be triggered as a result. Work described in Section 8 may be capable of better control [FT00], [FT01].

Because of these limitations, new capabilities are needed to enhance the routing function in IP networks. Some of these capabilities are summarized below.

- o Constraint-based routing computes routes to fulfill requirements subject to constraints. This can be useful in public IP backbones with complex topologies. Constraints may include bandwidth, hop count, delay, and administrative policy instruments such as resource class attributes [RFC2702], [RFC2386]. This makes it possible to select routes that satisfy a given set of

requirements. Routes computed by constraint-based routing are not necessarily the shortest paths. Constraint-based routing works best with path-oriented technologies that support explicit routing, such as MPLS.

Constraint-based routing can also be used as a way to distribute traffic onto the infrastructure, including for best effort traffic. For example, congestion problems caused by uneven traffic distribution may be avoided or reduced by knowing the reservable bandwidth attributes of the network links and by specifying the bandwidth requirements for path selection.

- o A number of enhancements to the link state IGPs are needed to allow them to distribute additional state information required for constraint-based routing. The extensions to OSPF are described in [RFC3630], and to IS-IS in [RFC5305]. Some of the additional topology state information includes link attributes such as reservable bandwidth and link resource class attribute (an administratively specified property of the link). The resource class attribute concept is defined in [RFC2702]. The additional topology state information is carried in new TLVs and sub-TLVs in IS-IS, or in the Opaque LSA in OSPF [RFC5305], [RFC3630].

An enhanced link-state IGP may flood information more frequently than a normal IGP. This is because even without changes in topology, changes in reservable bandwidth or link affinity can trigger the enhanced IGP to initiate flooding. A trade-off between the timeliness of the information flooded and the flooding frequency is typically implemented using a threshold based on the percentage change of the advertised resources to avoid excessive consumption of link bandwidth and computational resources, and to avoid instability in the TED.

- o In a TE system, it is also desirable for the routing subsystem to make the load splitting ratio among multiple paths (with equal cost or different cost) configurable. This capability gives network administrators more flexibility in the control of traffic distribution across the network. It can be very useful for avoiding/relieving congestion in certain situations. Examples can be found in [XIAO] and [I-D.ietf-bess-evpn-unequal-lb].
- o The routing system should also have the capability to control the routes of subsets of traffic without affecting the routes of other traffic if sufficient resources exist for this purpose. This capability allows a more refined control over the distribution of traffic across the network. For example, the ability to move traffic away from its original path to another path (without affecting other traffic paths) allows the traffic to be moved from

resource-poor network segments to resource-rich segments. Path oriented technologies such as MPLS-TE inherently support this capability as discussed in [AWD2].

- o Additionally, the routing subsystem should be able to select different paths for different classes of traffic (or for different traffic behavior aggregates) if the network supports multiple classes of service (different behavior aggregates).

### 6.3. Traffic Mapping Recommendations

Traffic mapping is the assignment of traffic workload onto (pre-established) paths to meet certain requirements. Thus, while constraint-based routing deals with path selection, traffic mapping deals with the assignment of traffic to established paths which may have been generated by constraint-based routing or by some other means. Traffic mapping can be performed by time-dependent or state-dependent mechanisms, as described in Section 4.1.

An important aspect of the traffic mapping function is the ability to establish multiple paths between an originating node and a destination node, and the capability to distribute the traffic between the two nodes across the paths according to some policies. A pre-condition for this scheme is the existence of flexible mechanisms to partition traffic and then assign the traffic partitions onto the parallel paths as noted in [RFC2702]. When traffic is assigned to multiple parallel paths, it is recommended that special care should be taken to ensure proper ordering of packets belonging to the same application (or traffic flow) at the destination node of the parallel paths.

Mechanisms that perform the traffic mapping functions should aim to map the traffic onto the network infrastructure to minimize congestion. If the total traffic load cannot be accommodated, or if the routing and mapping functions cannot react fast enough to changing traffic conditions, then a traffic mapping system may use short time scale congestion control mechanisms (such as queue management, scheduling, etc.) to mitigate congestion. Thus, mechanisms that perform the traffic mapping functions complement existing congestion control mechanisms. In an operational network, traffic should be mapped onto the infrastructure such that intra-class and inter-class resource contention are minimized (see Section 2).

When traffic mapping techniques that depend on dynamic state feedback (e.g., MATE [MATE] and such like) are used, special care must be taken to guarantee network stability.

#### 6.4. Measurement Recommendations

The importance of measurement in traffic engineering has been discussed throughout this document. A TE system should include mechanisms to measure and collect statistics from the network to support the TE function. Additional capabilities may be needed to help in the analysis of the statistics. The actions of these mechanisms should not adversely affect the accuracy and integrity of the statistics collected. The mechanisms for statistical data acquisition should also be able to scale as the network evolves.

Traffic statistics may be classified according to long-term or short-term timescales. Long-term traffic statistics are very useful for traffic engineering. Long-term traffic statistics may periodically record network workload (such as hourly, daily, and weekly variations in traffic profiles) as well as traffic trends. Aspects of the traffic statistics may also describe class of service characteristics for a network supporting multiple classes of service. Analysis of the long-term traffic statistics may yield other information such as busy hour characteristics, traffic growth patterns, persistent congestion problems, hot-spot, and imbalances in link utilization caused by routing anomalies.

A mechanism for constructing traffic matrices for both long-term and short-term traffic statistics should be in place. In multi-service IP networks, the traffic matrices may be constructed for different service classes. Each element of a traffic matrix represents a statistic about the traffic flow between a pair of abstract nodes. An abstract node may represent a router, a collection of routers, or a site in a VPN.

Traffic statistics should provide reasonable and reliable indicators of the current state of the network on the short-term scale. Some short term traffic statistics may reflect link utilization and link congestion status. Examples of congestion indicators include excessive packet delay, packet loss, and high resource utilization. Examples of mechanisms for distributing this kind of information include SNMP, probing tools, FTP, IGP link state advertisements, and NETCONF/RESTCONF, etc.

#### 6.5. Policing, Planning, and Access Control

The recommendations in Section 6.2 and Section 6.3 may be sub-optimal or even ineffective if the amount of traffic flowing on a route or path exceeds the capacity of the resource on that route or path. Several approaches can be used to increase the performance of TE systems.

- o The fundamental approach is some form of planning where traffic is steered onto paths so that it is distributed accross the available resources. This planning may be centralized or distributed, and must be aware of the planned traffic volumes amd available resources. However, this approach is only of value if the traffic is presented conformant to the planned traffic volumes.
- o Traffic flows may be policed at the edges of a network. This is a simple way to check that the actual traffic volumes are consistent with the planned volumes. Some form of measurement (see Section 6.4) is used to determine the rate of arrival of traffic and excess traffic could be discarded. Alternatively, excess traffic could be forwarded as best-effort within the network. However, this approach is only completley effective if the planning is stringent and network-wide, and if a harsh approach is taken to disposing of excess traffic.
- o Resource-based admission control is the process whereby network nodes decide whether to grant access to resources. The basis for the decision on a packet-by-packet basis is determination of the flow to which the packet belongs. This information is combined with policy instructions that have been locally configured, or installed through the management or control planes. The end result is that a packet may be allowed to access (or use) specific resources on the node if and only if the policy is conformed with for the flow to which the packet belongs.

Combining some element of all three of these measures is advisable to achieve a better TE system.

#### 6.6. Network Survivability

Network survivability refers to the capability of a network to maintain service continuity in the presence of faults. This can be accomplished by promptly recovering from network impairments and maintaining the required QoS for existing services after recovery. Survivability is an issue of great concern within the Internet community due to the demand to carry mission critical traffic, real-time traffic, and other high priority traffic over the Internet. Survivability can be addressed at the device level by developing network elements that are more reliable; and at the network level by incorporating redundancy into the architecture, design, and operation of networks. It is recommended that a philosophy of robustness and survivability should be adopted in the architecture, design, and operation of traffic engineering that control IP networks (especially public IP networks). Because different contexts may demand different levels of survivability, the mechanisms developed to support network survivability should be flexible so that they can be tailored to

different needs. A number of tools and techniques have been developed to enable network survivability including MPLS Fast Reroute [RFC4090], RSVP-TE Extensions in Support of End-to-End GMPLS Recovery [RFC4872], and GMPLS Segment Recovery [RFC4873].

The impact of service outages varies significantly for different service classes depending on the duration of the outage which can vary from milliseconds (with minor service impact) to seconds (with possible call drops for IP telephony and session time-outs for connection oriented transactions) to minutes and hours (with potentially considerable social and business impact). Different duration outages have different impacts depending on the throughput of the traffic flows that are interrupted.

Failure protection and restoration capabilities are available in multiple layers as network technologies have continued to evolve. Optical networks are capable of providing dynamic ring and mesh restoration functionality at the wavelength level. At the SONET/SDH layer survivability capability is provided with Automatic Protection Switching (APS) as well as self-healing ring and mesh architectures. Similar functionality is provided by layer 2 technologies such as Ethernet.

Rerouting is used at the IP layer to restore service following link and node outages. Rerouting at the IP layer occurs after a period of routing convergence which may require seconds to minutes to complete. Path-oriented technologies such as MPLS ([RFC3469]) can be used to enhance the survivability of IP networks in a potentially cost effective manner.

An important of multi-layer survivability is that technologies at different layers may provide protection and restoration capabilities at different granularities in terms of time scales and at different bandwidth granularity (from packet-level to wavelength level). Protection and restoration capabilities can also be sensitive to different service classes and different network utility models. Coordinating different protection and restoration capabilities across multiple layers in a cohesive manner to ensure network survivability is maintained at reasonable cost is a challenging task. Protection and restoration coordination across layers may not always be feasible, because networks at different layers may belong to different administrative domains.

The following paragraphs present some of the general recommendations for protection and restoration coordination.

- o Protection and restoration capabilities from different layers should be coordinated to provide network survivability in a



flexible and cost effective manner. Avoiding duplication of functions in different layers is one way to achieve the coordination. Escalation of alarms and other fault indicators from lower to higher layers may also be performed in a coordinated manner. The order of timing of restoration triggers from different layers is another way to coordinate multi-layer protection/restoration.

- o Network capacity reserved in one layer to provide protection and restoration is not available to carry traffic in a higher layer: it is not visible as spare capacity in the higher layer. Placing protection/restoration functions in many layers may increase redundancy and robustness, but it can result in significant inefficiencies in network resource utilization. Careful planning is needed to balance the trade-off between the desire for survivability and the optimal use of resources.
- o It is generally desirable to have protection and restoration schemes that are intrinsically bandwidth efficient.
- o Failure notifications throughout the network should be timely and reliable if they are to be acted on as triggers for effective protection and restoration actions.
- o Alarms and other fault monitoring and reporting capabilities should be provided at the right network layers so that the protection and restoration actions can be taken in those layers.

#### 6.6.1. Survivability in MPLS Based Networks

Because MPLS is path-oriented, it has the potential to provide faster and more predictable protection and restoration capabilities than conventional hop by hop routed IP systems. Protection types for MPLS networks can be divided into four categories.

- o Link Protection: The objective of link protection is to protect an LSP from the failure of a given link. Under link protection, a protection or backup LSP (the secondary LSP) follows a path that is disjoint from the path of the working or operational LSP (the primary LSP) at the particular link where link protection is required. When the protected link fails, traffic on the working LSP is switched to the protection LSP at the head-end of the failed link. As a local repair method, link protection can be fast. This form of protection may be most appropriate in situations where some network elements along a given path are known to be less reliable than others.

- o Node Protection: The objective of node protection is to protect an LSP from the failure of a given node. Under node protection, the secondary LSP follows a path that is disjoint from the path of the primary LSP at the particular node where node protection is required. The secondary LSP is also disjoint from the primary LSP at all links attached to the node to be protected. When the protected node fails, traffic on the working LSP is switched over to the protection LSP at the upstream LSR directly connected to the failed node. Node protection covers a slightly larger part of the network compared to link protection, but is otherwise fundamentally the same.
- o Path Protection: The goal of LSP path protection (or end-to-end protection) is to protect an LSP from any failure along its routed path. Under path protection, the path of the protection LSP is completely disjoint from the path of the working LSP. The advantage of path protection is that the backup LSP protects the working LSP from all possible link and node failures along the path, except for failures of ingress or egress LSR. Additionally, path protection may be more efficient in terms of resource usage than link or node protection applied at every hop along the path. However, path protection may be slower than link and node protection because the fault notifications have to be propagated further.
- o Segment Protection: An MPLS domain may be partitioned into multiple subdomains (protection domains). Path protection is applied to the path of each LSP as it crosses the domain from its ingress to the domain to where it egresses the domain. In cases where an LSP traverses multiple protection domains, a protection mechanism within a domain only needs to protect the segment of the LSP that lies within the domain. Segment protection will generally be faster than end-to-end path protection because recovery generally occurs closer to the fault and the notification doesn't have to propagate as far.

See [RFC3469] and [RFC6372] for a more comprehensive discussion of MPLS based recovery.

#### 6.6.2. Protection Options

Another issue to consider is the concept of protection options. We use notation such as "m:n protection", where m is the number of protection LSPs used to protect n working LSPs. In all cases except 1+1 protection, the resources associated with the protection LSPs can be used to carry preemptable best-effort traffic when the working LSP is functioning correctly.

- o 1:1 protection: One working LSP is protected/restored by one protection LSP.
- o 1:n protection: One protection LSP is used to protect/restore n working LSPs. Only one failed LSP can be restored at any time.
- o n:1 protection: One working LSP is protected/restored by n protection LSPs, possibly with load splitting across the protection LSPs. This may be especially useful when it is not feasible to find one path for the backup that can satisfy the bandwidth requirement of the primary LSP.
- o 1+1 protection: Traffic is sent concurrently on both the working LSP and a protection LSP. The egress LSR selects one of the two LSPs based on local policy (usually based on traffic integrity). When a fault disrupts the traffic on one LSP, the egress switches to receive traffic from the other LSP. This approach is expensive in how it consumes network but recovers from failures most rapidly.

#### 6.7. Multi-Layer Traffic Engineering

Networks are often arranged as layers. A layer relationship may represent the interaction between technologies (for example, an IP network operated over an optical network), or the relationship between different network operators (for example, a customer network operated over a service provider's network). Note that a multi-layer network does not imply the use of multiple technologies, although some form of encapsulation is often applied.

Multi-layer traffic engineering presents a number of challenges associated with scalability and confidentiality. These issues are addressed in [RFC7926] which discusses the sharing of information between domains through policy filters, aggregation, abstraction, and virtualization. That document also discusses how existing protocols can support this scenario with special reference to BGP-LS (see Section 5.1.12).

PCE (see Section 5.1.13) is also a useful tool for multi-layer networks as described in [RFC6805] and [RFC8685]. Signaling techniques for multi-layer traffic engineering are described in [RFC6107].

See also Appendix A.3.1 for a discussion of how the overlay model has been important in the development of traffic engineering.

## 6.8. Traffic Engineering in Diffserv Environments

Increasing requirements to support multiple classes of traffic in the Internet, such as best effort and mission critical data, calls for IP networks to differentiate traffic according to some criteria and to give preferential treatment to certain types of traffic. Large numbers of flows can be aggregated into a few behavior aggregates based on some criteria based on common performance requirements in terms of packet loss ratio, delay, and jitter, or in terms of common fields within the IP packet headers.

Differentiated Services (Diffserv) [RFC2475] can be used to ensure that SLAs defined to differentiate between traffic flows are met. Classes of service (CoS) can be supported in a Diffserv environment by concatenating per-hop behaviors (PHBs) along the routing path. A PHB is the forwarding behavior that a packet receives at a Diffserv-compliant node, and it can be configured at each router. PHBs are delivered using buffer management and packet scheduling mechanisms and require that the ingress nodes use traffic classification, marking, policing, and shaping.

Traffic engineering can complement Diffserv to improve utilization of network resources. Traffic engineering can be operated on an aggregated basis across all service classes [RFC3270], or on a per service class basis. The former is used to provide better distribution of the traffic load over the network resources (see [RFC3270] for detailed mechanisms to support aggregate traffic engineering). The latter case is discussed below since it is specific to the Diffserv environment, with so called Diffserv-aware traffic engineering [RFC4124].

For some Diffserv networks, it may be desirable to control the performance of some service classes by enforcing relationships between the traffic workload contributed by each service class and the amount of network resources allocated or provisioned for that service class. Such relationships between demand and resource allocation can be enforced using a combination of, for example:

- o TE mechanisms on a per service class basis that enforce the relationship between the amount of traffic contributed by a given service class and the resources allocated to that class.
- o Mechanisms that dynamically adjust the resources allocated to a given service class to relate to the amount of traffic contributed by that service class.

It may also be desirable to limit the performance impact of high priority traffic on relatively low priority traffic. This can be

achieved, for example, by controlling the percentage of high priority traffic that is routed through a given link. Another way to accomplish this is to increase link capacities appropriately so that lower priority traffic can still enjoy adequate service quality. When the ratio of traffic workload contributed by different service classes varies significantly from router to router, it may not be enough to rely on conventional IGP routing protocols or on TE mechanisms that are not sensitive to different service classes. Instead, it may be desirable to perform traffic engineering, especially routing control and mapping functions, on a per service class basis. One way to accomplish this in a domain that supports both MPLS and Diffserv is to define class specific LSPs and to map traffic from each class onto one or more LSPs that correspond to that service class. An LSP corresponding to a given service class can then be routed and protected/restored in a class dependent manner, according to specific policies.

Performing traffic engineering on a per class basis may require per-class parameters to be distributed. It is common to have some classes share some aggregate constraints (e.g., maximum bandwidth requirement) without enforcing the constraint on each individual class. These classes can be grouped into class-types, and per-class-type parameters can be distributed to improve scalability. This also allows better bandwidth sharing between classes in the same class-type. A class-type is a set of classes that satisfy the following two conditions:

- o Classes in the same class-type have common aggregate requirements to satisfy required performance levels.
- o There is no requirement to be enforced at the level of an individual class in the class-type. Note that it is, nevertheless, still possible to implement some priority policies for classes in the same class-type to permit preferential access to the class-type bandwidth through the use of preemption priorities.

See [RFC4124] for detailed requirements on Diffserv-aware traffic engineering.

## 6.9. Network Controllability

Offline and online (see Section 4.2) TE considerations are of limited utility if the network cannot be controlled effectively to implement the results of TE decisions and to achieve the desired network performance objectives.

Capacity augmentation is a coarse-grained solution to TE issues. However, it is simple and may be advantageous if bandwidth is abundant and cheap. However, bandwidth is not always abundant and cheap, and additional capacity might not always be the best solution. Adjustments of administrative weights and other parameters associated with routing protocols provide finer-grained control, but this approach is difficult to use and imprecise because of the way the routing protocols interact occur across the network.

Control mechanisms can be manual (e.g., static configuration), partially-automated (e.g., scripts), or fully-automated (e.g., policy based management systems). Automated mechanisms are particularly useful in large scale networks. Multi-vendor interoperability can be facilitated by standardized management systems (e.g., YANG models) to support the control functions required to address TE objectives.

Network control functions should be secure, reliable, and stable as these are often needed to operate correctly in times of network impairments (e.g., during network congestion or security attacks).

## 7. Inter-Domain Considerations

Inter-domain TE is concerned with performance optimization for traffic that originates in one administrative domain and terminates in a different one.

BGP [RFC4271] is the standard exterior gateway protocol used to exchange routing information between autonomous systems (ASes) in the Internet. BGP includes a sequential decision process that calculates the preference for routes to a given destination network. There are two fundamental aspects to inter-domain TE using BGP:

- o Route Redistribution: Controlling the import and export of routes between ASes, and controlling the redistribution of routes between BGP and other protocols within an AS.
- o Best path selection: Selecting the best path when there are multiple candidate paths to a given destination network. This is performed by the BGP decision process, selecting preferred exit points out of an AS towards specific destination networks taking a number of different considerations into account. The BGP path selection process can be influenced by manipulating the attributes associated with the process, including NEXT-HOP, WEIGHT, LOCAL-PREFERENCE, AS-PATH, ROUTE-ORIGIN, MULTI-EXIT-DESCRIMINATOR (MED), IGP METRIC, etc.

Route-maps provide the flexibility to implement complex BGP policies based on pre-configured logical conditions. They can be used to

control import and export policies for incoming and outgoing routes, control the redistribution of routes between BGP and other protocols, and influence the selection of best paths by manipulating the attributes associated with the BGP decision process. Very complex logical expressions that implement various types of policies can be implemented using a combination of Route-maps, BGP-attributes, Access-lists, and Community attributes.

When considering inter-domain TE with BGP, note that the outbound traffic exit point is controllable, whereas the interconnection point where inbound traffic is received typically is not. Therefore, it is up to each individual network to implement TE strategies that deal with the efficient delivery of outbound traffic from its customers to its peering points. The vast majority of TE policy is based on a "closest exit" strategy, which offloads inter-domain traffic at the nearest outbound peering point towards the destination AS. Most methods of manipulating the point at which inbound traffic enters a are either ineffective, or not accepted in the peering community.

Inter-domain TE with BGP is generally effective, but it is usually applied in a trial-and-error fashion because a TE system usually only has a view of the available network resources within one domain (an AS in this case). A systematic approach for inter-domain TE requires cooperation between the domains. Further, what may be considered a good solution in one domain may not necessarily be a good solution in another. Moreover, it is generally considered inadvisable for one domain to permit a control process from another domain to influence the routing and management of traffic in its network.

MPLS TE-tunnels (LSPs) can add a degree of flexibility in the selection of exit points for inter-domain routing by applying the concept of relative and absolute metrics. If BGP attributes are defined such that the BGP decision process depends on IGP metrics to select exit points for inter-domain traffic, then some inter-domain traffic destined to a given peer network can be made to prefer a specific exit point by establishing a TE-tunnel between the router making the selection and the peering point via a TE-tunnel and assigning the TE-tunnel a metric which is smaller than the IGP cost to all other peering points.

Similarly to intra-domain TE, inter-domain TE is best accomplished when a traffic matrix can be derived to depict the volume of traffic from one AS to another.

## 8. Overview of Contemporary TE Practices in Operational IP Networks

This section provides an overview of some traffic engineering practices in IP networks. The focus is on aspects of control of the routing function in operational contexts. The intent here is to provide an overview of the commonly used practices: the discussion is not intended to be exhaustive.

Service providers apply many of the traffic engineering mechanisms described in this document to optimize the performance of their IP networks. These techniques include capacity planning for long timescales; routing control using IGP metrics and MPLS, as well as path planning and path control using MPLS and Segment Routing for medium timescales; and traffic management mechanisms for short timescale.

Capacity planning is an important component of how a service provider plans an effective IP network. These plans may take the following aspects into account: location of and new links or nodes, existing and predicted traffic patterns, costs, link capacity, topology, routing design, and survivability.

Performance optimization of operational networks is usually an ongoing process in which traffic statistics, performance parameters, and fault indicators are continually collected from the network. This empirical data is analyzed and used to trigger TE mechanisms. Tools that perform what-if analysis can also be used to assist the TE process by reviewing scenarios before a new set of configurations are implemented in the operational network.

Real-time intra-domain TE using the IGP is done by increasing the OSPF or IS-IS metric of a congested link until enough traffic has been diverted away from that link. This approach has some limitations as discussed in Section 6.2. Intra-domain TE approaches ([RR94] [FT00] [FT01] [WANG]) take traffic matrix, network topology, and network performance objectives as input, and produce link metrics and load-sharing ratios. These processes open the possibility for intra-domain TE with IGP to be done in a more systematic way.

Administrators of MPLS-TE networks specify and configure link attributes and resource constraints such as maximum reservable bandwidth and resource class attributes for the links in the domain. A link state IGP that supports TE extensions (IS-IS-TE or OSPF-TE) is used to propagate information about network topology and link attributes to all routers in the domain. Network administrators specify the LSPs that are to originate at each router. For each LSP, the network administrator specifies the destination node and the attributes of the LSP which indicate the requirements that are to be



satisfied during the path selection process. The attributes may include an explicit path for the LSP to follow, or the originating router uses a local constraint-based routing process to compute the path of the LSP. RSVP-TE is used as a signaling protocol to instantiate the LSPs. By assigning proper bandwidth values to links and LSPs, congestion caused by uneven traffic distribution can be avoided or mitigated.

The bandwidth attributes of an LSP relate to the bandwidth requirements of traffic that flows through the LSP. The traffic attribute of an LSP can be modified to accommodate persistent shifts in demand (traffic growth or reduction). If network congestion occurs due to some unexpected events, existing LSPs can be rerouted to alleviate the situation or a network administrator can configure new LSPs to divert some traffic to alternative paths. The reservable bandwidth of the congested links can also be reduced to force some LSPs to be rerouted to other paths. A traffic matrix in an MPLS domain can also be estimated by monitoring the traffic on LSPs. Such traffic statistics can be used for a variety of purposes including network planning and network optimization.

Network management and planning systems have evolved and taken over a lot of the responsibility for determining traffic paths in TE networks. This allows a network-wide view of resources, and facilitates coordination of the use of resources for all traffic flows in the network. Initial solutions using a PCE to perform path computation on behalf of network routers have given way to an approach that follows the SDN architecture. A stateful PCE is able to track all of the LSPs in the network and can redistribute them to make better use of the available resources. Such a PCE can form part of a network orchestrator that uses PCEP or some other southbound interface to instruct the signaling protocol or directly program the routers.

Segment routing leverages a centralized TE controller and either an MPLS or IPv6 forwarding plane, but does not need to use a signaling protocol or management plane protocol to reserve resources in the routers. All resource reservation is logical within the controller, and not distributed to the routers. Packets are steered through the network using segment routing.

As mentioned in Section 7, there is usually no direct control over the distribution of inbound traffic to a domain. Therefore, the main goal of inter-domain TE is to optimize the distribution of outbound traffic between multiple inter-domain links. When operating a global network, maintaining the ability to operate the network in a regional fashion where desired, while continuing to take advantage of the benefits of a global network, also becomes an important objective.

Inter-domain TE with BGP begins with the placement of multiple peering interconnection points that are in close proximity to traffic sources/destination, and offer lowest cost paths across the network between the peering points and the sources/destinations. Some location-decision problems that arise in association with inter-domain routing are discussed in [AWD5].

Once the locations of the peering interconnects have been determined and implemented, the network operator decides how best to handle the routes advertised by the peer, as well as how to propagate the peer's routes within their network. One way to engineer outbound traffic flows in a network with many peering interconnects is to create a hierarchy of peers. Generally, the shortest AS paths will be chosen to forward traffic but BGP metrics can be used to prefer some peers and so favor particular paths. Preferred peers are those peers attached through peering interconnects with the most available capacity. Changes may be needed, for example, to deal with a "problem peer" who is difficult to work with on upgrades or is charging high prices for connectivity to their network. In that case, the peer may be given a reduced preference. This type of change can affect a large amount of traffic, and is only used after other methods have failed to provide the desired results.

When there are multiple exit points toward a given peer, and only one of them is congested, it is not necessary to shift traffic away from the peer entirely, but only from the one congested connections. This can be achieved by using passive IGP-metrics, AS-path filtering, or prefix filtering.

## 9. Security Considerations

This document does not introduce new security issues.

Network security is, of course, an important issue. In general, TE mechanisms are security neutral: they may use tunnels which can slightly help protect traffic from inspection and which, in some cases, can be secured using encryption; they put traffic onto predictable paths within the network that may make it easier to find and attack; they increase the complexity of operation and management of the network; and they enable traffic to be steered onto more secure links or to more secure parts of the network.

The consequences of attacks on the control and management protocols used to operate TE networks can be significant: traffic can be hijacked to pass through specific nodes that perform inspection, or even to be delivered to the wrong place; traffic can be steered onto paths that deliver quality that is below the desired quality; and, networks can be congested or have resources on key links consumed.

Thus, it is important to use adequate protection mechanisms on all protocols used to deliver TE.

Certain aspects of a network may be deduced from the details of the TE paths that are used. For example, the link connectivity of the network, and the quality and load on individual links may be assumed from knowing the paths of traffic and the requirements they place on the network (for example, by seeing the control messages or through path- trace techniques). Such knowledge can be used to launch targeted attacks (for example, taking down critical links) or can reveal commercially sensitive information (for example, whether a network is close to capacity). Network operators may, therefore, choose techniques that mask or hide information from within the network.

#### 10. IANA Considerations

This draft makes no requests for IANA action.

#### 11. Acknowledgments

Much of the text in this document is derived from RFC 3272. The authors of this document would like to express their gratitude to all involved in that work. Although the source text has been edited in the production of this document, the original authors should be considered as Contributors to this work. They were:

Daniel O. Awduche  
Movaz Networks

Angela Chiu  
Celion Networks

Anwar Elwalid  
Lucent Technologies

Indra Widjaja  
Bell Labs, Lucent Technologies

XiPeng Xiao  
Redback Networks

The acknowledgements in RFC3272 were as below. All people who helped in the production of that document also need to be thanked for the carry-over into this new document.

The authors would like to thank Jim Boyle for inputs on the recommendations section, Francois Le Faucheur for inputs on Diffserv aspects, Blaine Christian for inputs on measurement, Gerald Ash for inputs on routing in telephone networks and for text on event-dependent TE methods, Steven Wright for inputs on network controllability, and Jonathan Aufderheide for inputs on inter-domain TE with BGP. Special thanks to Randy Bush for proposing the TE taxonomy based on "tactical versus strategic" methods. The subsection describing an "Overview of ITU Activities Related to Traffic Engineering" was adapted from a contribution by Waisum Lai. Useful feedback and pointers to relevant materials were provided by J. Noel Chiappa. Additional comments were provided by Glenn Grotefeld during the working last call process. Finally, the authors would like to thank Ed Kern, the TEWG co-chair, for his comments and support.

The early versions of this document were produced by the TEAS Working Group's RFC3272bis Design Team. The full list of members of this team is:

Acee Lindem  
Adrian Farrel  
Aijun Wang  
Daniele Ceccarelli  
Dieter Beller  
Jeff Tantsura  
Julien Meuric  
Liu Hua  
Loa Andersson  
Luis Miguel Contreras  
Martin Horneffer  
Tarek Saad  
Xufeng Liu

The production of this document includes a fix to the original text resulting from an Errata Report by Jean-Michel Grimaldi.

The author of this document would also like to thank Dhurv Dhody for review comments.

## 12. Contributors

The following people contributed substantive text to this document:

Gert Grammel  
EMail: ggrammel@juniper.net

Loa Andersson  
EMail: loa@pi.nu

Xufeng Liu  
EMail: xufeng.liu.ietf@gmail.com

Lou Berger  
EMail: lberger@labn.net

Jeff Tantsura  
EMail: jefftant.ietf@gmail.com

Daniel King  
EMail: daniel@olddog.co.uk

Boris Hassanov  
EMail: bhassanov@yandex-team.ru

Kiran Makhijani  
Email: kiranm@futurewei.com

Dhruv Dhody  
Email: dhruv.ietf@gmail.com

### 13. Informative References

- [AJ19]     Adekitan, A., Abolade, J., and O. Shobayo, "Data mining approach for predicting the daily Internet data traffic of a smart university", Article Journal of Big Data, 2019, Volume 6, Number 1, Page 1, 1998.
- [ASH2]     Ash, J., "Dynamic Routing in Telecommunications Networks", Book McGraw Hill, 1998.
- [AWD2]     Awduche, D., "MPLS and Traffic Engineering in IP Networks", Article IEEE Communications Magazine, December 1999.
- [AWD5]     Awduche, D., "An Approach to Optimal Peering Between Autonomous Systems in the Internet", Paper International Conference on Computer Communications and Networks (ICCCN'98), October 1998.

- [FLJA93]    Floyd, S. and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", Article IEEE/ACM Transactions on Networking, Vol. 1, p. 387-413, November 1993.
- [FLOY94]    Floyd, S., "TCP and Explicit Congestion Notification", Article ACM Computer Communication Review, V. 24, No. 5, p. 10-23, October 1994.
- [FT00]    Fortz, B. and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights", Article IEEE INFOCOM 2000, March 2000.
- [FT01]    Fortz, B. and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World", n.d.,  
<<http://www.research.att.com/~mthorup/PAPERS/papers.html>>.
- [HUSS87]    Hurley, B., Seidl, C., and W. Sewel, "A Survey of Dynamic Routing Methods for Circuit-Switched Traffic", Article IEEE Communication Magazine, September 1987.
- [I-D.ietf-alto-performance-metrics]  
Wu, Q., Yang, Y. R., Lee, Y., Dhody, D., Randriamasy, S., and L. M. Contreras, "ALTO Performance Cost Metrics", draft-ietf-alto-performance-metrics-15 (work in progress), February 2021.
- [I-D.ietf-bess-evpn-unequal-lb]  
Malhotra, N., Sajassi, A., Rabadan, J., Drake, J., Lingala, A., and S. Thoria, "Weighted Multi-Path Procedures for EVPN All-Active Multi-Homing", draft-ietf-bess-evpn-unequal-lb-08 (work in progress), February 2021.
- [I-D.ietf-detnet-ip-over-tsn]  
Varga, B., Farkas, J., Malis, A. G., and S. Bryant, "DetNet Data Plane: IP over IEEE 802.1 Time Sensitive Networking (TSN)", draft-ietf-detnet-ip-over-tsn-07 (work in progress), February 2021.
- [I-D.ietf-idr-performance-routing]  
Xu, X., Hegde, S., Talaulikar, K., Boucadair, M., and C. Jacquenet, "Performance-based BGP Routing Mechanism", draft-ietf-idr-performance-routing-03 (work in progress), December 2020.

- [I-D.ietf-idr-segment-routing-te-policy]  
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P.,  
Rosen, E., Jain, D., and S. Lin, "Advertising Segment  
Routing Policies in BGP", draft-ietf-idr-segment-routing-  
te-policy-11 (work in progress), November 2020.
- [I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and  
A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-  
algo-15 (work in progress), April 2021.
- [I-D.ietf-lsr-ip-flexalgo]  
Britto, W., Hegde, S., Kaneriy, P., Shetty, R., Bonica,  
R., and P. Psenak, "IGP Flexible Algorithms (Flex-  
Algorithm) In IP Networks", draft-ietf-lsr-ip-flexalgo-02  
(work in progress), April 2021.
- [I-D.ietf-quic-transport]  
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed  
and Secure Transport", draft-ietf-quic-transport-34 (work  
in progress), January 2021.
- [I-D.ietf-spring-segment-routing-policy]  
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and  
P. Mattes, "Segment Routing Policy Architecture", draft-  
ietf-spring-segment-routing-policy-11 (work in progress),  
April 2021.
- [I-D.ietf-teas-enhanced-vpn]  
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A  
Framework for Enhanced Virtual Private Network (VPN+)  
Services", draft-ietf-teas-enhanced-vpn-07 (work in  
progress), February 2021.
- [I-D.ietf-teas-ietf-network-slices]  
Farrel, A., Gray, E., Drake, J., Rokui, R., Homma, S.,  
Makhijani, K., Contreras, L. M., and J. Tantsura,  
"Framework for IETF Network Slices", draft-ietf-teas-ietf-  
network-slices-00 (work in progress), April 2021.
- [I-D.ietf-tewg-qos-routing]  
Ash, G., "Traffic Engineering & QoS Methods for IP-, ATM-,  
& Based Multiservice Networks", draft-ietf-tewg-qos-  
routing-04 (work in progress), October 2001.

- [I-D.irtf-nmrg-ibn-concepts-definitions]  
Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", draft-irtf-nmrg-ibn-concepts-definitions-03 (work in progress), February 2021.
- [ITU-E600]  
"Terms and Definitions of Traffic Engineering",  
Recommendation ITU-T Recommendation E.600, March 1993.
- [ITU-E701]  
"Reference Connections for Traffic Engineering",  
Recommendation ITU-T Recommendation E.701, October 1993.
- [ITU-E801]  
"Framework for Service Quality Agreement",  
Recommendation ITU-T Recommendation E.801, October 1996.
- [MA]  
Ma, Q., "Quality of Service Routing in Integrated Services Networks", Ph.D. PhD Dissertation, CMU-CS-98-138, CMU, 1998.
- [MATE]  
Elwalid, A., Jin, C., Low, S., and I. Widjaja, "MATE - MPLS Adaptive Traffic Engineering",  
Proceedings INFOCOM'01, April 2001.
- [MCQ80]  
McQuillan, J., Richer, I., and E. Rosen, "The New Routing Algorithm for the ARPANET", Transaction IEEE Transactions on Communications, vol. 28, no. 5, p. 711-719, May 1980.
- [MR99]  
Mitra, D. and K. Ramakrishnan, "A Case Study of Multiservice, Multipriority Traffic Engineering Design for Data Networks", Proceedings Globecom'99, December 1999.
- [RFC0791]  
Postel, J., "Internet Protocol", STD 5, RFC 791,  
DOI 10.17487/RFC0791, September 1981,  
<<https://www.rfc-editor.org/info/rfc791>>.
- [RFC1102]  
Clark, D., "Policy routing in Internet protocols",  
RFC 1102, DOI 10.17487/RFC1102, May 1989,  
<<https://www.rfc-editor.org/info/rfc1102>>.
- [RFC1104]  
Braun, H., "Models of policy based routing", RFC 1104,  
DOI 10.17487/RFC1104, June 1989,  
<<https://www.rfc-editor.org/info/rfc1104>>.



- [RFC1992] Castineyra, I., Chiappa, N., and M. Steenstrup, "The Nimrod Routing Architecture", RFC 1992, DOI 10.17487/RFC1992, August 1996, <<https://www.rfc-editor.org/info/rfc1992>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<https://www.rfc-editor.org/info/rfc2330>>.
- [RFC2386] Crawley, E., Nair, R., Rajagopalan, B., and H. Sandick, "A Framework for QoS-based Routing in the Internet", RFC 2386, DOI 10.17487/RFC2386, August 1998, <<https://www.rfc-editor.org/info/rfc2386>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <<https://www.rfc-editor.org/info/rfc2475>>.
- [RFC2597] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, DOI 10.17487/RFC2597, June 1999, <<https://www.rfc-editor.org/info/rfc2597>>.
- [RFC2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, DOI 10.17487/RFC2678, September 1999, <<https://www.rfc-editor.org/info/rfc2678>>.
- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <<https://www.rfc-editor.org/info/rfc2702>>.

- [RFC2722] Brownlee, N., Mills, C., and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2722, DOI 10.17487/RFC2722, October 1999, <<https://www.rfc-editor.org/info/rfc2722>>.
- [RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, DOI 10.17487/RFC2753, January 2000, <<https://www.rfc-editor.org/info/rfc2753>>.
- [RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <<https://www.rfc-editor.org/info/rfc2961>>.
- [RFC2998] Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J., and E. Felstaine, "A Framework for Integrated Services Operation over Diffserv Networks", RFC 2998, DOI 10.17487/RFC2998, November 2000, <<https://www.rfc-editor.org/info/rfc2998>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3086] Nichols, K. and B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification", RFC 3086, DOI 10.17487/RFC3086, April 2001, <<https://www.rfc-editor.org/info/rfc3086>>.
- [RFC3124] Balakrishnan, H. and S. Seshan, "The Congestion Manager", RFC 3124, DOI 10.17487/RFC3124, June 2001, <<https://www.rfc-editor.org/info/rfc3124>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <<https://www.rfc-editor.org/info/rfc3270>>.

- [RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <<https://www.rfc-editor.org/info/rfc3272>>.
- [RFC3469] Sharma, V., Ed. and F. Hellstrand, Ed., "Framework for Multi-Protocol Label Switching (MPLS)-based Recovery", RFC 3469, DOI 10.17487/RFC3469, February 2003, <<https://www.rfc-editor.org/info/rfc3469>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <<https://www.rfc-editor.org/info/rfc3945>>.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<https://www.rfc-editor.org/info/rfc4090>>.
- [RFC4124] Le Faucheur, F., Ed., "Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering", RFC 4124, DOI 10.17487/RFC4124, June 2005, <<https://www.rfc-editor.org/info/rfc4124>>.
- [RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <<https://www.rfc-editor.org/info/rfc4203>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, DOI 10.17487/RFC4594, August 2006, <<https://www.rfc-editor.org/info/rfc4594>>.

- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <<https://www.rfc-editor.org/info/rfc4872>>.
- [RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, DOI 10.17487/RFC4873, May 2007, <<https://www.rfc-editor.org/info/rfc4873>>.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <<https://www.rfc-editor.org/info/rfc5250>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed., "Traffic Engineering Extensions to OSPF Version 3", RFC 5329, DOI 10.17487/RFC5329, September 2008, <<https://www.rfc-editor.org/info/rfc5329>>.
- [RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <<https://www.rfc-editor.org/info/rfc5331>>.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, DOI 10.17487/RFC5394, December 2008, <<https://www.rfc-editor.org/info/rfc5394>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <<https://www.rfc-editor.org/info/rfc5541>>.

- [RFC5557] Lee, Y., Le Roux, J.L., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, DOI 10.17487/RFC5557, July 2009, <<https://www.rfc-editor.org/info/rfc5557>>.
- [RFC5664] Halevy, B., Welch, B., and J. Zelenka, "Object-Based Parallel NFS (pNFS) Operations", RFC 5664, DOI 10.17487/RFC5664, January 2010, <<https://www.rfc-editor.org/info/rfc5664>>.
- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <<https://www.rfc-editor.org/info/rfc5693>>.
- [RFC6107] Shiomoto, K., Ed. and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <<https://www.rfc-editor.org/info/rfc6107>>.
- [RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, DOI 10.17487/RFC6119, February 2011, <<https://www.rfc-editor.org/info/rfc6119>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6372] Sprecher, N., Ed. and A. Farrel, Ed., "MPLS Transport Profile (MPLS-TP) Survivability Framework", RFC 6372, DOI 10.17487/RFC6372, September 2011, <<https://www.rfc-editor.org/info/rfc6372>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <<https://www.rfc-editor.org/info/rfc6805>>.

- [RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <<https://www.rfc-editor.org/info/rfc7149>>.
- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <<https://www.rfc-editor.org/info/rfc7285>>.
- [RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <<https://www.rfc-editor.org/info/rfc7390>>.
- [RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <<https://www.rfc-editor.org/info/rfc7426>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, DOI 10.17487/RFC7491, March 2015, <<https://www.rfc-editor.org/info/rfc7491>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.

- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <<https://www.rfc-editor.org/info/rfc7923>>.
- [RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <<https://www.rfc-editor.org/info/rfc7926>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<https://www.rfc-editor.org/info/rfc8051>>.
- [RFC8189] Randriamasy, S., Roome, W., and N. Schwan, "Multi-Cost Application-Layer Traffic Optimization (ALTO)", RFC 8189, DOI 10.17487/RFC8189, October 2017, <<https://www.rfc-editor.org/info/rfc8189>>.
- [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <<https://www.rfc-editor.org/info/rfc8231>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.

- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.
- [RFC8283] Farrel, A., Ed., Zhao, Q., Ed., Li, Z., and C. Zhou, "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", RFC 8283, DOI 10.17487/RFC8283, December 2017, <<https://www.rfc-editor.org/info/rfc8283>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <<https://www.rfc-editor.org/info/rfc8570>>.
- [RFC8571] Ginsberg, L., Ed., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, DOI 10.17487/RFC8571, March 2019, <<https://www.rfc-editor.org/info/rfc8571>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <<https://www.rfc-editor.org/info/rfc8664>>.



- [RFC8685] Zhang, F., Zhao, Q., Gonzalez de Dios, O., Casellas, R., and D. King, "Path Computation Element Communication Protocol (PCEP) Extensions for the Hierarchical Path Computation Element (H-PCE) Architecture", RFC 8685, DOI 10.17487/RFC8685, December 2019, <<https://www.rfc-editor.org/info/rfc8685>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.
- [RFC8896] Randriamasy, S., Yang, R., Wu, Q., Deng, L., and N. Schwan, "Application-Layer Traffic Optimization (ALTO) Cost Calendar", RFC 8896, DOI 10.17487/RFC8896, November 2020, <<https://www.rfc-editor.org/info/rfc8896>>.
- [RFC8938] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC 8938, DOI 10.17487/RFC8938, November 2020, <<https://www.rfc-editor.org/info/rfc8938>>.
- [RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <<https://www.rfc-editor.org/info/rfc8955>>.
- [RR94] Rodrigues, M. and K. Ramakrishnan, "Optimal Routing in Shortest Path Networks", Proceedings ITS'94, Rio de Janeiro, Brazil, 1994.
- [SLDC98] Suter, B., Lakshman, T., Stiliadis, D., and A. Choudhury, "Design Considerations for Supporting TCP with Per-flow Queueing", Proceedings INFOCOM'98, p. 299-306, 1998.
- [WANG] Wang, Y., Wang, Z., and L. Zhang, "Internet traffic engineering without full mesh overlaying", Proceedings INFOCOM'2001, April 2001.
- [XIAO] Xiao, X., Hannan, A., Bailey, B., and L. Ni, "Traffic Engineering with MPLS in the Internet", Article IEEE Network Magazine, March 2000.
- [YARE95] Yang, C. and A. Reddy, "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks", Article IEEE Network Magazine, p. 34-45, 1995.

## Appendix A.    Historic Overview

### A.1.    Traffic Engineering in Classical Telephone Networks

This subsection presents a brief overview of traffic engineering in telephone networks which often relates to the way user traffic is steered from an originating node to the terminating node. This subsection presents a brief overview of this topic. A detailed description of the various routing strategies applied in telephone networks is included in the book by G. Ash [ASH2].

The early telephone network relied on static hierarchical routing, whereby routing patterns remained fixed independent of the state of the network or time of day. The hierarchy was intended to accommodate overflow traffic, improve network reliability via alternate routes, and prevent call looping by employing strict hierarchical rules. The network was typically over-provisioned since a given fixed route had to be dimensioned so that it could carry user traffic during a busy hour of any busy day. Hierarchical routing in the telephony network was found to be too rigid upon the advent of digital switches and stored program control which were able to manage more complicated traffic engineering rules.

Dynamic routing was introduced to alleviate the routing inflexibility in the static hierarchical routing so that the network would operate more efficiently. This resulted in significant economic gains [HUSS87]. Dynamic routing typically reduces the overall loss probability by 10 to 20 percent (compared to static hierarchical routing). Dynamic routing can also improve network resilience by recalculating routes on a per-call basis and periodically updating routes.

There are three main types of dynamic routing in the telephone network. They are time-dependent routing, state-dependent routing (SDR), and event dependent routing (EDR).

In time-dependent routing, regular variations in traffic loads (such as time of day or day of week) are exploited in pre-planned routing tables. In state-dependent routing, routing tables are updated online according to the current state of the network (e.g., traffic demand, utilization, etc.). In event dependent routing, routing changes are triggered by events (such as call setups encountering congested or blocked links) whereupon new paths are searched out using learning models. EDR methods are real-time adaptive, but they do not require global state information as does SDR. Examples of EDR schemes include the dynamic alternate routing (DAR) from BT, the state-and-time dependent routing (STR) from NTT, and the success-to-the-top (STT) routing from AT&T.

Dynamic non-hierarchical routing (DNHR) is an example of dynamic routing that was introduced in the AT&T toll network in the 1980's to respond to time-dependent information such as regular load variations as a function of time. Time-dependent information in terms of load may be divided into three timescales: hourly, weekly, and yearly. Correspondingly, three algorithms are defined to pre-plan the routing tables. The network design algorithm operates over a year-long interval while the demand servicing algorithm operates on a weekly basis to fine tune link sizes and routing tables to correct forecast errors on the yearly basis. At the smallest timescale, the routing algorithm is used to make limited adjustments based on daily traffic variations. Network design and demand servicing are computed using offline calculations. Typically, the calculations require extensive searches on possible routes. On the other hand, routing may need online calculations to handle crankback. DNHR adopts a "two-link" approach whereby a path can consist of two links at most. The routing algorithm presents an ordered list of route choices between an originating switch and a terminating switch. If a call overflows, a via switch (a tandem exchange between the originating switch and the terminating switch) would send a crankback signal to the originating switch. This switch would then select the next route, and so on, until there are no alternative routes available in which the call is blocked.

#### A.2. Evolution of Traffic Engineering in Packet Networks

This subsection reviews related prior work that was intended to improve the performance of data networks. Indeed, optimization of the performance of data networks started in the early days of the ARPANET. Other early commercial networks such as SNA also recognized the importance of performance optimization and service differentiation.

In terms of traffic management, the Internet has been a best effort service environment until recently. In particular, very limited traffic management capabilities existed in IP networks to provide differentiated queue management and scheduling services to packets belonging to different classes.

In terms of routing control, the Internet has employed distributed protocols for intra-domain routing. These protocols are highly scalable and resilient. However, they are based on simple algorithms for path selection which have very limited functionality to allow flexible control of the path selection process.

In the following subsections, the evolution of practical traffic engineering mechanisms in IP networks and its predecessors are reviewed.

#### A.2.1. Adaptive Routing in the ARPANET

The early ARPANET recognized the importance of adaptive routing where routing decisions were based on the current state of the network [MCQ80]. Early minimum delay routing approaches forwarded each packet to its destination along a path for which the total estimated transit time was the smallest. Each node maintained a table of network delays, representing the estimated delay that a packet would experience along a given path toward its destination. The minimum delay table was periodically transmitted by a node to its neighbors. The shortest path, in terms of hop count, was also propagated to give the connectivity information.

One drawback to this approach is that dynamic link metrics tend to create "traffic magnets" causing congestion to be shifted from one location of a network to another location, resulting in oscillation and network instability.

#### A.2.2. Dynamic Routing in the Internet

The Internet evolved from the ARPANET and adopted dynamic routing algorithms with distributed control to determine the paths that packets should take en-route to their destinations. The routing algorithms are adaptations of shortest path algorithms where costs are based on link metrics. The link metric can be based on static or dynamic quantities. The link metric based on static quantities may be assigned administratively according to local criteria. The link metric based on dynamic quantities may be a function of a network congestion measure such as delay or packet loss.

It was apparent early that static link metric assignment was inadequate because it can easily lead to unfavorable scenarios in which some links become congested while others remain lightly loaded. One of the many reasons for the inadequacy of static link metrics is that link metric assignment was often done without considering the traffic matrix in the network. Also, the routing protocols did not take traffic attributes and capacity constraints into account when making routing decisions. This results in traffic concentration being localized in subsets of the network infrastructure and potentially causing congestion. Even if link metrics are assigned in accordance with the traffic matrix, unbalanced loads in the network can still occur due to a number factors including:

- o Resources may not be deployed in the most optimal locations from a routing perspective.
- o Forecasting errors in traffic volume and/or traffic distribution.

- o Dynamics in traffic matrix due to the temporal nature of traffic patterns, BGP policy change from peers, etc.

The inadequacy of the legacy Internet interior gateway routing system is one of the factors motivating the interest in path oriented technology with explicit routing and constraint-based routing capability such as MPLS.

#### A.2.3. ToS Routing

Type-of-Service (ToS) routing involves different routes going to the same destination with selection dependent upon the ToS field of an IP packet [RFC2474]. The ToS classes may be classified as low delay and high throughput. Each link is associated with multiple link costs and each link cost is used to compute routes for a particular ToS. A separate shortest path tree is computed for each ToS. The shortest path algorithm must be run for each ToS resulting in very expensive computation. Classical ToS-based routing is now outdated as the IP header field has been replaced by a Diffserv field. Effective traffic engineering is difficult to perform in classical ToS-based routing because each class still relies exclusively on shortest path routing which results in localization of traffic concentration within the network.

#### A.2.4. Equal Cost Multi-Path

Equal Cost Multi-Path (ECMP) is another technique that attempts to address the deficiency in the Shortest Path First (SPF) interior gateway routing systems [RFC2328]. In the classical SPF algorithm, if two or more shortest paths exist to a given destination, the algorithm will choose one of them. The algorithm is modified slightly in ECMP so that if two or more equal cost shortest paths exist between two nodes, the traffic between the nodes is distributed among the multiple equal-cost paths. Traffic distribution across the equal-cost paths is usually performed in one of two ways: (1) packet-based in a round-robin fashion, or (2) flow-based using hashing on source and destination IP addresses and possibly other fields of the IP header. The first approach can easily cause out-of-order packets while the second approach is dependent upon the number and distribution of flows. Flow-based load sharing may be unpredictable in an enterprise network where the number of flows is relatively small and less heterogeneous (for example, hashing may not be uniform), but it is generally effective in core public networks where the number of flows is large and heterogeneous.

In ECMP, link costs are static and bandwidth constraints are not considered, so ECMP attempts to distribute the traffic as equally as possible among the equal-cost paths independent of the congestion

status of each path. As a result, given two equal-cost paths, it is possible that one of the paths will be more congested than the other. Another drawback of ECMP is that load sharing cannot be achieved on multiple paths which have non-identical costs.

#### A.2.5. Nimrod

Nimrod was a routing system developed to provide heterogeneous service specific routing in the Internet, while taking multiple constraints into account [RFC1992]. Essentially, Nimrod was a link state routing protocol to support path oriented packet forwarding. It used the concept of maps to represent network connectivity and services at multiple levels of abstraction. Mechanisms allowed restriction of the distribution of routing information.

Even though Nimrod did not enjoy deployment in the public Internet, a number of key concepts incorporated into the Nimrod architecture, such as explicit routing which allows selection of paths at originating nodes, are beginning to find applications in some recent constraint-based routing initiatives.

### A.3. Development of Internet Traffic Engineering

#### A.3.1. Overlay Model

In the overlay model, a virtual-circuit network, such as Synchronous Optical Network / Synchronous Digital Hierarchy (SONET/SDH), Optical Transport Network (OTN), or Wavelength Division Multiplexing (WDM), provides virtual-circuit connectivity between routers that are located at the edges of a virtual-circuit cloud. In this mode, two routers that are connected through a virtual circuit see a direct adjacency between themselves independent of the physical route taken by the virtual circuit through the ATM, frame relay, or WDM network. Thus, the overlay model essentially decouples the logical topology that routers see from the physical topology that the ATM, frame relay, or WDM network manages. The overlay model based on ATM or frame relay enables a network administrator or an automaton to employ traffic engineering concepts to perform path optimization by re-configuring or rearranging the virtual circuits so that a virtual circuit on a congested or sub-optimal physical link can be re-routed to a less congested or more optimal one. In the overlay model, traffic engineering is also employed to establish relationships between the traffic management parameters (e.g., Peak Cell Rate, Sustained Cell Rate, and Maximum Burst Size for ATM) of the virtual-circuit technology and the actual traffic that traverses each circuit. These relationships can be established based upon known or projected traffic profiles, and some other factors.

## Appendix B.   Overview of Traffic Engineering Related Work in Other SDOs

## B.1.   Overview of ITU Activities Related to Traffic Engineering

This section provides an overview of prior work within the ITU-T pertaining to traffic engineering in traditional telecommunications networks.

ITU-T Recommendations E.600 [ITU-E600], E.701 [ITU-E701], and E.801 [ITU-E801] address traffic engineering issues in traditional telecommunications networks. Recommendation E.600 provides a vocabulary for describing traffic engineering concepts, while E.701 defines reference connections, Grade of Service (GoS), and traffic parameters for ISDN. Recommendation E.701 uses the concept of a reference connection to identify representative cases of different types of connections without describing the specifics of their actual realizations by different physical means. As defined in Recommendation E.600, "a connection is an association of resources providing means for communication between two or more devices in, or attached to, a telecommunication network." Also, E.600 defines "a resource as any set of physically or conceptually identifiable entities within a telecommunication network, the use of which can be unambiguously determined" [ITU-E600]. There can be different types of connections as the number and types of resources in a connection may vary.

Typically, different network segments are involved in the path of a connection. For example, a connection may be local, national, or international. The purposes of reference connections are to clarify and specify traffic performance issues at various interfaces between different network domains. Each domain may consist of one or more service provider networks.

Reference connections provide a basis to define grade of service (GoS) parameters related to traffic engineering within the ITU-T framework. As defined in E.600, "GoS refers to a number of traffic engineering variables which are used to provide a measure of the adequacy of a group of resources under specified conditions." These GoS variables may be probability of loss, dial tone, delay, etc. They are essential for network internal design and operation as well as for component performance specification.

GoS is different from quality of service (QoS) in the ITU framework. QoS is the performance perceivable by a telecommunication service user and expresses the user's degree of satisfaction of the service. QoS parameters focus on performance aspects observable at the service access points and network interfaces, rather than their causes within the network. GoS, on the other hand, is a set of network oriented

measures which characterize the adequacy of a group of resources under specified conditions. For a network to be effective in serving its users, the values of both GoS and QoS parameters must be related, with GoS parameters typically making a major contribution to the QoS.

Recommendation E.600 stipulates that a set of GoS parameters must be selected and defined on an end-to-end basis for each major service category provided by a network to assist the network provider with improving efficiency and effectiveness of the network. Based on a selected set of reference connections, suitable target values are assigned to the selected GoS parameters under normal and high load conditions. These end-to-end GoS target values are then apportioned to individual resource components of the reference connections for dimensioning purposes.

#### Appendix C. Summary of Changes Since RFC 3272

The changes to this document since RFC 3272 are substantial and not easily summarized as section-by-section changes. The material in the document has been moved around considerably, some of it removed, and new text added.

The approach taken here is to list the table of content of both the previous RFC and this document saying, respectively, where the text has been place and where the text came from.

##### C.1. RFC 3272

1.0 Introduction: Edited in place in Section 1.

1.1 What is Internet Traffic Engineering?: Edited in place in Section 1.1.

1.2 Scope: Moved to Section 1.3.

1.3 Terminology: Moved to Section 1.4 with some obsolete terms removed and a little editing.

2.0 Background: Retained as Section 2 with some text removed.

2.1 Context of Internet Traffic Engineering: Retained as Section 2.1.

2.2 Network Context: Rewritten as Section 2.2.

2.3 Problem Context: Rewritten as Section 2.3.



- 2.3.1 Congestion and its Ramifications: Retained as Section 2.3.1.
- 2.4 Solution Context: Edited as Section 2.4.
  - 2.4.1 Combating the Congestion Problem: Reformatted as Section 2.4.1.
- 2.5 Implementation and Operational Context: Retained as Section 2.5.
- 3.0 Traffic Engineering Process Model: Retained as Section 3.
  - 3.1 Components of the Traffic Engineering Process Model: Retained as Section 3.1.
  - 3.2 Measurement: Merged into Section 3.1.
  - 3.3 Modeling, Analysis, and Simulation: Merged into Section 3.1.
  - 3.4 Optimization: Merged into Section 3.1.
- 4.0 Historical Review and Recent Developments: Retained as Section 5, but the very historic aspects moved to Appendix A.
  - 4.1 Traffic Engineering in Classical Telephone Networks: Moved to Appendix A.1.
  - 4.2 Evolution of Traffic Engineering in the Internet: Moved to Appendix A.2.
    - 4.2.1 Adaptive Routing in ARPANET: Moved to Appendix A.2.1.
    - 4.2.2 Dynamic Routing in the Internet: Moved to Appendix A.2.2.
    - 4.2.3 ToS Routing: Moved to Appendix A.2.3.
    - 4.2.4 Equal Cost Multi-Path: Moved to Appendix A.2.4.
    - 4.2.5 Nimrod: Moved to Appendix A.2.5.
  - 4.3 Overlay Model: Moved to Appendix A.3.1.
  - 4.4 Constraint-Based Routing: Retained as Section 5.1.1, but moved into Section 5.1.

- 4.5 Overview of Other IETF Projects Related to Traffic Engineering:  
Retained as Section 5.1 with many new subsections.
  - 4.5.1 Integrated Services: Retained as Section 5.1.2.
  - 4.5.2 RSVP: Retained as Section 5.1.3 with some edits.
  - 4.5.3 Differentiated Services: Retained as Section 5.1.4.
  - 4.5.4 MPLS: Retained as Section 5.1.6.
  - 4.5.5 IP Performance Metrics: Retained as Section 5.1.8.
  - 4.5.6 Flow Measurement: Retained as Section 5.1.9 with some reformatting.
  - 4.5.7 Endpoint Congestion Management: Retained as Section 5.1.10.
- 4.6 Overview of ITU Activities Related to Traffic Engineering: Moved to Appendix B.1.
- 4.7 Content Distribution: Retained as Section 5.2.
- 5.0 Taxonomy of Traffic Engineering Systems: Retained as Section 4.
  - 5.1 Time-Dependent Versus State-Dependent: Retained as Section 4.1.
  - 5.2 Offline Versus Online: Retained as Section 4.2.
  - 5.3 Centralized Versus Distributed: Retained as Section 4.3 with additions.
  - 5.4 Local Versus Global: Retained as Section 4.4.
  - 5.5 Prescriptive Versus Descriptive: Retained as Section 4.5 with additions.
  - 5.6 Open-Loop Versus Closed-Loop: Retained as Section 4.6.
  - 5.7 Tactical vs Strategic: Retained as Section 4.7.
- 6.0 Recommendations for Internet Traffic Engineering: Retained as Section 6.
  - 6.1 Generic Non-functional Recommendations: Retained as Section 6.1.

6.2 Routing Recommendations:    Retained as Section 6.2 with edits.

6.3 Traffic Mapping Recommendations:    Retained as Section 6.3.

6.4 Measurement Recommendations:    Retained as Section 6.4.

6.5 Network Survivability:    Retained as Section 6.6.

6.5.1 Survivability in MPLS Based Networks:    Retained as  
Section 6.6.1.

6.5.2 Protection Option:    Retained as Section 6.6.2.

6.6 Traffic Engineering in Diffserv Environments:    Retained as  
Section 6.8 with edits.

6.7 Network Controllability:    Retained as Section 6.9.

7.0 Inter-Domain Considerations:    Retained as Section 7.

8.0 Overview of Contemporary TE Practices in Operational IP Networks:  
Retained as Section 8.

9.0 Conclusion:    Removed.

10.0 Security Considerations:    Retained as Section 9 with  
considerable new text.

## C.2. This Document

- o Section 1: Based on Section 1 of RFC 3272.
  - \* Section 1.1: Based on Section 1.1 of RFC 3272.
  - \* Section 1.2: New for this document.
  - \* Section 1.3: Based on Section 1.2 of RFC 3272.
  - \* Section 1.4: Based on Section 1.3 of RFC 3272.
- o Section 2: Based on Section 2. of RFC 3272.
  - \* Section 2.1: Based on Section 2.1 of RFC 3272.
  - \* Section 2.2: Based on Section 2.2 of RFC 3272.
  - \* Section 2.3: Based on Section 2.3 of RFC 3272.

- + Section 2.3.1: Based on Section 2.3.1 of RFC 3272.
- \* Section 2.4: Based on Section 2.4 of RFC 3272.
  - + Section 2.4.1: Based on Section 2.4.1 of RFC 327
- \* Section 2.5: Based on Section 2.5 of RFC 3272.
- o Section 3: Based on Section 3 of RFC 3272.
  - \* Section 3.1: Based on Sections 3.1, 3.2, 3.3, and 3.4 of RFC 3272.
- o Section 4: Based on Section 5 of RFC 3272.
  - \* Section 4.1: Based on Section 5.1 of RFC 3272.
  - \* Section 4.2: Based on Section 5.2 of RFC 3272.
  - \* Section 4.3: Based on Section 5.3 of RFC 3272.
    - + Section 4.3.1: New for this document.
    - + Section 4.3.2: New for this document.
  - \* Section 4.4: Based on Section 5.4 of RFC 3272.
  - \* Section 4.5: Based on Section 5.5 of RFC 3272.
    - + Section 4.5.1: New for this document.
  - \* Section 4.6: Based on Section 5.6 of RFC 3272.
  - \* Section 4.7: Based on Section 5.7 of RFC 3272.
- o Section 5: Based on Section 4 of RFC 3272.
  - \* Section 5.1: Based on Section 4.5 of RFC 3272.
    - + Section 5.1.1: Based on Section 4.4 of RFC 3272.
      - Section 5.1.1.1: New for this document.
    - + Section 5.1.2: Based on Section 4.5.1 of RFC 3272.
    - + Section 5.1.3: Based on Section 4.5.2 of RFC 3272.
    - + Section 5.1.4: Based on Section 4.5.3 of RFC 3272.

- + Section 5.1.5: New for this document.
- + Section 5.1.6: Based on Section 4.5.4 of RFC 3272.
- + Section 5.1.7: New for this document.
- + Section 5.1.8: Based on Section 4.5.5 of RFC 3272.
- + Section 5.1.9: Based on Section 4.5.6 of RFC 3272.
- + Section 5.1.10: Based on Section 4.5.7 of RFC 3272.
- + Section 5.1.11: New for this document.
- + Section 5.1.12: New for this document.
- + Section 5.1.13: New for this document.
- + Section 5.1.14: New for this document.
- + Section 5.1.15: New for this document.
- + Section 5.1.16: New for this document.
- + Section 5.1.17: New for this document.
- + Section 5.1.18: New for this document.
- + Section 5.1.19: New for this document.
- + Section 5.1.20: New for this document.
- + Section 5.1.21: New for this document.
- \* Section 5.2: Based on Section 4.7 of RFC 3272.
- o Section 6: Based on Section 6 of RFC 3272.
  - \* Section 6.1: Based on Section 6.1 of RFC 3272.
  - \* Section 6.2: Based on Section 6.2 of RFC 3272.
  - \* Section 6.3: Based on Section 6.3 of RFC 3272.
  - \* Section 6.4: Based on Section 6.4 of RFC 3272.
  - \* Section 6.5: New for this document.

- \* Section 6.6: Based on Section 6.5 of RFC 3272.
  - + Section 6.6.1: Based on Section 6.5.1 of RFC 3272.
  - + Section 6.6.2: Based on Section 6.5.2 of RFC 3272.
- \* Section 6.7: New for this document.
- \* Section 6.8: Based on Section 6.6. of RFC 3272.
- \* Section 6.9: Based on Section 6.7 of RFC 3272.
- o Section 7: Based on Section 7 of RFC 3272.
- o Section 8: Based on Section 8 of RFC 3272.
- o Section 9: Based on Section 10 of RFC 3272.
- o Appendix A: New for this document.
  - \* Appendix A.1: Based on Section 4.1 of RFC 3272.
  - \* Appendix A.2: Based on Section 4.2 of RFC 3272.
    - + Appendix A.2.1: Based on Section 4.2.1 of RFC 3272.
    - + Appendix A.2.2: Based on Section 4.2.2 of RFC 3272.
    - + Appendix A.2.3: Based on Section 4.2.3 of RFC 3272.
    - + Appendix A.2.4: Based on Section 4.2.4 of RFC 3272.
    - + Appendix A.2.5: Based on Section 4.2.5 of RFC 3272.
  - \* Appendix A.3: New for this document.
    - + Appendix A.3.1: Based on Section 4.3 of RFC 3272.
- o Appendix B: New for this document.
  - \* Appendix B.1: Based on Section 4.7 of RFC 3272.

#### Author's Address

Adrian Farrel (editor)  
Old Dog Consulting

Email: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)



TEAS Working Group  
Internet-Draft  
Obsoletes: 3272 (if approved)  
Intended status: Informational  
Expires: 25 September 2022

A. Farrel, Ed.  
Old Dog Consulting  
24 March 2022

Overview and Principles of Internet Traffic Engineering  
draft-ietf-teas-rfc3272bis-16

Abstract

This document describes the principles of traffic engineering (TE) in the Internet. The document is intended to promote better understanding of the issues surrounding traffic engineering in IP networks and the networks that support IP networking, and to provide a common basis for the development of traffic engineering capabilities for the Internet. The principles, architectures, and methodologies for performance evaluation and performance optimization of operational networks are also discussed.

This work was first published as RFC 3272 in May 2002. This document obsoletes RFC 3272 by making a complete update to bring the text in line with best current practices for Internet traffic engineering and to include references to the latest relevant work in the IETF.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 25 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.



This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 3  |
| 1.1. What is Internet Traffic Engineering? . . . . .                           | 4  |
| 1.2. Components of Traffic Engineering . . . . .                               | 6  |
| 1.3. Scope . . . . .   | 8  |
| 1.4. Terminology . . . . .   | 8  |
| 2. Background . . . . .  | 11 |
| 2.1. Context of Internet Traffic Engineering . . . . .                         | 11 |
| 2.2. Network Domain Context . . . . .  | 12 |
| 2.3. Problem Context . . . . .   | 14 |
| 2.3.1. Congestion and its Ramifications . . . . .                              | 15 |
| 2.4. Solution Context . . . . .  | 15 |
| 2.4.1. Combating the Congestion Problem . . . . .                              | 17 |
| 2.5. Implementation and Operational Context . . . . .                          | 20 |
| 3. Traffic Engineering Process Models . . . . .                                | 21 |
| 3.1. Components of the Traffic Engineering Process Model . . . . .             | 21 |
| 4. Taxonomy of Traffic Engineering Systems . . . . .                           | 22 |
| 4.1. Time-Dependent Versus State-Dependent Versus<br>Event-Dependent . . . . . | 22 |
| 4.2. Offline Versus Online . . . . .   | 24 |
| 4.3. Centralized Versus Distributed . . . . .                                  | 24 |
| 4.3.1. Hybrid Systems . . . . .  | 25 |
| 4.3.2. Considerations for Software Defined Networking . . . . .                | 25 |
| 4.4. Local Versus Global . . . . .   | 26 |
| 4.5. Prescriptive Versus Descriptive . . . . .                                 | 26 |
| 4.5.1. Intent-Based Networking . . . . .                                       | 27 |
| 4.6. Open-Loop Versus Closed-Loop . . . . .                                    | 27 |
| 4.7. Tactical versus Strategic . . . . .                                       | 27 |
| 5. Review of TE Techniques . . . . .   | 28 |
| 5.1. Overview of IETF Projects Related to Traffic<br>Engineering . . . . .     | 28 |
| 5.1.1. IETF TE Mechanisms . . . . .  | 28 |
| 5.1.2. IETF Approaches Relying on TE Mechanisms . . . . .                      | 33 |
| 5.1.3. IETF Techniques Used by TE Mechanisms . . . . .                         | 35 |
| 5.2. Content Distribution . . . . .  | 44 |
| 6. Recommendations for Internet Traffic Engineering . . . . .                  | 45 |
| 6.1. Generic Non-functional Recommendations . . . . .                          | 45 |
| 6.2. Routing Recommendations . . . . .   | 47 |

|   |    |
|---|----|
| 6.3. Traffic Mapping Recommendations . . . . .                                      | 50 |
| 6.4. Measurement Recommendations . . . . .  | 51 |
| 6.5. Policing, Planning, and Access Control . . . . .                               | 51 |
| 6.6. Network Survivability . . . . .  | 52 |
| 6.6.1. Survivability in MPLS Based Networks . . . . .                               | 54 |
| 6.6.2. Protection Options . . . . .   | 55 |
| 6.7. Multi-Layer Traffic Engineering . . . . .                                      | 56 |
| 6.8. Traffic Engineering in Diffserv Environments . . . . .                         | 57 |
| 6.9. Network Controllability . . . . .  | 58 |
| 7. Inter-Domain Considerations . . . . .  | 59 |
| 8. Overview of Contemporary TE Practices in Operational IP<br>Networks . . . . .    | 60 |
| 9. Security Considerations . . . . .  | 63 |
| 10. IANA Considerations . . . . .   | 64 |
| 11. Acknowledgments . . . . .   | 64 |
| 12. Contributors . . . . .  | 66 |
| 13. Informative References . . . . .  | 66 |
| Appendix A. Historic Overview . . . . .   | 81 |
| A.1. Traffic Engineering in Classical Telephone Networks . . . . .                  | 81 |
| A.2. Evolution of Traffic Engineering in Packet Networks . . . . .                  | 82 |
| A.2.1. Adaptive Routing in the ARPANET . . . . .                                    | 83 |
| A.2.2. Dynamic Routing in the Internet . . . . .                                    | 83 |
| A.2.3. ToS Routing . . . . .  | 84 |
| A.2.4. Equal Cost Multi-Path . . . . .  | 84 |
| A.2.5. Nimrod . . . . .   | 85 |
| A.3. Development of Internet Traffic Engineering . . . . .                          | 85 |
| A.3.1. Overlay Model . . . . .  | 85 |
| Appendix B. Overview of Traffic Engineering Related Work in Other<br>SDOs . . . . . | 86 |
| B.1. Overview of ITU Activities Related to Traffic<br>Engineering . . . . .         | 86 |
| Appendix C. Summary of Changes Since RFC 3272 . . . . .                             | 87 |
| C.1. RFC 3272 . . . . .   | 87 |
| C.2. This Document . . . . .  | 90 |
| Author's Address . . . . .  | 94 |

## 1. Introduction

This document describes the principles of Internet traffic engineering (TE). The objective of the document is to articulate the general issues and principles for Internet TE, and where appropriate to provide recommendations, guidelines, and options for the development of preplanned (offline) and dynamic (online) Internet TE capabilities and support systems.

Even though Internet TE is most effective when applied end-to-end, the focus of this document is TE within a given domain (such as an autonomous system). However, because a preponderance of Internet

traffic tends to originate in one autonomous system and terminate in another, this document also provides an overview of aspects pertaining to inter-domain TE.

This document provides a terminology and taxonomy for describing and understanding common Internet TE concepts.

This work was first published as [RFC3272] in May 2002. This document obsoletes [RFC3272] by making a complete update to bring the text in line with best current practices for Internet TE and to include references to the latest relevant work in the IETF. It is worth noting around three fifths of the RFCs referenced in this document post-date the publication of RFC 3272. Appendix C provides a summary of changes between RFC 3272 and this document.

### 1.1. What is Internet Traffic Engineering?

One of the most significant functions performed in the Internet is the routing and forwarding of traffic from ingress nodes to egress nodes. Therefore, one of the most distinctive functions performed by Internet traffic engineering is the control and optimization of these routing and forwarding functions, to steer traffic through the network.

Internet traffic engineering is defined as that aspect of Internet network engineering dealing with the issues of performance evaluation and performance optimization of operational IP networks. Traffic engineering encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic [RFC2702], [AWD2].

It is the performance of the network as seen by end users of network services that is paramount. The characteristics visible to end users are the emergent properties of the network, which are the characteristics of the network when viewed as a whole. A central goal of the service provider, therefore, is to enhance the emergent properties of the network while taking economic considerations into account. This is accomplished by addressing traffic oriented performance requirements while utilizing network resources without waste and in a reliable way. Traffic oriented performance measures include delay, delay variation, packet loss, and throughput.

Internet TE responds to network events (such as link or node failures, reported or predicted congestion, planned maintenance, service degradation, planned changes in the traffic matrix, etc.). Aspects of capacity management respond at intervals ranging from days to years. Routing control functions operate at intervals ranging from milliseconds to days. Packet level processing functions operate

at very fine levels of temporal resolution, ranging from picoseconds to milliseconds while reacting to the real-time statistical behavior of traffic.

Thus, the optimization aspects of TE can be viewed from a control perspective, and can be both pro-active and reactive. In the pro-active case, the TE control system takes preventive action to protect against predicted unfavorable future network states, for example, by engineering backup paths. It may also take action that will lead to a more desirable future network state. In the reactive case, the control system responds to correct issues and adapt to network events, such as routing after failure.

Another important objective of Internet TE is to facilitate reliable network operations [RFC2702]. Reliable network operations can be facilitated by providing mechanisms that enhance network integrity and by embracing policies emphasizing network survivability. This reduces the vulnerability of services to outages arising from errors, faults, and failures occurring within the network infrastructure.

The optimization aspects of TE can be achieved through capacity management and traffic management. In this document, capacity management includes capacity planning, routing control, and resource management. Network resources of particular interest include link bandwidth, buffer space, and computational resources. In this document, traffic management includes:

1. Nodal traffic control functions such as traffic conditioning, queue management, and scheduling.
2. Other functions that regulate the flow of traffic through the network or that arbitrate access to network resources between different packets or between different traffic flows.

One major challenge of Internet TE is the realization of automated control capabilities that adapt quickly and cost effectively to significant changes in network state, while still maintaining stability of the network. Performance evaluation can assess the effectiveness of TE methods, and the results of this evaluation can be used to identify existing problems, guide network re-optimization, and aid in the prediction of potential future problems. However, this process can also be time consuming and may not be suitable to act on short-lived changes in the network.

Performance evaluation can be achieved in many different ways. The most notable techniques include analytic methods, simulation, and empirical methods based on measurements.

Traffic engineering comes in two flavors:

- \* A background process that constantly monitors traffic and network conditions, and optimizes the use of resources to improve performance.
- \* A form of a pre-planned optimized traffic distribution that is considered optimal.

In the later case, any deviation from the optimum distribution (e.g., caused by a fiber cut) is reverted upon repair without further optimization. However, this form of TE relies upon the notion that the planned state of the network is optimal. Hence, in such a mode there are two levels of TE: the TE-planning task to enable optimum traffic distribution, and the routing and forwarding tasks that keep traffic flows attached to the pre-planned distribution.

As a general rule, TE concepts and mechanisms must be sufficiently specific and well-defined to address known requirements, but simultaneously flexible and extensible to accommodate unforeseen future demands (see Section 6.1).

## 1.2. Components of Traffic Engineering

As mentioned in Section 1.1, Internet traffic engineering provides performance optimization of IP networks while utilizing network resources economically and reliably. Such optimization is supported at the control/controller level and within the data/forwarding plane.

The key elements required in any TE solution are as follows:

1. Policy
2. Path steering
3. Resource management

Some TE solutions rely on these elements to a lesser or greater extent. Debate remains about whether a solution can truly be called TE if it does not include all of these elements. For the sake of this document, we assert that all TE solutions must include some aspects of all of these elements. Other solutions can be classed as "partial TE" and also fall in scope of this document.

Policy allows for the selection of paths (including next hops) based on information beyond basic reachability. Early definitions of routing policy, e.g., [RFC1102] and [RFC1104], discuss routing policy being applied to restrict access to network resources at an aggregate

level. BGP is an example of a commonly used mechanism for applying such policies, see [RFC4271] and [RFC8955]. In the TE context, policy decisions are made within the control plane or by controllers in the management plane, and govern the selection of paths. Examples can be found in [RFC4655] and [RFC5394]. Standard TE solutions may cover the mechanisms to distribute and/or enforce policies, but specific policy definition is generally unspecified.

Path steering is the ability to forward packets using more information than just knowledge of the next hop. Examples of path steering include IPv4 source routes [RFC0791], RSVP-TE explicit routes [RFC3209], Segment Routing [RFC8402], and Service Function Chaining [RFC7665]. Path steering for TE can be supported via control plane protocols, by encoding in the data plane headers, or by a combination of the two. This includes when control is provided by a controller using a network-facing control protocol.

Resource management provides resource-aware control and forwarding. Examples of resources are bandwidth, buffers, and queues, all of which can be managed to control loss and latency.

- \* Resource reservation is the control aspect of resource management. It provides for domain-wide consensus about which network resources are used by a particular flow. This determination may be made at a very coarse or very fine level. Note that this consensus exists at the network control or controller level, not within the data plane. It may be composed purely of accounting/bookkeeping, but it typically includes an ability to admit, reject, or reclassify a flow based on policy. Such accounting can be done based on any combination of a static understanding of resource requirements, and the use of dynamic mechanisms to collect requirements (e.g., via [RFC3209]) and resource availability (e.g., via [RFC4203]).

- \* Resource allocation is the data plane aspect of resource management. It provides for the allocation of specific node and link resources to specific flows. Example resources include buffers, policing, and rate-shaping mechanisms that are typically supported via queuing. It also includes the matching of a flow (i.e., flow classification) to a particular set of allocated resources. The method of flow classification and granularity of resource management is technology specific. Examples include Diffserv with dropping and remarking [RFC4594], MPLS-TE [RFC3209], and GMPLS based label switched paths [RFC3945], as well as controller-based solutions [RFC8453]. This level of resource control, while optional, is important in networks that wish to support congestion management policies to control or regulate the offered traffic to deliver different levels of service and alleviate congestion problems, or those networks that wish to control the latency experienced by specific traffic flows.

### 1.3. Scope

The scope of this document is intra-domain TE. That is, TE within a given autonomous system in the Internet. This document discusses concepts pertaining to intra-domain traffic control, including such issues as routing control, micro and macro resource allocation, and the control coordination problems that arise consequently.

This document describes and characterizes techniques already in use or in advanced development for Internet TE. The way these techniques fit together is discussed and scenarios in which they are useful will be identified.

Although the emphasis in this document is on intra-domain traffic engineering, an overview of the high level considerations pertaining to inter-domain TE is provided in Section 7. Inter-domain Internet TE is crucial to the performance enhancement of the global Internet infrastructure.

Whenever possible, relevant requirements from existing IETF documents and other sources are incorporated by reference.

### 1.4. Terminology

This section provides terminology which is useful for Internet TE. The definitions presented apply to this document. These terms may have other meanings elsewhere.

**Busy hour:** A one hour period within a specified interval of time (typically 24 hours) in which the traffic load in a network or sub-network is greatest.

- Congestion:** A state of a network resource in which the traffic incident on the resource exceeds its output capacity over an interval of time.
- Congestion avoidance:** An approach to congestion management that attempts to obviate the occurrence of congestion.
- Congestion control:** An approach to congestion management that attempts to remedy congestion problems that have already occurred.
- Constraint-based routing:** A class of routing protocols that take specified traffic attributes, network constraints, and policy constraints into account when making routing decisions. Constraint-based routing is applicable to traffic aggregates as well as flows. It is a generalization of QoS-based routing.
- Demand side congestion management:** A congestion management scheme that addresses congestion problems by regulating or conditioning offered load.
- Effective bandwidth:** The minimum amount of bandwidth that can be assigned to a flow or traffic aggregate in order to deliver 'acceptable service quality' to the flow or traffic aggregate.
- Hot-spot:** A network element or subsystem which is in a state of congestion.
- Inter-domain traffic:** Traffic that originates in one Autonomous system and terminates in another.
- Metric:** A parameter defined in terms of standard units of measurement.
- Measurement methodology:** A repeatable measurement technique used to derive one or more metrics of interest.
- Network survivability:** The capability to provide a prescribed level of QoS for existing services after a given number of failures occur within the network.
- Offline traffic engineering:** A traffic engineering system that exists outside of the network.
- Online traffic engineering:** A traffic engineering system that exists within the network, typically implemented on or as adjuncts to operational network elements.
- Performance measures:** Metrics that provide quantitative or



qualitative measures of the performance of systems or subsystems of interest.

**Performance metric:** A performance parameter defined in terms of standard units of measurement.

**Provisioning:** The process of assigning or configuring network resources to meet certain requests.

**QoS routing:** Class of routing systems that selects paths to be used by a flow based on the QoS requirements of the flow.

**Service Level Agreement (SLA):** A contract between a provider and a customer that guarantees specific levels of performance and reliability at a certain cost.

**Service Level Objective (SLO):** A key element of an SLA between a provider and a customer. SLOs are agreed upon as a means of measuring the performance of the Service Provider and are outlined as a way of avoiding disputes between the two parties based on misunderstanding.

**Stability:** An operational state in which a network does not oscillate in a disruptive manner from one mode to another mode.

**Supply-side congestion management:** A congestion management scheme that provisions additional network resources to address existing and/or anticipated congestion problems.

**Traffic characteristic:** A description of the temporal behavior or a description of the attributes of a given traffic flow or traffic aggregate.

**Traffic engineering system:** A collection of objects, mechanisms, and protocols that are used together to accomplish traffic engineering objectives.

**Traffic flow:** A stream of packets between two end-points that can be characterized in a certain way. A common classification for a traffic flow selects packets with the "five-tuple" of source and destination addresses, source and destination ports, and protocol ID.

**Traffic matrix:** A representation of the traffic demand between a set of origin and destination abstract nodes. An abstract node can consist of one or more network elements.

**Traffic monitoring:** The process of observing traffic characteristics

at a given point in a network and collecting the traffic information for analysis and further action.

**Traffic trunk:** An aggregation of traffic flows belonging to the same class which are forwarded through a common path. A traffic trunk may be characterized by an ingress and egress node, and a set of attributes which determine its behavioral characteristics and requirements from the network.

## 2. Background

The Internet aims to convey IP packets from ingress nodes to egress nodes efficiently, expeditiously, and economically. Furthermore, in a multiclass service environment (e.g., Diffserv capable networks - see Section 5.1.1.2), the resource sharing parameters of the network must be appropriately determined and configured according to prevailing policies and service models to resolve resource contention issues arising from mutual interference between packets traversing through the network. Thus, consideration must be given to resolving competition for network resources between traffic flows belonging to the same service class (intra-class contention resolution) and traffic flows belonging to different classes (inter-class contention resolution).

### 2.1. Context of Internet Traffic Engineering

The context of Internet traffic engineering includes:

1. A network domain context that defines the scope under consideration, and in particular the situations in which the TE problems occur. The network domain context includes network structure, network policies, network characteristics, network constraints, network quality attributes, and network optimization criteria.
2. A problem context defining the general and concrete issues that TE addresses. The problem context includes identification, abstraction of relevant features, representation, formulation, specification of the requirements on the solution space, and specification of the desirable features of acceptable solutions.
3. A solution context suggesting how to address the issues identified by the problem context. The solution context includes analysis, evaluation of alternatives, prescription, and resolution.

4. An implementation and operational context in which the solutions are instantiated. The implementation and operational context includes planning, organization, and execution.

The context of Internet TE and the different problem scenarios are discussed in the following subsections.

## 2.2. Network Domain Context

IP networks range in size from small clusters of routers situated within a given location, to thousands of interconnected routers, switches, and other components distributed all over the world.

At the most basic level of abstraction, an IP network can be represented as a distributed dynamic system consisting of:

- \* a set of interconnected resources which provide transport services for IP traffic subject to certain constraints
- \* a demand system representing the offered load to be transported through the network
- \* a response system consisting of network processes, protocols, and related mechanisms which facilitate the movement of traffic through the network (see also [AWD2]).

The network elements and resources may have specific characteristics restricting the manner in which the traffic demand is handled. Additionally, network resources may be equipped with traffic control mechanisms managing the way in which the demand is serviced. Traffic control mechanisms may be used to:

- \* control packet processing activities within a given resource
- \* arbitrate contention for access to the resource by different packets
- \* regulate traffic behavior through the resource.

A configuration management and provisioning system may allow the settings of the traffic control mechanisms to be manipulated by external or internal entities in order to exercise control over the way in which the network elements respond to internal and external stimuli.

The details of how the network carries packets are specified in the policies of the network administrators and are installed through network configuration management and policy based provisioning

systems. Generally, the types of service provided by the network also depend upon the technology and characteristics of the network elements and protocols, the prevailing service and utility models, and the ability of the network administrators to translate policies into network configurations.

Internet networks have two significant characteristics:

- \* they provide real-time services
- \* their operating environments are very dynamic.

The dynamic characteristics of IP and IP/MPLS networks can be attributed in part to fluctuations in demand, to the interaction between various network protocols and processes, to the rapid evolution of the infrastructure which demands the constant inclusion of new technologies and new network elements, and to transient and persistent faults which occur within the system.

Packets contend for the use of network resources as they are conveyed through the network. A network resource is considered to be congested if, for an interval of time, the arrival rate of packets exceed the output capacity of the resource. Congestion may result in some of the arriving packets being delayed or even dropped.

Congestion increases transit delay, delay variation, may lead to packet loss, and reduces the predictability of network services. Clearly, congestion is highly undesirable. Combating congestion at a reasonable cost is a major objective of Internet TE.

Efficient sharing of network resources by multiple traffic flows is a basic operational premise for the Internet. A fundamental challenge in network operation is to increase resource utilization while minimizing the possibility of congestion.

The Internet has to function in the presence of different classes of traffic with different service requirements. This requirement is clarified in [RFC2475] which also provides an architecture for Differentiated Services (Diffserv). That document describes how packets can be grouped into behavior aggregates such that each aggregate has a common set of behavioral characteristics or a common set of delivery requirements. Delivery requirements of a specific set of packets may be specified explicitly or implicitly. Two of the most important traffic delivery requirements are:

- \* Capacity constraints can be expressed statistically as peak rates, mean rates, burst sizes, or as some deterministic notion of effective bandwidth.

- \* QoS requirements can be expressed in terms of:
  - integrity constraints such as packet loss
  - temporal constraints such as timing restrictions for the delivery of each packet (delay) and timing restrictions for the delivery of consecutive packets belonging to the same traffic stream (delay variation).

### 2.3. Problem Context

There are several problems associated with operating a network described in the previous section. This section analyzes the problem context in relation to TE. The identification, abstraction, representation, and measurement of network features relevant to TE are significant issues.

A particular challenge is to formulate the problems that traffic engineering attempts to solve. For example:

- \* How to identify the requirements on the solution space?
- \* How to specify the desirable features of solutions?
- \* How to actually solve the problems?
- \* How to measure and characterize the effectiveness of solutions?

Another class of problems is how to measure and estimate relevant network state parameters. Effective TE relies on a good estimate of the offered traffic load as well as a view of the underlying topology and associated resource constraints. A network-wide view of the topology is also a must for offline planning.

Still another class of problem is how to characterize the state of the network and how to evaluate its performance. The performance evaluation problem is two-fold: one aspect relates to the evaluation of the system-level performance of the network; the other aspect relates to the evaluation of resource-level performance, which restricts attention to the performance analysis of individual network resources.

In this document, we refer to the system-level characteristics of the network as the "macro-states" and the resource-level characteristics as the "micro-states." The system-level characteristics are also known as the emergent properties of the network. Correspondingly, we refer to the TE schemes dealing with network performance optimization at the systems level as "macro-TE" and the schemes that optimize at

the individual resource level as "micro-TE." Under certain circumstances, the system-level performance can be derived from the resource-level performance using appropriate rules of composition, depending upon the particular performance measures of interest.

Another fundamental class of problem concerns how to effectively optimize network performance. Performance optimization may entail translating solutions for specific TE problems into network configurations. Optimization may also entail some degree of resource management control, routing control, and capacity augmentation.

#### 2.3.1. Congestion and its Ramifications

Congestion is one of the most significant problems in an operational IP context. A network element is said to be congested if it experiences sustained overload over an interval of time. Congestion almost always results in degradation of service quality to end users. Congestion control schemes can include demand-side policies and supply-side policies. Demand-side policies may restrict access to congested resources or dynamically regulate the demand to alleviate the overload situation. Supply-side policies may expand or augment network capacity to better accommodate offered traffic. Supply-side policies may also re-allocate network resources by redistributing traffic over the infrastructure. Traffic redistribution and resource re-allocation serve to increase the 'effective capacity' of the network.

The emphasis of this document is primarily on congestion management schemes falling within the scope of the network, rather than on congestion management systems dependent upon sensitivity and adaptivity from end-systems. That is, the aspects that are considered in this document with respect to congestion management are those solutions that can be provided by control entities operating on the network and by the actions of network administrators and network operations systems.

#### 2.4. Solution Context

The solution context for Internet TE involves analysis, evaluation of alternatives, and choice between alternative courses of action. Generally, the solution context is based on making inferences about the current or future state of the network, and making decisions that may involve a preference between alternative sets of action. More specifically, the solution context demands reasonable estimates of traffic workload, characterization of network state, derivation of solutions which may be implicitly or explicitly formulated, and possibly instantiating a set of control actions. Control actions may involve the manipulation of parameters associated with routing,

control over tactical capacity acquisition, and control over the traffic management functions.

The following list of instruments may be applicable to the solution context of Internet TE.

- \* A set of policies, objectives, and requirements (which may be context dependent) for network performance evaluation and performance optimization.
- \* A collection of online and possibly offline tools and mechanisms for measurement, characterization, modeling, and control traffic, and control over the placement and allocation of network resources, as well as control over the mapping or distribution of traffic onto the infrastructure.
- \* A set of constraints on the operating environment, the network protocols, and the TE system itself.
- \* A set of quantitative and qualitative techniques and methodologies for abstracting, formulating, and solving TE problems.
- \* A set of administrative control parameters which may be manipulated through a configuration management system. Such system itself may include a configuration control subsystem, a configuration repository, a configuration accounting subsystem, and a configuration auditing subsystem.
- \* A set of guidelines for network performance evaluation, performance optimization, and performance improvement.

Determining traffic characteristics through measurement or estimation is very useful within the realm the TE solution space. Traffic estimates can be derived from customer subscription information, traffic projections, traffic models, and from actual measurements. The measurements may be performed at different levels, e.g., at the traffic-aggregate level or at the flow level. Measurements at the flow level or on small traffic aggregates may be performed at edge nodes, when traffic enters and leaves the network. Measurements for large traffic-aggregates may be performed within the core of the network.

To conduct performance studies and to support planning of existing and future networks, a routing analysis may be performed to determine the paths the routing protocols will choose for various traffic demands, and to ascertain the utilization of network resources as traffic is routed through the network. Routing analysis captures the selection of paths through the network, the assignment of traffic

across multiple feasible routes, and the multiplexing of IP traffic over traffic trunks (if such constructs exist) and over the underlying network infrastructure. A model of network topology is necessary to perform routing analysis. A network topology model may be extracted from:

- \* network architecture documents
- \* network designs
- \* information contained in router configuration files
- \* routing databases
- \* routing tables
- \* automated tools that discover and collate network topology information.

Topology information may also be derived from servers that monitor network state, and from servers that perform provisioning functions.

Routing in operational IP networks can be administratively controlled at various levels of abstraction including the manipulation of BGP attributes and interior gateway protocol (IGP) metrics. For path oriented technologies such as MPLS, routing can be further controlled by the manipulation of relevant TE parameters, resource parameters, and administrative policy constraints. Within the context of MPLS, the path of an explicitly routed label switched path (LSP) can be computed and established in various ways including:

- \* manually
- \* automatically, online using constraint-based routing processes implemented on label switching routers
- \* automatically, offline using constraint-based routing entities implemented on external TE support systems.

#### 2.4.1. Combating the Congestion Problem

Minimizing congestion is a significant aspect of Internet traffic engineering. This subsection gives an overview of the general approaches that have been used or proposed to combat congestion.

Congestion management policies can be categorized based upon the following criteria (see [YARE95] for a more detailed taxonomy of congestion control schemes):



## 1. Congestion Management Based on Response Time Scales

- \* Long (weeks to months): Expanding network capacity by adding new equipment, routers, and links takes time and is comparatively costly. Capacity planning needs to take this into consideration. Network capacity is expanded based on estimates or forecasts of future traffic development and traffic distribution. These upgrades are typically carried out over weeks or months, or maybe even years.
- \* Medium (minutes to days): Several control policies fall within the medium timescale category. Examples include:
  - a. Adjusting routing protocol parameters to route traffic away or towards certain segments of the network.
  - b. Setting up or adjusting explicitly routed LSPs in MPLS networks to route traffic trunks away from possibly congested resources or toward possibly more favorable routes.
  - c. Re-configuring the logical topology of the network to make it correlate more closely with the spatial traffic distribution using, for example, an underlying path-oriented technology such as MPLS LSPs or optical channel trails.

Many of these adaptive schemes rely on measurement systems. A measurement system monitors changes in traffic distribution, traffic loads, and network resource utilization and then provides feedback to the online or offline TE mechanisms and tools so that they can trigger control actions within the network. The TE mechanisms and tools can be implemented in a distributed or centralized fashion. A centralized scheme may have global visibility into the network state and may produce more optimal solutions. However, centralized schemes are prone to single points of failure and may not scale as well as distributed schemes. Moreover, the information utilized by a centralized scheme may be stale and might not reflect the actual state of the network. It is not an objective of this document to make a recommendation between distributed and centralized schemes: that is a choice that network administrators must make based on their specific needs.

- \* Short (picoseconds to minutes): This category includes packet level processing functions and events that are recorded on the order of several round trip times. It also includes router mechanisms such as passive and active buffer management. All

of these mechanisms are used to control congestion or signal congestion to end systems so that they can adaptively regulate the rate at which traffic is injected into the network. One of the most popular active queue management schemes, especially for TCP traffic, is Random Early Detection (RED) [FLJA93]. During congestion (but before the queue is filled), the RED scheme chooses arriving packets to "mark" according to a probabilistic algorithm which takes into account the average queue size. A router that does not utilize explicit congestion notification (ECN) [FLOY94] can simply drop marked packets to alleviate congestion and implicitly notify the receiver about the congestion. On the other hand, if the router supports ECN, it can set the ECN field in the packet header. Several variations of RED have been proposed to support different drop precedence levels in multi-class environments [RFC2597]. RED provides congestion avoidance which is not worse than traditional Tail-Drop (TD) queue management (drop arriving packets only when the queue is full). Importantly, RED reduces the possibility of global synchronization where retransmission burst become synchronized across the whole network, and improves fairness among different TCP sessions. However, RED by itself cannot prevent congestion and unfairness caused by sources unresponsive to RED, e.g., UDP traffic and some misbehaved greedy connections. Other schemes have been proposed to improve the performance and fairness in the presence of unresponsive traffic. Some of those schemes (such as Longest Queue Drop (LQD) and Dynamic Soft Partitioning with Random Drop (RND) [SLDC98]) were proposed as theoretical frameworks and are typically not available in existing commercial products. Advice on the use of Active Queue Management (AQM) schemes is provided in [RFC7567].

## 2. Reactive Versus Preventive Congestion Management Schemes

- \* Reactive (recovery) congestion management policies react to existing congestion problems. All the policies described above for the long and medium time scales can be categorized as being reactive. They are based on monitoring and identifying congestion problems that exist in the network, and on the initiation of relevant actions to ease a situation.
- \* Preventive (predictive/avoidance) policies take proactive action to prevent congestion based on estimates and predictions of future congestion problems (e.g., traffic matrix forecasts). Some of the policies described for the long and medium time scales fall into this category. Preventive policies do not necessarily respond immediately to

existing congestion problems. Instead, forecasts of traffic demand and workload distribution are considered, and action may be taken to prevent potential future congestion problems. The schemes described for the short time scale can also be used for congestion avoidance because dropping or marking packets before queues actually overflow would trigger corresponding TCP sources to slow down.

### 3. Supply-Side Versus Demand-Side Congestion Management Schemes

- \* Supply-side congestion management policies increase the effective capacity available to traffic in order to control or reduce congestion. This can be accomplished by increasing capacity or by balancing distribution of traffic over the network. Capacity planning aims to provide a physical topology and associated link bandwidths that match or exceed estimated traffic workload and traffic distribution subject to traffic forecasts and budgetary or other constraints. If the actual traffic distribution does not fit the topology derived from capacity planning, then the traffic can be mapped onto the topology by using routing control mechanisms, by applying path oriented technologies (e.g., MPLS LSPs and optical channel trails) to modify the logical topology, or by employing some other load redistribution mechanisms.
- \* Demand-side congestion management policies control or regulate the offered traffic to alleviate congestion problems. For example, some of the short time scale mechanisms described earlier as well as policing and rate-shaping mechanisms attempt to regulate the offered load in various ways.

#### 2.5. Implementation and Operational Context

The operational context of Internet TE is characterized by constant changes that occur at multiple levels of abstraction. The implementation context demands effective planning, organization, and execution. The planning aspects may involve determining prior sets of actions to achieve desired objectives. Organizing involves arranging and assigning responsibility to the various components of the TE system and coordinating the activities to accomplish the desired TE objectives. Execution involves measuring and applying corrective or perfective actions to attain and maintain desired TE goals.

### 3. Traffic Engineering Process Models

This section describes a generic process model that captures the high-level practical aspects of Internet traffic engineering in an operational context. The process model is described as a sequence of actions that must be carried out to optimize the performance of an operational network (see also [RFC2702], [AWD2]). This process model may be enacted explicitly or implicitly, by a software process or by a human.

The TE process model is iterative [AWD2]. The four phases of the process model described below are repeated as a continual sequence.

- \* Define the relevant control policies that govern the operation of the network.
- \* Acquire measurement data from the operational network.
- \* Analyze the network state and characterize the traffic workload. Proactive analysis identifies potential problems that could manifest in the future. Reactive analysis identifies existing problems and determines their causes.
- \* Optimize the performance of the network. This involves a decision process which selects and implements a set of actions from a set of alternatives given the results of the three previous steps. Optimization actions may include the use of techniques to control the offered traffic and to control the distribution of traffic across the network.

#### 3.1. Components of the Traffic Engineering Process Model

The key components of the traffic engineering process model are as follows.

1. Measurement is crucial to the TE function. The operational state of a network can only be conclusively determined through measurement. Measurement is also critical to the optimization function because it provides feedback data which is used by TE control subsystems. This data is used to adaptively optimize network performance in response to events and stimuli originating within and outside the network. Measurement in support of the TE function can occur at different levels of abstraction. For example, measurement can be used to derive packet level characteristics, flow level characteristics, user or customer level characteristics, traffic aggregate characteristics, component level characteristics, and network wide characteristics.

2. Modeling, analysis, and simulation are important aspects of Internet TE. Modeling involves constructing an abstract or physical representation which depicts relevant traffic characteristics and network attributes. A network model is an abstract representation of the network which captures relevant network features, attributes, and characteristic. Network simulation tools are extremely useful for TE. Because of the complexity of realistic quantitative analysis of network behavior, certain aspects of network performance studies can only be conducted effectively using simulation.
3. Network performance optimization involves resolving network issues by transforming such issues into concepts that enable a solution, identification of a solution, and implementation of the solution. Network performance optimization can be corrective or perfective. In corrective optimization, the goal is to remedy a problem that has occurred or that is incipient. In perfective optimization, the goal is to improve network performance even when explicit problems do not exist and are not anticipated.

#### 4. Taxonomy of Traffic Engineering Systems

This section presents a short taxonomy of traffic engineering systems constructed based on TE styles and views as listed below and described in greater detail in the following subsections of this document.

- \* Time-dependent versus State-dependent versus Event-dependent
- \* Offline versus Online
- \* Centralized versus Distributed
- \* Local versus Global Information
- \* Prescriptive versus Descriptive
- \* Open Loop versus Closed Loop
- \* Tactical versus Strategic

##### 4.1. Time-Dependent Versus State-Dependent Versus Event-Dependent

Traffic engineering methodologies can be classified as time-dependent, state-dependent, or event-dependent. All TE schemes are considered to be dynamic in this document. Static TE implies that no TE methodology or algorithm is being applied - it is a feature of network planning, but lacks the reactive and flexible nature of TE.

In time-dependent TE, historical information based on periodic variations in traffic (such as time of day) is used to pre-program routing and other TE control mechanisms. Additionally, customer subscription or traffic projection may be used. Pre-programmed routing plans typically change on a relatively long time scale (e.g., daily). Time-dependent algorithms do not attempt to adapt to short-term variations in traffic or changing network conditions. An example of a time-dependent algorithm is a global centralized optimizer where the input to the system is a traffic matrix and multi-class QoS requirements as described [MR99]. Another example of such a methodology is the application of data mining to Internet traffic [AJ19] which enables the use of various machine learning algorithms to identify patterns within historically collected datasets about Internet traffic, and to extract information in order to guide decision-making, and to improve efficiency and productivity of operational processes.

State-dependent TE adapts the routing plans based on the current state of the network which provides additional information on variations in actual traffic (i.e., perturbations from regular variations) that could not be predicted using historical information. Constraint-based routing is an example of state-dependent TE operating in a relatively long time scale. An example operating in a relatively short timescale is a load-balancing algorithm described in [MATE]. The state of the network can be based on parameters flooded by the routers. Another approach is for a particular router performing adaptive TE to send probe packets along a path to gather the state of that path. [RFC6374] defines protocol extensions to collect performance measurements from MPLS networks. Another approach is for a management system to gather the relevant information directly from network elements using telemetry data collection "publication/subscription" techniques [RFC7923]. Timely gathering and distribution of state information is critical for adaptive TE. While time-dependent algorithms are suitable for predictable traffic variations, state-dependent algorithms may be applied to increase network efficiency and resilience to adapt to the prevailing network state.

Event-dependent TE methods can also be used for TE path selection. Event-dependent TE methods are distinct from time-dependent and state-dependent TE methods in the manner in which paths are selected. These algorithms are adaptive and distributed in nature and typically use learning models to find good paths for TE in a network. While state-dependent TE models typically use available-link-bandwidth (ALB) flooding for TE path selection, event-dependent TE methods do not require ALB flooding. Rather, event-dependent TE methods typically search out capacity by learning models, as in the success-to-the-top (STT) method. ALB flooding can be resource intensive,

since it requires link bandwidth to carry LSAs, processor capacity to process LSAs, and the overhead can limit area/Autonomous System (AS) size. Modeling results suggest that event-dependent TE methods could lead to a reduction in ALB flooding overhead without loss of network throughput performance [I-D.ietf-tewg-qos-routing].

#### 4.2. Offline Versus Online

Traffic engineering requires the computation of routing plans. The computation may be performed offline or online. The computation can be done offline for scenarios where routing plans need not be executed in real-time. For example, routing plans computed from forecast information may be computed offline. Typically, offline computation is also used to perform extensive searches on multi-dimensional solution spaces.

Online computation is required when the routing plans must adapt to changing network conditions as in state-dependent algorithms. Unlike offline computation (which can be computationally demanding), online computation is geared toward relative simple and fast calculations to select routes, fine-tune the allocations of resources, and perform load balancing.

#### 4.3. Centralized Versus Distributed

Under centralized control there is a central authority which determines routing plans and perhaps other TE control parameters on behalf of each router. The central authority periodically collects network-state information from all routers, and sends routing information to the routers. The update cycle for information exchange in both directions is a critical parameter directly impacting the performance of the network being controlled. Centralized control may need high processing power and high bandwidth control channels.

Distributed control determines route selection by each router autonomously based on the router's view of the state of the network. The network state information may be obtained by the router using a probing method or distributed by other routers on a periodic basis using link state advertisements. Network state information may also be disseminated under exception conditions. Examples of protocol extensions used to advertise network link state information are defined in [RFC5305], [RFC6119], [RFC7471], [RFC8570], and [RFC8571]. See also Section 5.1.3.8.

#### 4.3.1. Hybrid Systems

In practice, most TE systems will be a hybrid of central and distributed control. For example, a popular MPLS approach to TE is to use a central controller based on an active, stateful PCE, but to use routing and signaling protocols to make local decisions at routers within the network. Local decisions may be able to respond more quickly to network events, but may result in conflicts with decisions made by other routers.

Network operations for TE systems may also use a hybrid of offline and online computation. TE paths may be precomputed based on stable-state network information and planned traffic demands, but may then be modified in the active network depending on variations in network state and traffic load. Furthermore, responses to network events may be precomputed offline to allow rapid reactions without further computation, or may be derived online depending on the nature of the events.

Lastly, note that a fully functional TE system is likely to use all aspects of time-dependent, state-dependent, and event-dependent methodologies as described in Section 4.1.

#### 4.3.2. Considerations for Software Defined Networking

As discussed in Section 5.1.2.2, one of the main drivers for SDN is a decoupling of the network control plane from the data plane [RFC7149]. However, SDN may also combine centralized control of resources, and facilitate application-to-network interaction via an application programming interface (API) such as [RFC8040]. Combining these features provides a flexible network architecture that can adapt to network requirements of a variety of higher-layer applications, a concept often referred to as the "programmable network" [RFC7426].

The centralized control aspect of SDN helps improve global network resource utilization compared with distributed network control, where local policy may often override global optimization goals. In an SDN environment, the data plane forwards traffic to its desired destination. However, before traffic reaches the data plane, the logically centralized SDN control plane often determines the end-to-end path the application traffic will take in the network. Therefore, the SDN control plane needs to be aware of the underlying network topology, capabilities and current node and link resource state.



Using a PCE-based SDN control framework [RFC7491], the available network topology may be discovered by running a passive instance of OSPF or IS-IS, or via BGP-LS [RFC7752], to generate a TED (see Section 5.1.3.12). The PCE is used to compute a path (see Section 5.1.3.10) based on the TED and available bandwidth, and further path optimization may be based on requested objective functions [RFC5541]. When a suitable path has been computed the programming of the explicit network path may be performed using either end-to-end signaling protocol [RFC3209] or per-hop with each node being directly programmed [RFC8283] by the SDN controller.

By utilizing a centralized approach to network control, additional network benefits are also available, including Global Concurrent Optimization (GCO) [RFC5557]. A GCO path computation request will simultaneously use the network topology and set of new end-to-end path requests, along with their respective constraints, for optimal placement in the network. Correspondingly, a GCO-based computation may be applied to recompute existing network paths to groom traffic and to mitigate congestion.

#### 4.4. Local Versus Global

Traffic engineering algorithms may require local and global network-state information.

Local information is the state of a portion of the domain. Examples include the bandwidth and packet loss rate of a particular path, or the state and capabilities of a network link. Local state information may be sufficient for certain instances of distributed control TE.

Global information is the state of the entire TE domain. Examples include a global traffic matrix, and loading information on each link throughout the domain of interest. Global state information is typically required with centralized control. Distributed TE systems may also need global information in some cases.

#### 4.5. Prescriptive Versus Descriptive

TE systems may also be classified as prescriptive or descriptive.

Prescriptive traffic engineering evaluates alternatives and recommends a course of action. Prescriptive TE can be further categorized as either corrective or perfective. Corrective TE prescribes a course of action to address an existing or predicted anomaly. Perfective TE prescribes a course of action to evolve and improve network performance even when no anomalies are evident.

Descriptive traffic engineering, on the other hand, characterizes the state of the network and assesses the impact of various policies without recommending any particular course of action.

#### 4.5.1. Intent-Based Networking

One way to express a service request is through "intent". Intent-Based Networking aims to produce networks that are simpler to manage and operate, requiring only minimal intervention. Intent is defined in [I-D.irtf-nmrg-ibn-concepts-definitions] as a set of operational goals (that a network should meet) and outcomes (that a network is supposed to deliver), defined in a declarative manner without specifying how to achieve or implement them.

Intent provides data and functional abstraction so that users and operators do not need to be concerned with low-level device configuration or the mechanisms used to achieve a given intent. This approach can be conceptually easier for a user, but may be less expressive in terms of constraints and guidelines.

Intent-Based Networking is applicable to TE because many of the high-level objectives may be expressed as "intent." For example, load balancing, delivery of services, and robustness against failures. The intent is converted by the management system into TE actions within the network.

#### 4.6. Open-Loop Versus Closed-Loop

Open-loop traffic engineering control is where control action does not use feedback information from the current network state. The control action may use its own local information for accounting purposes, however.

Closed-loop traffic engineering control is where control action utilizes feedback information from the network state. The feedback information may be in the form of historical information or current measurement.

#### 4.7. Tactical versus Strategic

Tactical traffic engineering aims to address specific performance problems (such as hot-spots) that occur in the network from a tactical perspective, without consideration of overall strategic imperatives. Without proper planning and insights, tactical TE tends to be ad hoc in nature.

Strategic traffic engineering approaches the TE problem from a more organized and systematic perspective, taking into consideration the immediate and longer term consequences of specific policies and actions.

## 5. Review of TE Techniques

This section briefly reviews different TE-related approaches proposed and implemented in telecommunications and computer networks using IETF protocols and architectures. These approaches are organized into three categories:

- \* TE mechanisms which adhere to the definition provided in Section 1.2.
- \* Approaches that rely upon those TE mechanisms.
- \* Techniques that are used by those TE mechanisms and approaches.

The discussion is not intended to be comprehensive. It is primarily intended to illuminate existing approaches to TE in the Internet. A historic overview of TE in telecommunications networks is provided in Appendix A, while Appendix B describes approaches in other standards bodies.

### 5.1. Overview of IETF Projects Related to Traffic Engineering

This subsection reviews a number of IETF activities pertinent to Internet traffic engineering.

#### 5.1.1. IETF TE Mechanisms

##### 5.1.1.1. Integrated Services

The IETF developed the Integrated Services (Intserv) model that requires resources, such as bandwidth and buffers, to be reserved a priori for a given traffic flow to ensure that the quality of service requested by the traffic flow is satisfied. The Integrated Services model includes additional components beyond those used in the best-effort model such as packet classifiers, packet schedulers, and admission control. A packet classifier is used to identify flows that are to receive a certain level of service. A packet scheduler handles the scheduling of service to different packet flows to ensure that QoS commitments are met. Admission control is used to determine whether a router has the necessary resources to accept a new flow.

The main issue with the Integrated Services model has been scalability [RFC2998], especially in large public IP networks which may potentially have millions of active traffic flows in transit concurrently.

A notable feature of the Integrated Services model is that it requires explicit signaling of QoS requirements from end systems to routers [RFC2753]. The Resource Reservation Protocol (RSVP) performs this signaling function and is a critical component of the Integrated Services model. RSVP is described in Section 5.1.3.2.

#### 5.1.1.2. Differentiated Services

The goal of Differentiated Services (Diffserv) within the IETF was to devise scalable mechanisms for categorization of traffic into behavior aggregates, which ultimately allows each behavior aggregate to be treated differently, especially when there is a shortage of resources such as link bandwidth and buffer space [RFC2475]. One of the primary motivations for Diffserv was to devise alternative mechanisms for service differentiation in the Internet that mitigate the scalability issues encountered with the Intserv model.

Diffserv uses the Differentiated Services field in the IP header (the DS field) consisting of six bits in what was formerly known as the Type of Service (TOS) octet. The DS field is used to indicate the forwarding treatment that a packet should receive at a transit node [RFC2474]. Diffserv includes the concept of Per-Hop Behavior (PHB) groups. Using the PHBs, several classes of services can be defined using different classification, policing, shaping, and scheduling rules.

For an end-user of network services to utilize Differentiated Services provided by its Internet Service Provider (ISP), it may be necessary for the user to have an SLA with the ISP. An SLA may explicitly or implicitly specify a Traffic Conditioning Agreement (TCA) which defines classifier rules as well as metering, marking, discarding, and shaping rules.

Packets are classified, and possibly policed and shaped at the ingress to a Diffserv network. When a packet traverses the boundary between different Diffserv domains, the DS field of the packet may be re-marked according to existing agreements between the domains.

Differentiated Services allows only a finite number of service classes to be specified by the DS field. The main advantage of the Diffserv approach relative to the Intserv model is scalability. Resources are allocated on a per-class basis and the amount of state information is proportional to the number of classes rather than to the number of application flows.

The Diffserv model deals with traffic management issues on a per hop basis. The Diffserv control model consists of a collection of micro-TE control mechanisms. Other TE capabilities, such as capacity management (including routing control), are also required in order to deliver acceptable service quality in Diffserv networks. The concept of Per Domain Behaviors has been introduced to better capture the notion of Differentiated Services across a complete domain [RFC3086].

Diffserv procedures can also be applied in an MPLS context. See Section 6.8 for more information.

#### 5.1.1.3. Segment Routing Policy

SR Policy [I-D.ietf-spring-segment-routing-policy] is an evolution of Segment Routing (see Section 5.1.3.11) to enhance the TE capabilities of SR. It is a framework that enables instantiation of an ordered list of segments on a node for implementing a source routing policy with a specific intent for traffic steering from that node.

An SR Policy is identified through the tuple <head-end, color, endpoint>. The head-end is the IP address of the node where the policy is instantiated. The endpoint is the IP address of the destination of the policy. The color is an index that associates the SR Policy with an intent (e.g., low-latency).

The head-end node is notified of SR Policies and associated SR paths via configuration or by extensions to protocols such as PCEP [RFC8664] or BGP [I-D.ietf-idr-segment-routing-te-policy]. Each SR path consists of a Segment-List (an SR source-routed path), and the head-end uses the endpoint and color parameters to classify packets to match the SR policy and so determine along which path to forward them. If an SR Policy is associated with a set of SR paths, each is associated with a weight for weighted load balancing. Furthermore, multiple SR Policies may be associated with a set of SR paths to allow multiple traffic flows to be placed on the same paths.

An SR Binding SID (BSID) are also be associated with each candidate path associated with an SR Policy, or with the SR Policy itself. The head-end node installs a BSID-keyed entry in the forwarding plane and assigns it the action of steering packets that match the entry to the selected path of the SR Policy. This steering can be done in various ways:

- \* SID Steering: Incoming packets have an active SID matching a local BSID at the head-end.
- \* Per-destination Steering: Incoming packets match a BGP/Service route which indicates an SR Policy.
- \* Per-flow Steering: Incoming packets match a forwarding array (for example, the classic 5-tuple) which indicates an SR Policies.
- \* Policy-based Steering: Incoming packets match a routing policy which directs them to an SR Policy.

#### 5.1.1.4. Transport-Based TE

In addition to IP-based TE mechanisms, transport-based TE approaches can be considered in specific deployment contexts (e.g., data centers, multi-homing). For example, the 3GPP defines the Access Traffic Steering, Switching, and Splitting (ATSSS) [ATSSS] service functions as follows.

**Access Traffic Steering:** This is the selection of an access network for a new flow and the transfer of the traffic of that flow over the selected access network.

**Access Traffic Switching:** This is the migration of all packets of an ongoing flow from one access network to another access network. Only one access network is in use at a time.

**Access Traffic Splitting:** This is about forwarding the packets of a flow across multiple access networks simultaneously.

The control plane is used to provide hosts and specific network devices with a set or policies that specify which flows are eligible to use the ATSSS service. The traffic that matches an ATSSS policy can be distributed among the available access networks following one of the following four modes.

**Active-Standby:** The traffic is forwarded via a specific access (called "active access") and switched to another access (called "standby access") when the active access is unavailable.

Priority-based: Network accesses are assigned priority levels that indicate which network access is to be used first. The traffic associated with the matching flow will be steered onto the network access with the highest priority until congestion is detected, then the overflow will be forwarded over the next highest priority access.

Load-Balancing: The traffic is distributed among the available access networks following a distribution ratio (e.g., 75% - 25%).

Smallest Delay: The traffic is forwarded via the access that presents the smallest round-trip-time (RTT).

For resource management purposes, hosts and network devices support means such as congestion control, RTT measurement, and packet scheduling.

For TCP traffic, Multipath TCP [RFC8684] and the 0-RTT Convert Protocol [RFC8803] are used to provide the ATSSS service.

QUIC [RFC9000] is a UDP-based multiplexed and secure transport protocol. QUIC provides applications with flow-controlled streams for structured communication, low-latency connection establishment, and network path migration.

QUIC is a connection-oriented protocol that creates a stateful interaction between a client and server. QUIC uses a handshake procedure that combines negotiation of cryptographic and transport parameters. This is a key differentiation from other transport protocols.

With QUIC it is possible to support the ATSSS switching and steering functions, but splitting is not yet supported. Indeed, QUIC supports a connection migration procedure that allows peers to change their transport coordinates (IP addresses, port numbers) without breaking the underlying QUIC connection.

#### 5.1.1.5. Deterministic Networking

Deterministic Networking (DetNet) [RFC8655] is an architecture for applications with critical timing and reliability requirements. The layered architecture particularly focuses on developing DetNet service capabilities in the data plane [RFC8938]. The DetNet service sub-layer provides a set of Packet Replication, Elimination, and Ordering Functions (PREOF) functions to provide end-to-end service assurance. The DetNet forwarding sub-layer provides corresponding forwarding assurance (low packet loss, bounded latency, and in-order delivery) functions using resource allocations and explicit route

mechanisms.

The separation into two sub-layers allows a greater flexibility to adapt Detnet capability over a number of TE data plane mechanisms such as IP, MPLS, and Segment Routing. More importantly it interconnects IEEE 802.1 Time Sensitive Networking (TSN) [RFC9023] deployed in Industry Control and Automation Systems (ICAS).

DetNet can be seen as a specialized branch of TE, since it sets up explicit optimized paths with allocation of resources as requested. A DetNet application can express its QoS attributes or traffic behavior using any combination of DetNet functions described in sub-layers. They are then distributed and provisioned using well-established control and provisioning mechanisms adopted for traffic-engineering.

In DetNet, a considerable state information is required to maintain per flow queuing disciplines and resource reservation for a large number of individual flows. This can be quite challenging for network operations during network events such as faults, change in traffic volume or re-provisioning. Therefore, DetNet recommends support for aggregated flows, however, it still requires large amount of control signaling to establish and maintain DetNet flows.

#### 5.1.2. IETF Approaches Relying on TE Mechanisms

##### 5.1.2.1. Application-Layer Traffic Optimization

This document describes various TE mechanisms available in the network. However, distributed applications in general and, in particular, bandwidth-greedy P2P applications that are used, for example, for file sharing, cannot directly use those techniques. As per [RFC5693], applications could greatly improve traffic distribution and quality by cooperating with external services that are aware of the network topology. Addressing the Application-Layer Traffic Optimization (ALTO) problem means, on the one hand, deploying an ALTO service to provide applications with information regarding the underlying network (e.g., basic network location structure and preferences of network paths) and, on the other hand, enhancing applications in order to use such information to perform better-than-random selection of the endpoints with which they establish connections.

The basic function of ALTO is based on abstract maps of a network. These maps provide a simplified view, yet enough information about a network for applications to effectively utilize them. Additional services are built on top of the maps. [RFC7285] describes a protocol implementing the ALTO services as an information-publishing



interface that allows a network to publish its network information such as network locations, costs between them at configurable granularities, and end-host properties to network applications. The information published by the ALTO Protocol should benefit both the network and the applications. The ALTO Protocol uses a REST-ful design and encodes its requests and responses using JSON [RFC8259] with a modular design by dividing ALTO information publication into multiple ALTO services (e.g., the Map service, the Map-Filtering Service, the Endpoint Property Service, and the Endpoint Cost Service).

[RFC8189] defines a new service that allows an ALTO Client to retrieve several cost metrics in a single request for an ALTO filtered cost map and endpoint cost map. [RFC8896] extends the ALTO cost information service so that applications decide not only 'where' to connect, but also 'when'. This is useful for applications that need to perform bulk data transfer and would like to schedule these transfers during an off-peak hour, for example. [I-D.ietf-alto-performance-metrics] introducing network performance metrics, including network delay, jitter, packet loss rate, hop count, and bandwidth. The ALTO server may derive and aggregate such performance metrics from BGP-LS (see Section 5.1.3.9) or IGP-TE (see Section 5.1.3.8), or management tools, and then expose the information to allow applications to determine 'where' to connect based on network performance criteria. ALTO WG is evaluating the use of network TE properties while making application decisions for new use-cases such as Edge computing and Datacenter interconnect.

#### 5.1.2.2. Network Virtualization and Abstraction

One of the main drivers for Software Defined Networking (SDN) [RFC7149] is a decoupling of the network control plane from the data plane. This separation has been achieved for TE networks with the development of MPLS/GMPLS (see Section 5.1.3.3 and Section 5.1.3.4) and the Path Computation Element (PCE) (Section 5.1.3.10). One of the advantages of SDN is its logically centralized control regime that allows a global view of the underlying networks. Centralized control in SDN helps improve network resource utilization compared with distributed network control.

Abstraction and Control of TE Networks (ACTN) [RFC8453] defines a hierarchical SDN architecture which describes the functional entities and methods for the coordination of resources across multiple domains, to provide end-to-end traffic engineered services. ACTN facilitates end-to-end connections and provides them to the user. ACTN is focused on:

- \* Abstraction of the underlying network resources and how they are provided to higher-layer applications and customers.
- \* Virtualization of underlying resources for use by the customer, application, or service. The creation of a virtualized environment allows operators to view and control multi-domain networks as a single virtualized network.
- \* Presentation to customers of networks as a virtual network via open and programmable interfaces.

The ACTN managed infrastructure is built from traffic engineered network resources, which may include statistical packet bandwidth, physical forwarding plane sources (such as wavelengths and time slots), forwarding and cross-connect capabilities. The type of network virtualization seen in ACTN allows customers and applications (tenants) to utilize and independently control allocated virtual network resources as if resources as if they were physically their own resource. The ACTN network is "sliced", with tenants being given a different partial and abstracted topology view of the physical underlying network.

#### 5.1.2.3. Network Slicing

An IETF Network Slice is a logical network topology connecting a number of endpoints using a set of shared or dedicated network resources [I-D.ietf-teas-ietf-network-slices]. The resources are used to satisfy specific Service Level Objectives (SLOs) specified by the consumer.

IETF network slices are not, of themselves, TE constructs. However, a network operator that offers IETF network slices is likely to use many TE tools in order to manage their network and provide the services.

IETF Network Slices are defined such that they are independent of the underlying infrastructure connectivity and technologies used. From a customer's perspective an IETF Network Slice looks like a VPN connectivity matrix with additional information about the level of service required between endpoints. From an operator's perspective the IETF Network Slice looks like a set of routing or tunneling instructions with the network resource reservations necessary to provide the required service levels as specified by the SLOs. The concept of an IETF network slice is consistent with an enhanced VPN (VPN+) [I-D.ietf-teas-enhanced-vpn].

#### 5.1.3. IETF Techniques Used by TE Mechanisms

#### 5.1.3.1. Constraint-Based Routing

Constraint-based routing refers to a class of routing systems that compute routes through a network subject to the satisfaction of a set of constraints and requirements. In the most general case, constraint-based routing may also seek to optimize overall network performance while minimizing costs.

The constraints and requirements may be imposed by the network itself or by administrative policies. Constraints may include bandwidth, hop count, delay, and policy instruments such as resource class attributes. Constraints may also include domain specific attributes of certain network technologies and contexts which impose restrictions on the solution space of the routing function. Path oriented technologies such as MPLS have made constraint-based routing feasible and attractive in public IP networks.

The concept of constraint-based routing within the context of MPLS TE requirements in IP networks was first described in [RFC2702] and led to developments such as MPLS-TE [RFC3209] as described in Section 5.1.3.3.

Unlike QoS-based routing (for example, see [RFC2386], [MA], and [I-D.ietf-idr-performance-routing]) which generally addresses the issue of routing individual traffic flows to satisfy prescribed flow-based QoS requirements subject to network resource availability, constraint-based routing is applicable to traffic aggregates as well as flows and may be subject to a wide variety of constraints which may include policy restrictions.

##### 5.1.3.1.1. IGP Flexible Algorithms (Flex-Algos)

The traditional approach to routing in an IGP network relies on the IGP's deriving "shortest paths" over the network based solely on the IGP metric assigned to the links. Such an approach is often limited: traffic may tend to converge toward the destination, possibly causing congestion; and it is not possible to steer traffic onto paths depending on the end-to-end qualities demanded by the applications.

To overcome this limitation, various sorts of TE have been widely deployed (as described in this document), where the TE component is responsible for computing the path based on additional metrics and/or constraints. Such paths (or tunnels) need to be installed in the routers' forwarding tables in addition to, or as a replacement for the original paths computed by IGP's. The main drawback of these TE approaches is the additional complexity of protocols and management, and the state that may need to be maintained within the network.

IGP flexible algorithms (flex-algos) [I-D.ietf-lsr-flex-algo] allow IGPs to construct constraint-based paths over the network by computing constraint-based next hops. The intent of flex-algos is to reduce TE complexity by letting an IGP perform some basic TE computation capabilities. Flex-algo includes a set of extensions to the IGPs that enable a router to send TLVs that:

- \* describe a set of constraints on the topology
- \* identify calculation-type
- \* describe a metric-type that is to be used to compute the best paths through the constrained topology.

A given combination of calculation-type, metric-type, and constraints is known as a "Flexible Algorithm Definition" (or FAD). A router that sends such a set of TLVs also assigns a specific identifier (the Flexible Algorithm) to the specified combination of calculation-type, metric-type, and constraints.

There are two use cases for flex-algo: in IP networks [I-D.ietf-lsr-ip-flexalgo] and in segment routing networks [I-D.ietf-lsr-flex-algo]. In the first case, flex-algo computes paths to an IPv4 or IPv6 address, in the second case, flex-algo computes paths to a prefix SID (see Section 5.1.3.11).

There are many use cases where flex-algo can bring big value, such as:

- \* Expansion of functionality of IP Performance metrics [RFC5664] when points of interest could instantiate specific constraint-based routing (flex-algo) based on the measurement results.
- \* Nested usage of flex-algo and TE extensions for IGP (see Section 5.1.3.8) when we can form 'underlay' by means of flex-algo and 'overlay' by TE.
- \* Flex-algo in SR-MPLS (Section 5.1.3.11) is a base use case when we can easily benefit from TE-like topology that will be built without external TE component on routers or PCE (see Section 5.1.3.10).
- \* Building of network slices [I-D.ietf-teas-ietf-network-slices] where particular IETF network slice SLO can be guaranteed by flex-algo.

## 5.1.3.2.    RSVP

RSVP is a soft state signaling protocol [RFC2205]. It supports receiver initiated establishment of resource reservations for both multicast and unicast flows. RSVP was originally developed as a signaling protocol within the Integrated Services framework (see Section 5.1.1.1) for applications to communicate QoS requirements to the network and for the network to reserve relevant resources to satisfy the QoS requirements [RFC2205].

In RSVP, the traffic sender or source node sends a PATH message to the traffic receiver with the same source and destination addresses as the traffic which the sender will generate. The PATH message contains: (1) a sender traffic specification describing the characteristics of the traffic, (2) a sender template specifying the format of the traffic, and (3) an optional advertisement specification which is used to support the concept of One Pass With Advertising (OPWA) [RFC2205]. Every intermediate router along the path forwards the PATH message to the next hop determined by the routing protocol. Upon receiving a PATH message, the receiver responds with a RESV message which includes a flow descriptor used to request resource reservations. The RESV message travels to the sender or source node in the opposite direction along the path that the PATH message traversed. Every intermediate router along the path can reject or accept the reservation request of the RESV message. If the request is rejected, the rejecting router will send an error message to the receiver and the signaling process will terminate. If the request is accepted, link bandwidth and buffer space are allocated for the flow and the related flow state information is installed in the router.

One of the issues with the original RSVP specification was Scalability. This is because reservations were required for micro-flows, so that the amount of state maintained by network elements tends to increase linearly with the number of traffic flows. These issues are described in [RFC2961] which also modifies and extends RSVP to mitigate the scaling problems to make RSVP a versatile signaling protocol for the Internet. For example, RSVP has been extended to reserve resources for aggregation of flows, to set up MPLS explicit label switched paths (see Section 5.1.3.3), and to perform other signaling functions within the Internet. [RFC2961] also describes a mechanism to reduce the amount of Refresh messages required to maintain established RSVP sessions.

#### 5.1.3.3. Multiprotocol Label Switching (MPLS)

MPLS is an advanced forwarding scheme which also includes extensions to conventional IP control plane protocols. MPLS extends the Internet routing model and enhances packet forwarding and path control [RFC3031].

At the ingress to an MPLS domain, Label Switching Routers (LSRs) classify IP packets into Forwarding Equivalence Classes (FECs) based on a variety of factors, including, e.g., a combination of the information carried in the IP header of the packets and the local routing information maintained by the LSRs. An MPLS label stack entry is then prepended to each packet according to their forwarding equivalence classes. The MPLS label stack entry is 32 bits long and contains a 20-bit label field.

An LSR makes forwarding decisions by using the label prepended to packets as the index into a local next hop label forwarding entry (NHLFE). The packet is then processed as specified in the NHLFE. The incoming label may be replaced by an outgoing label (label swap), and the packet may be forwarded to the next LSR. Before a packet leaves an MPLS domain, its MPLS label may be removed (label pop). A Label Switched Path (LSP) is the path between an ingress LSRs and an egress LSRs through which a labeled packet traverses. The path of an explicit LSP is defined at the originating (ingress) node of the LSP. MPLS can use a signaling protocol such as RSVP or LDP to set up LSPs.

MPLS is a very powerful technology for Internet TE because it supports explicit LSPs which allow constraint-based routing to be implemented efficiently in IP networks [AWD2]. The requirements for TE over MPLS are described in [RFC2702]. Extensions to RSVP to support instantiation of explicit LSP are discussed in [RFC3209].

#### 5.1.3.4. Generalized MPLS (GMPLS)

GMPLS extends MPLS control protocols to encompass time-division (e.g., Synchronous Optical Network / Synchronous Digital Hierarchy (SONET/SDH), Plesiochronous Digital Hierarchy (PDH), Optical Transport Network (OTN)), wavelength ( $\lambda$ s), and spatial switching (e.g., incoming port or fiber to outgoing port or fiber) as well as continuing to support packet switching. GMPLS provides a common set of control protocols for all of these layers (including some technology-specific extensions) each of which has a diverse data or forwarding plane. GMPLS covers both the signaling and the routing part of that control plane and is based on the TE extensions to MPLS (see Section 5.1.3.3).

In GMPLS, the original MPLS architecture is extended to include LSRs whose forwarding planes rely on circuit switching, and therefore cannot forward data based on the information carried in either packet or cell headers. Specifically, such LSRs include devices where the switching is based on time slots, wavelengths, or physical ports. These additions impact basic LSP properties: how labels are requested and communicated, the unidirectional nature of MPLS LSPs, how errors are propagated, and information provided for synchronizing the ingress and egress LSRs.

#### 5.1.3.5. IP Performance Metrics

The IETF IP Performance Metrics (IPPM) working group has developed a set of standard metrics that can be used to monitor the quality, performance, and reliability of Internet services. These metrics can be applied by network operators, end-users, and independent testing groups to provide users and service providers with a common understanding of the performance and reliability of the Internet component 'clouds' they use/provide [RFC2330]. The criteria for performance metrics developed by the IPPM working group are described in [RFC2330]. Examples of performance metrics include one-way packet loss [RFC7680], one-way delay [RFC7679], and connectivity measures between two nodes [RFC2678]. Other metrics include second-order measures of packet loss and delay.

Some of the performance metrics specified by the IPPM working group are useful for specifying SLAs. SLAs are sets of service level objectives negotiated between users and service providers, wherein each objective is a combination of one or more performance metrics, possibly subject to certain constraints.

#### 5.1.3.6. Flow Measurement

The IETF Real Time Flow Measurement (RTFM) working group produced an architecture that defines a method to specify traffic flows as well as a number of components for flow measurement (meters, meter readers, manager) [RFC2722]. A flow measurement system enables network traffic flows to be measured and analyzed at the flow level for a variety of purposes. As noted in RFC 2722, a flow measurement system can be very useful in the following contexts:

- \* understanding the behavior of existing networks
- \* planning for network development and expansion
- \* quantification of network performance
- \* verifying the quality of network service

- \* attribution of network usage to users.

A flow measurement system consists of meters, meter readers, and managers. A meter observes packets passing through a measurement point, classifies them into groups, accumulates usage data (such as the number of packets and bytes for each group), and stores the usage data in a flow table. A group may represent any collection of user applications, hosts, networks, etc. A meter reader gathers usage data from various meters so it can be made available for analysis. A manager is responsible for configuring and controlling meters and meter readers. The instructions received by a meter from a manager include flow specifications, meter control parameters, and sampling techniques. The instructions received by a meter reader from a manager include the address of the meter whose data is to be collected, the frequency of data collection, and the types of flows to be collected.

#### 5.1.3.7. Endpoint Congestion Management

[RFC3124] provides a set of congestion control mechanisms for the use of transport protocols. It also allows the development of mechanisms for unifying congestion control across a subset of an endpoint's active unicast connections (called a congestion group). A congestion manager continuously monitors the state of the path for each congestion group under its control. The manager uses that information to instruct a scheduler on how to partition bandwidth among the connections of that congestion group.

#### 5.1.3.8. TE Extensions to the IGPs

[RFC5305] describes the extensions to the Intermediate System to Intermediate System (IS-IS) protocol to support TE, similarly [RFC3630] specifies TE extensions for OSPFv2 ([RFC5329] has the same description for OSPFv3).

The idea of redistribution of TE extensions such as link type and ID, local and remote IP addresses, TE metric, maximum bandwidth, maximum reservable bandwidth and unreserved bandwidth, admin group in IGP is a common for both IS-IS and OSPF. The information distributed by the IGPs in this way can be used to build a view of the state and capabilities of a TE network (see Section 5.1.3.12).

The difference is in the details of their transmission: IS-IS uses the Extended IS Reachability TLV (type 22) and Sub-TLVs for those TE parameters, OSPFv2 uses Opaque LSA [RFC5250] type 10 (OSPFv3 uses Intra-Area-TE-LSA) with two top-level TLV (Router Address and Link) also with Sub-TLVs for that purpose.



IS-IS also uses the Extended IP Reachability TLV (type 135, which have the new 32 bit metric) and the TE Router ID TLV (type 134). Those Sub-TLV details are described in [RFC8570] for IS-IS and in [RFC7471] for OSPFv2 ([RFC5329] for OSPFv3).

#### 5.1.3.9. Link-State BGP

In a number of environments, a component external to a network is called upon to perform computations based on the network topology and current state of the connections within the network, including TE information. This is information typically distributed by IGP routing protocols within the network (see Section 5.1.3.8).

The Border Gateway Protocol (BGP) (see also Section 7) is one of the essential routing protocols that glue the Internet together. BGP Link State (BGP-LS) [RFC7752] is a mechanism by which link-state and TE information can be collected from networks and shared with external components using the BGP routing protocol. The mechanism is applicable to physical and virtual IGP links, and is subject to policy control.

Information collected by BGP-LS can be used to construct the Traffic Engineering Database (TED, see Section 5.1.3.12) for use by the Path Computation Element (PCE, see Section 5.1.3.10), or may be used by Application-Layer Traffic Optimization (ALTO) servers (see Section 5.1.2.1).

#### 5.1.3.10. Path Computation Element

Constraint-based path computation is a fundamental building block for TE in MPLS and GMPLS networks. Path computation in large, multi-domain networks is complex and may require special computational components and cooperation between the elements in different domains. The Path Computation Element (PCE) [RFC4655] is an entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.

Thus, a PCE can provide a central component in a TE system operating on the TE Database (TED, see Section 5.1.3.12) with delegated responsibility for determining paths in MPLS, GMPLS, or Segment Routing networks. The PCE uses the Path Computation Element Communication Protocol (PCEP) [RFC5440] to communicate with Path Computation Clients (PCCs), such as MPLS LSRs, to answer their requests for computed paths or to instruct them to initiate new paths [RFC8281] and maintain state about paths already installed in the network [RFC8231].

PCEs form key components of a number of TE systems. More information about the applicability of PCE can be found in [RFC8051], while [RFC6805] describes the application of PCE to determining paths across multiple domains. PCE also has potential use in Abstraction and Control of TE Networks (ACTN) (see Section 5.1.2.2), Centralized Network Control [RFC8283], and Software Defined Networking (SDN) (see Section 4.3.2).

#### 5.1.3.11. Segment Routing

Segment Routing (SR) [RFC8402] leverages the source routing and tunneling paradigms. The path a packet takes is defined at the ingress and the packet is tunneled to the egress. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with an SR header: a label stack in MPLS case; a series of 128-bit segment identifiers in the IPv6 case.

Segments are identified by Segment Identifiers (SIDs). There are four types of SID that are relevant for TE.

Prefix SID: Unique within the routing domain used to identify a prefix.

Node SID: A Prefix SID with the 'N' bit set to identify a node.

Adjacency SID: Identifies a unidirectional adjacency.

Binding SID: A Binding SID has two purposes:

1. Used to advertise the mappings of prefixes to SIDs/Labels.
2. Used to advertise a path available for a Forwarding Equivalence Class.

A segment can represent any instruction, topological or service-based, thanks to the MPLS architecture [RFC3031]. Labels can be looked up in a global context (platform wide) as well as in some other context (see "context labels" in Section 3 of [RFC5331]).

The application of "policy" to segment routing can make SR into a TE mechanism as described in Section 5.1.1.3.

#### 5.1.3.12. Network TE State Definition and Presentation

The network states that are relevant to the TE need to be stored in the system and presented to the user. The Traffic Engineering Database (TED) is a collection of all TE information about all TE nodes and TE links in the network, which is an essential component of a TE system, such as MPLS-TE [RFC2702] and GMPLS [RFC3945]. In order to formally define the data in the TED and to present the data to the user with high usability, the data modeling language YANG [RFC7950] can be used as described in [RFC8795].

#### 5.1.3.13. System Management and Control Interfaces

The TE control system needs to have a management interface that is human-friendly and a control interfaces that is programmable for automation. The Network Configuration Protocol (NETCONF) [RFC6241] or the RESTCONF Protocol [RFC8040] provide programmable interfaces that are also human-friendly. These protocols use XML or JSON encoded messages. When message compactness or protocol bandwidth consumption needs to be optimized for the control interface, other protocols, such as Group Communication for the Constrained Application Protocol (CoAP) [RFC7390] or gRPC, are available, especially when the protocol messages are encoded in a binary format. Along with any of these protocols, the data modeling language YANG [RFC7950] can be used to formally and precisely define the interface data.

The Path Computation Element Communication Protocol (PCEP) [RFC5440] is another protocol that has evolved to be an option for the TE system control interface. The messages of PCEP are TLV-based, not defined by a data modeling language such as YANG.

### 5.2. Content Distribution

The Internet is dominated by client-server interactions, principally Web traffic although in the future, more sophisticated media servers may become dominant. The location and performance of major information servers has a significant impact on the traffic patterns within the Internet as well as on the perception of service quality by end users.

A number of dynamic load balancing techniques have been devised to improve the performance of replicated information servers. These techniques can cause spatial traffic characteristics to become more dynamic in the Internet because information servers can be dynamically picked based upon the location of the clients, the location of the servers, the relative utilization of the servers, the relative performance of different networks, and the relative

performance of different parts of a network. This process of assignment of distributed servers to clients is called traffic directing. It is an application layer function.

Traffic directing schemes that allocate servers in multiple geographically dispersed locations to clients may require empirical network performance statistics to make more effective decisions. In the future, network measurement systems may need to provide this type of information.

When congestion exists in the network, traffic directing and traffic engineering systems should act in a coordinated manner. This topic is for further study.

The issues related to location and replication of information servers, particularly web servers, are important for Internet traffic engineering because these servers contribute a substantial proportion of Internet traffic.

## 6. Recommendations for Internet Traffic Engineering

This section describes high-level recommendations for traffic engineering in the Internet in general terms.

The recommendations describe the capabilities needed to solve a TE problem or to achieve a TE objective. Broadly speaking, these recommendations can be categorized as either functional or non-functional recommendations.

- \* Functional recommendations describe the functions that a traffic engineering system should perform. These functions are needed to realize TE objectives by addressing traffic engineering problems.
- \* Non-functional recommendations relate to the quality attributes or state characteristics of a TE system. These recommendations may contain conflicting assertions and may sometimes be difficult to quantify precisely.

### 6.1. Generic Non-functional Recommendations

The generic non-functional recommendations for Internet traffic engineering are listed in the paragraphs that follow. In a given context, some of these recommendations may be critical while others may be optional. Therefore, prioritization may be required during the development phase of a TE system to tailor it to a specific operational context.

Usability: Usability is a human aspect of TE systems. It refers to

the ease with which a TE system can be deployed and operated. In general, it is desirable to have a TE system that can be readily deployed in an existing network. It is also desirable to have a TE system that is easy to operate and maintain.

**Automation:** Whenever feasible, a TE system should automate as many TE functions as possible to minimize the amount of human effort needed to analyze and control operational networks. Automation is particularly important in large-scale public networks because of the high cost of the human aspects of network operations and the high risk of network problems caused by human errors. Automation may entail the incorporation of automatic feedback and intelligence into some components of the TE system.

**Scalability:** Public networks continue to grow rapidly with respect to network size and traffic volume. Therefore, to remain applicable as the network evolves, a TE system should be scalable. In particular, a TE system should remain functional as the network expands with regard to the number of routers and links, and with respect to the traffic volume. A TE system should have a scalable architecture, should not adversely impair other functions and processes in a network element, and should not consume too many network resources when collecting and distributing state information, or when exerting control.

**Stability:** Stability is a very important consideration in TE systems that respond to changes in the state of the network. State-dependent TE methodologies typically include a trade-off between responsiveness and stability. It is strongly recommended that when a trade-off between responsiveness and stability is needed, it should be made in favor of stability (especially in public IP backbone networks).

**Flexibility:** A TE system should allow for changes in optimization policy. In particular, a TE system should provide sufficient configuration options so that a network administrator can tailor the system to a particular environment. It may also be desirable to have both online and offline TE subsystems which can be independently enabled and disabled. TE systems that are used in multi-class networks should also have options to support class based performance evaluation and optimization.

**Visibility:** Mechanisms should exist as part of the TE system to collect statistics from the network and to analyze these statistics to determine how well the network is functioning. Derived statistics such as traffic matrices, link utilization, latency, packet loss, and other performance measures of interest which are determined from network measurements can be used as

indicators of prevailing network conditions. The capabilities of the various components of the routing system are other examples of status information which should be observable.

**Simplicity:** A TE system should be as simple as possible. Simplicity in user interface does not necessarily imply that the TE system will use naive algorithms. When complex algorithms and internal structures are used, the user interface should hide such complexities from the network administrator as much as possible.

**Interoperability:** Whenever feasible, TE systems and their components should be developed with open standards-based interfaces to allow interoperation with other systems and components.

**Security:** Security is a critical consideration in TE systems. Such systems typically exert control over functional aspects of the network to achieve the desired performance objectives. Therefore, adequate measures must be taken to safeguard the integrity of the TE system. Adequate measures must also be taken to protect the network from vulnerabilities that originate from security breaches and other impairments within the TE system.

The remaining subsections of this section focus on some of the high-level functional recommendations for TE.

## 6.2. Routing Recommendations

Routing control is a significant aspect of Internet traffic engineering. Routing impacts many of the key performance measures associated with networks, such as throughput, delay, and utilization. Generally, it is very difficult to provide good service quality in a wide area network without effective routing control. A desirable TE routing system is one that takes traffic characteristics and network constraints into account during route selection while maintaining stability.

Shortest path first (SPF) IGP algorithms are based on shortest path algorithms and have limited control capabilities for TE [RFC2702], [AWD2]. These limitations include:

1. Pure SPF protocols do not take network constraints and traffic characteristics into account during route selection. For example, IGPs always select the shortest paths based on link metrics assigned by administrators so load sharing cannot be performed across paths of different costs. Using shortest paths to forward traffic may cause the following problems:

- \* If traffic from a source to a destination exceeds the capacity of a link along the shortest path, the link (and hence the shortest path) becomes congested while a longer path between these two nodes may be under-utilized
  - \* The shortest paths from different sources can overlap at some links. If the total traffic from the sources exceeds the capacity of any of these links, congestion will occur.
  - \* Problems can also occur because traffic demand changes over time, but network topology and routing configuration cannot be changed as rapidly. This causes the network topology and routing configuration to become sub-optimal over time, which may result in persistent congestion problems.
2. The Equal-Cost Multi-Path (ECMP) capability of SPF IGPs supports sharing of traffic among equal cost paths between two nodes. However, ECMP attempts to divide the traffic as equally as possible among the equal cost shortest paths. Generally, ECMP does not support configurable load sharing ratios among equal cost paths. The result is that one of the paths may carry significantly more traffic than other paths because it may also carry traffic from other sources. This situation can result in congestion along the path that carries more traffic. Weighted ECMP (WECMP) (see, for example, [I-D.ietf-bess-evpn-unequal-lb]) provides some mitigation.
  3. Modifying IGP metrics to control traffic routing tends to have network-wide effects. Consequently, undesirable and unanticipated traffic shifts can be triggered as a result. Work described in Section 8 may be capable of better control [FT00], [FT01].

Because of these limitations, new capabilities are needed to enhance the routing function in IP networks. Some of these capabilities are summarized below.

- \* Constraint-based routing computes routes to fulfill requirements subject to constraints. This can be useful in public IP backbones with complex topologies. Constraints may include bandwidth, hop count, delay, and administrative policy instruments such as resource class attributes [RFC2702], [RFC2386]. This makes it possible to select routes that satisfy a given set of requirements. Routes computed by constraint-based routing are not necessarily the shortest paths. Constraint-based routing works best with path-oriented technologies that support explicit routing, such as MPLS.

- \* Constraint-based routing can also be used as a way to distribute traffic onto the infrastructure, including for best effort traffic. For example, congestion problems caused by uneven traffic distribution may be avoided or reduced by knowing the reservable bandwidth attributes of the network links and by specifying the bandwidth requirements for path selection.
- \* A number of enhancements to the link state IGPs are needed to allow them to distribute additional state information required for constraint-based routing. The extensions to OSPF are described in [RFC3630], and to IS-IS in [RFC5305]. Some of the additional topology state information includes link attributes such as reservable bandwidth and link resource class attribute (an administratively specified property of the link). The resource class attribute concept is defined in [RFC2702]. The additional topology state information is carried in new TLVs and sub-TLVs in IS-IS, or in the Opaque LSA in OSPF [RFC5305], [RFC3630].
- \* An enhanced link-state IGP may flood information more frequently than a normal IGP. This is because even without changes in topology, changes in reservable bandwidth or link affinity can trigger the enhanced IGP to initiate flooding. A trade-off between the timeliness of the information flooded and the flooding frequency is typically implemented using a threshold based on the percentage change of the advertised resources to avoid excessive consumption of link bandwidth and computational resources, and to avoid instability in the TED.
- \* In a TE system, it is also desirable for the routing subsystem to make the load splitting ratio among multiple paths (with equal cost or different cost) configurable. This capability gives network administrators more flexibility in the control of traffic distribution across the network. It can be very useful for avoiding/relieving congestion in certain situations. Examples can be found in [XIAO] and [I-D.ietf-bess-evpn-unequal-lb].
- \* The routing system should also have the capability to control the routes of subsets of traffic without affecting the routes of other traffic if sufficient resources exist for this purpose. This capability allows a more refined control over the distribution of traffic across the network. For example, the ability to move traffic away from its original path to another path (without affecting other traffic paths) allows the traffic to be moved from resource-poor network segments to resource-rich segments. Path oriented technologies such as MPLS-TE inherently support this capability as discussed in [AWD2].



- \* Additionally, the routing subsystem should be able to select different paths for different classes of traffic (or for different traffic behavior aggregates) if the network supports multiple classes of service (different behavior aggregates).

### 6.3. Traffic Mapping Recommendations

Traffic mapping is the assignment of traffic workload onto (pre-established) paths to meet certain requirements. Thus, while constraint-based routing deals with path selection, traffic mapping deals with the assignment of traffic to established paths which may have been generated by constraint-based routing or by some other means. Traffic mapping can be performed by time-dependent or state-dependent mechanisms, as described in Section 4.1.

An important aspect of the traffic mapping function is the ability to establish multiple paths between an originating node and a destination node, and the capability to distribute the traffic between the two nodes across the paths according to some policies. A pre-condition for this scheme is the existence of flexible mechanisms to partition traffic and then assign the traffic partitions onto the parallel paths as noted in [RFC2702]. When traffic is assigned to multiple parallel paths, it is recommended that special care should be taken to ensure proper ordering of packets belonging to the same application (or traffic flow) at the destination node of the parallel paths.

Mechanisms that perform the traffic mapping functions should aim to map the traffic onto the network infrastructure to minimize congestion. If the total traffic load cannot be accommodated, or if the routing and mapping functions cannot react fast enough to changing traffic conditions, then a traffic mapping system may use short time scale congestion control mechanisms (such as queue management, scheduling, etc.) to mitigate congestion. Thus, mechanisms that perform the traffic mapping functions complement existing congestion control mechanisms. In an operational network, traffic should be mapped onto the infrastructure such that intra-class and inter-class resource contention are minimized (see Section 2).

When traffic mapping techniques that depend on dynamic state feedback (e.g., MATE [MATE] and such like) are used, special care must be taken to guarantee network stability.

#### 6.4. Measurement Recommendations

The importance of measurement in TE has been discussed throughout this document. A TE system should include mechanisms to measure and collect statistics from the network to support the TE function. Additional capabilities may be needed to help in the analysis of the statistics. The actions of these mechanisms should not adversely affect the accuracy and integrity of the statistics collected. The mechanisms for statistical data acquisition should also be able to scale as the network evolves.

Traffic statistics may be classified according to long-term or short-term timescales. Long-term traffic statistics are very useful for traffic engineering. Long-term traffic statistics may periodically record network workload (such as hourly, daily, and weekly variations in traffic profiles) as well as traffic trends. Aspects of the traffic statistics may also describe class of service characteristics for a network supporting multiple classes of service. Analysis of the long-term traffic statistics may yield other information such as busy hour characteristics, traffic growth patterns, persistent congestion problems, hot-spot, and imbalances in link utilization caused by routing anomalies.

A mechanism for constructing traffic matrices for both long-term and short-term traffic statistics should be in place. In multi-service IP networks, the traffic matrices may be constructed for different service classes. Each element of a traffic matrix represents a statistic about the traffic flow between a pair of abstract nodes. An abstract node may represent a router, a collection of routers, or a site in a VPN.

Traffic statistics should provide reasonable and reliable indicators of the current state of the network on the short-term scale. Some short term traffic statistics may reflect link utilization and link congestion status. Examples of congestion indicators include excessive packet delay, packet loss, and high resource utilization. Examples of mechanisms for distributing this kind of information include SNMP, probing tools, FTP, IGP link state advertisements, and NETCONF/RESTCONF, etc.

#### 6.5. Policing, Planning, and Access Control

The recommendations in Section 6.2 and Section 6.3 may be sub-optimal or even ineffective if the amount of traffic flowing on a route or path exceeds the capacity of the resource on that route or path. Several approaches can be used to increase the performance of TE systems.

- \* The fundamental approach is some form of planning where traffic is steered onto paths so that it is distributed across the available resources. This planning may be centralized or distributed, and must be aware of the planned traffic volumes and available resources. However, this approach is only of value if the traffic is presented conformant to the planned traffic volumes.
- \* Traffic flows may be policed at the edges of a network. This is a simple way to check that the actual traffic volumes are consistent with the planned volumes. Some form of measurement (see Section 6.4) is used to determine the rate of arrival of traffic and excess traffic could be discarded. Alternatively, excess traffic could be forwarded as best-effort within the network. However, this approach is only completely effective if the planning is stringent and network-wide, and if a harsh approach is taken to disposing of excess traffic.
- \* Resource-based admission control is the process whereby network nodes decide whether to grant access to resources. The basis for the decision on a packet-by-packet basis is determination of the flow to which the packet belongs. This information is combined with policy instructions that have been locally configured, or installed through the management or control planes. The end result is that a packet may be allowed to access (or use) specific resources on the node if and only if the policy is conformed with for the flow to which the packet belongs.

Combining some element of all three of these measures is advisable to achieve a better TE system.

#### 6.6. Network Survivability

Network survivability refers to the capability of a network to maintain service continuity in the presence of faults. This can be accomplished by promptly recovering from network impairments and maintaining the required QoS for existing services after recovery. Survivability is an issue of great concern within the Internet community due to the demand to carry mission critical traffic, real-time traffic, and other high priority traffic over the Internet. Survivability can be addressed at the device level by developing network elements that are more reliable; and at the network level by incorporating redundancy into the architecture, design, and operation of networks. It is recommended that a philosophy of robustness and survivability should be adopted in the architecture, design, and operation of TE that control IP networks (especially public IP networks). Because different contexts may demand different levels of survivability, the mechanisms developed to support network survivability should be flexible so that they can be tailored to

different needs. A number of tools and techniques have been developed to enable network survivability including MPLS Fast Reroute [RFC4090], Topology Independent Loop-free Alternate Fast Re-route for Segment Routing [I-D.ietf-rtgwg-segment-routing-ti-lfa] RSVP-TE Extensions in Support of End-to-End GMPLS Recovery [RFC4872], and GMPLS Segment Recovery [RFC4873].

The impact of service outages varies significantly for different service classes depending on the duration of the outage which can vary from milliseconds (with minor service impact) to seconds (with possible call drops for IP telephony and session time-outs for connection oriented transactions) to minutes and hours (with potentially considerable social and business impact). Different duration outages have different impacts depending on the throughput of the traffic flows that are interrupted.

Failure protection and restoration capabilities are available in multiple layers as network technologies have continued to evolve. Optical networks are capable of providing dynamic ring and mesh restoration functionality at the wavelength level. At the SONET/SDH layer survivability capability is provided with Automatic Protection Switching (APS) as well as self-healing ring and mesh architectures. Similar functionality is provided by layer 2 technologies such as Ethernet.

Rerouting is used at the IP layer to restore service following link and node outages. Rerouting at the IP layer occurs after a period of routing convergence which may require seconds to minutes to complete. Path-oriented technologies such as MPLS ([RFC3469]) can be used to enhance the survivability of IP networks in a potentially cost effective manner.

An important aspect of multi-layer survivability is that technologies at different layers may provide protection and restoration capabilities at different granularities in terms of time scales and at different bandwidth granularity (from packet-level to wavelength level). Protection and restoration capabilities can also be sensitive to different service classes and different network utility models. Coordinating different protection and restoration capabilities across multiple layers in a cohesive manner to ensure network survivability is maintained at reasonable cost is a challenging task. Protection and restoration coordination across layers may not always be feasible, because networks at different layers may belong to different administrative domains.

The following paragraphs present some of the general recommendations for protection and restoration coordination.

- \* Protection and restoration capabilities from different layers should be coordinated to provide network survivability in a flexible and cost effective manner. Avoiding duplication of functions in different layers is one way to achieve the coordination. Escalation of alarms and other fault indicators from lower to higher layers may also be performed in a coordinated manner. The order of timing of restoration triggers from different layers is another way to coordinate multi-layer protection/restoration.
- \* Network capacity reserved in one layer to provide protection and restoration is not available to carry traffic in a higher layer: it is not visible as spare capacity in the higher layer. Placing protection/restoration functions in many layers may increase redundancy and robustness, but it can result in significant inefficiencies in network resource utilization. Careful planning is needed to balance the trade-off between the desire for survivability and the optimal use of resources.
- \* It is generally desirable to have protection and restoration schemes that are intrinsically bandwidth efficient.
- \* Failure notifications throughout the network should be timely and reliable if they are to be acted on as triggers for effective protection and restoration actions.
- \* Alarms and other fault monitoring and reporting capabilities should be provided at the right network layers so that the protection and restoration actions can be taken in those layers.

#### 6.6.1. Survivability in MPLS Based Networks

Because MPLS is path-oriented, it has the potential to provide faster and more predictable protection and restoration capabilities than conventional hop by hop routed IP systems. Protection types for MPLS networks can be divided into four categories.

- \* **Link Protection:** The objective of link protection is to protect an LSP from the failure of a given link. Under link protection, a protection or backup LSP (the secondary LSP) follows a path that is disjoint from the path of the working or operational LSP (the primary LSP) at the particular link where link protection is required. When the protected link fails, traffic on the working LSP is switched to the protection LSP at the head-end of the failed link. As a local repair method, link protection can be fast. This form of protection may be most appropriate in situations where some network elements along a given path are known to be less reliable than others.

- \* **Node Protection:** The objective of node protection is to protect an LSP from the failure of a given node. Under node protection, the secondary LSP follows a path that is disjoint from the path of the primary LSP at the particular node where node protection is required. The secondary LSP is also disjoint from the primary LSP at all links attached to the node to be protected. When the protected node fails, traffic on the working LSP is switched over to the protection LSP at the upstream LSR directly connected to the failed node. Node protection covers a slightly larger part of the network compared to link protection, but is otherwise fundamentally the same.
- \* **Path Protection:** The goal of LSP path protection (or end-to-end protection) is to protect an LSP from any failure along its routed path. Under path protection, the path of the protection LSP is completely disjoint from the path of the working LSP. The advantage of path protection is that the backup LSP protects the working LSP from all possible link and node failures along the path, except for failures of ingress or egress LSR. Additionally, path protection may be more efficient in terms of resource usage than link or node protection applied at every hop along the path. However, path protection may be slower than link and node protection because the fault notifications have to be propagated further.
- \* **Segment Protection:** An MPLS domain may be partitioned into multiple subdomains (protection domains). Path protection is applied to the path of each LSP as it crosses the domain from its ingress to the domain to where it egresses the domain. In cases where an LSP traverses multiple protection domains, a protection mechanism within a domain only needs to protect the segment of the LSP that lies within the domain. Segment protection will generally be faster than end-to-end path protection because recovery generally occurs closer to the fault and the notification doesn't have to propagate as far.

See [RFC3469] and [RFC6372] for a more comprehensive discussion of MPLS based recovery.

#### 6.6.2. Protection Options

Another issue to consider is the concept of protection options. We use notation such as "m:n protection", where m is the number of protection LSPs used to protect n working LSPs. In all cases except 1+1 protection, the resources associated with the protection LSPs can be used to carry preemptable best-effort traffic when the working LSP is functioning correctly.

- \* 1:1 protection: One working LSP is protected/restored by one protection LSP.
- \* 1:n protection: One protection LSP is used to protect/restore n working LSPs. Only one failed LSP can be restored at any time.
- \* n:1 protection: One working LSP is protected/restored by n protection LSPs, possibly with load splitting across the protection LSPs. This may be especially useful when it is not feasible to find one path for the backup that can satisfy the bandwidth requirement of the primary LSP.
- \* 1+1 protection: Traffic is sent concurrently on both the working LSP and a protection LSP. The egress LSR selects one of the two LSPs based on local policy (usually based on traffic integrity). When a fault disrupts the traffic on one LSP, the egress switches to receive traffic from the other LSP. This approach is expensive in how it consumes network but recovers from failures most rapidly.

#### 6.7. Multi-Layer Traffic Engineering

Networks are often arranged as layers. A layer relationship may represent the interaction between technologies (for example, an IP network operated over an optical network), or the relationship between different network operators (for example, a customer network operated over a service provider's network). Note that a multi-layer network does not imply the use of multiple technologies, although some form of encapsulation is often applied.

Multi-layer traffic engineering presents a number of challenges associated with scalability and confidentiality. These issues are addressed in [RFC7926] which discusses the sharing of information between domains through policy filters, aggregation, abstraction, and virtualization. That document also discusses how existing protocols can support this scenario with special reference to BGP-LS (see Section 5.1.3.9).

PCE (see Section 5.1.3.10) is also a useful tool for multi-layer networks as described in [RFC6805], [RFC8685], and [RFC5623]. Signaling techniques for multi-layer TE are described in [RFC6107].

See also Appendix A.3.1 for a discussion of how the overlay model has been important in the development of TE, and Section 6.6 for examination of multi-layer network survivability.

## 6.8. Traffic Engineering in Diffserv Environments

Increasing requirements to support multiple classes of traffic in the Internet, such as best effort and mission critical data, calls for IP networks to differentiate traffic according to some criteria and to give preferential treatment to certain types of traffic. Large numbers of flows can be aggregated into a few behavior aggregates based on some criteria based on common performance requirements in terms of packet loss ratio, delay, and jitter, or in terms of common fields within the IP packet headers.

Differentiated Services (Diffserv) [RFC2475] can be used to ensure that SLAs defined to differentiate between traffic flows are met. Classes of service (CoS) can be supported in a Diffserv environment by concatenating per-hop behaviors (PHBs) along the routing path. A PHB is the forwarding behavior that a packet receives at a Diffserv-compliant node, and it can be configured at each router. PHBs are delivered using buffer management and packet scheduling mechanisms and require that the ingress nodes use traffic classification, marking, policing, and shaping.

TE can complement Diffserv to improve utilization of network resources. TE can be operated on an aggregated basis across all service classes [RFC3270], or on a per service class basis. The former is used to provide better distribution of the traffic load over the network resources (see [RFC3270] for detailed mechanisms to support aggregate TE). The latter case is discussed below since it is specific to the Diffserv environment, with so called Diffserv-aware traffic engineering [RFC4124].

For some Diffserv networks, it may be desirable to control the performance of some service classes by enforcing relationships between the traffic workload contributed by each service class and the amount of network resources allocated or provisioned for that service class. Such relationships between demand and resource allocation can be enforced using a combination of, for example:

- \* TE mechanisms on a per service class basis that enforce the relationship between the amount of traffic contributed by a given service class and the resources allocated to that class.
- \* Mechanisms that dynamically adjust the resources allocated to a given service class to relate to the amount of traffic contributed by that service class.

It may also be desirable to limit the performance impact of high priority traffic on relatively low priority traffic. This can be achieved, for example, by controlling the percentage of high priority



traffic that is routed through a given link. Another way to accomplish this is to increase link capacities appropriately so that lower priority traffic can still enjoy adequate service quality. When the ratio of traffic workload contributed by different service classes varies significantly from router to router, it may not be enough to rely on conventional IGP routing protocols or on TE mechanisms that are not sensitive to different service classes. Instead, it may be desirable to perform TE, especially routing control and mapping functions, on a per service class basis. One way to accomplish this in a domain that supports both MPLS and Diffserv is to define class specific LSPs and to map traffic from each class onto one or more LSPs that correspond to that service class. An LSP corresponding to a given service class can then be routed and protected/restored in a class dependent manner, according to specific policies.

Performing TE on a per class basis may require per-class parameters to be distributed. It is common to have some classes share some aggregate constraints (e.g., maximum bandwidth requirement) without enforcing the constraint on each individual class. These classes can be grouped into class-types, and per-class-type parameters can be distributed to improve scalability. This also allows better bandwidth sharing between classes in the same class-type. A class-type is a set of classes that satisfy the following two conditions:

- \* Classes in the same class-type have common aggregate requirements to satisfy required performance levels.
- \* There is no requirement to be enforced at the level of an individual class in the class-type. Note that it is, nevertheless, still possible to implement some priority policies for classes in the same class-type to permit preferential access to the class-type bandwidth through the use of preemption priorities.

See [RFC4124] for detailed requirements on Diffserv-aware TE.

## 6.9. Network Controllability

Offline and online (see Section 4.2) TE considerations are of limited utility if the network cannot be controlled effectively to implement the results of TE decisions and to achieve the desired network performance objectives.

Capacity augmentation is a coarse-grained solution to TE issues. However, it is simple and may be advantageous if bandwidth is abundant and cheap. However, bandwidth is not always abundant and cheap, and additional capacity might not always be the best solution.

Adjustments of administrative weights and other parameters associated with routing protocols provide finer-grained control, but this approach is difficult to use and imprecise because of the the way the routing protocols interact occur across the network.

Control mechanisms can be manual (e.g., static configuration), partially-automated (e.g., scripts), or fully-automated (e.g., policy based management systems). Automated mechanisms are particularly useful in large scale networks. Multi-vendor interoperability can be facilitated by standardized management systems (e.g., YANG models) to support the control functions required to address TE objectives.

Network control functions should be secure, reliable, and stable as these are often needed to operate correctly in times of network impairments (e.g., during network congestion or security attacks).

## 7. Inter-Domain Considerations

Inter-domain TE is concerned with performance optimization for traffic that originates in one administrative domain and terminates in a different one.

BGP [RFC4271] is the standard exterior gateway protocol used to exchange routing information between autonomous systems (ASes) in the Internet. BGP includes a sequential decision process that calculates the preference for routes to a given destination network. There are two fundamental aspects to inter-domain TE using BGP:

- \* Route Redistribution: Controlling the import and export of routes between ASes, and controlling the redistribution of routes between BGP and other protocols within an AS.
- \* Best path selection: Selecting the best path when there are multiple candidate paths to a given destination network. This is performed by the BGP decision process, selecting preferred exit points out of an AS towards specific destination networks taking a number of different considerations into account. The BGP path selection process can be influenced by manipulating the attributes associated with the process, including NEXT-HOP, WEIGHT, LOCAL-PREFERENCE, AS-PATH, ROUTE-ORIGIN, MULTI-EXIT-DESCRIMINATOR (MED), IGP METRIC, etc.

Route-maps provide the flexibility to implement complex BGP policies based on pre-configured logical conditions. They can be used to control import and export policies for incoming and outgoing routes, control the redistribution of routes between BGP and other protocols, and influence the selection of best paths by manipulating the attributes associated with the BGP decision process. Very complex

logical expressions that implement various types of policies can be implemented using a combination of Route-maps, BGP-attributes, Access-lists, and Community attributes.

When considering inter-domain TE with BGP, note that the outbound traffic exit point is controllable, whereas the interconnection point where inbound traffic is received typically is not. Therefore, it is up to each individual network to implement TE strategies that deal with the efficient delivery of outbound traffic from its customers to its peering points. The vast majority of TE policy is based on a "closest exit" strategy, which offloads inter-domain traffic at the nearest outbound peering point towards the destination AS. Most methods of manipulating the point at which inbound traffic enters a are either ineffective, or not accepted in the peering community.

Inter-domain TE with BGP is generally effective, but it is usually applied in a trial-and-error fashion because a TE system usually only has a view of the available network resources within one domain (an AS in this case). A systematic approach for inter-domain TE requires cooperation between the domains. Further, what may be considered a good solution in one domain may not necessarily be a good solution in another. Moreover, it is generally considered inadvisable for one domain to permit a control process from another domain to influence the routing and management of traffic in its network.

MPLS TE-tunnels (LSPs) can add a degree of flexibility in the selection of exit points for inter-domain routing by applying the concept of relative and absolute metrics. If BGP attributes are defined such that the BGP decision process depends on IGP metrics to select exit points for inter-domain traffic, then some inter-domain traffic destined to a given peer network can be made to prefer a specific exit point by establishing a TE-tunnel between the router making the selection and the peering point via a TE-tunnel and assigning the TE-tunnel a metric which is smaller than the IGP cost to all other peering points. RSVP-TE protocol extensions for inter-domain MPLS and GMPLS are described in [RFC5151].

Similarly to intra-domain TE, inter-domain TE is best accomplished when a traffic matrix can be derived to depict the volume of traffic from one AS to another.

## 8. Overview of Contemporary TE Practices in Operational IP Networks

This section provides an overview of some TE practices in IP networks. The focus is on aspects of control of the routing function in operational contexts. The intent here is to provide an overview of the commonly used practices: the discussion is not intended to be exhaustive.

Service providers apply many of the TE mechanisms described in this document to optimize the performance of their IP networks. These techniques include capacity planning for long timescales; routing control using IGP metrics and MPLS, as well as path planning and path control using MPLS and Segment Routing for medium timescales; and traffic management mechanisms for short timescale.

Capacity planning is an important component of how a service provider plans an effective IP network. These plans may take the following aspects into account: location of and new links or nodes, existing and predicted traffic patterns, costs, link capacity, topology, routing design, and survivability.

Performance optimization of operational networks is usually an ongoing process in which traffic statistics, performance parameters, and fault indicators are continually collected from the network. This empirical data is analyzed and used to trigger TE mechanisms. Tools that perform what-if analysis can also be used to assist the TE process by reviewing scenarios before a new set of configurations are implemented in the operational network.

Real-time intra-domain TE using the IGP is done by increasing the OSPF or IS-IS metric of a congested link until enough traffic has been diverted away from that link. This approach has some limitations as discussed in Section 6.2. Intra-domain TE approaches ([RR94] [FT00] [FT01] [WANG]) take traffic matrix, network topology, and network performance objectives as input, and produce link metrics and load-sharing ratios. These processes open the possibility for intra-domain TE with IGP to be done in a more systematic way.

Administrators of MPLS-TE networks specify and configure link attributes and resource constraints such as maximum reservable bandwidth and resource class attributes for the links in the domain. A link state IGP that supports TE extensions (IS-IS-TE or OSPF-TE) is used to propagate information about network topology and link attributes to all routers in the domain. Network administrators specify the LSPs that are to originate at each router. For each LSP, the network administrator specifies the destination node and the attributes of the LSP which indicate the requirements that are to be satisfied during the path selection process. The attributes may include an explicit path for the LSP to follow, or originating router uses a local constraint-based routing process to compute the path of the LSP. RSVP-TE is used as a signaling protocol to instantiate the LSPs. By assigning proper bandwidth values to links and LSPs, congestion caused by uneven traffic distribution can be avoided or mitigated.

The bandwidth attributes of an LSP relates to the bandwidth requirements of traffic that flows through the LSP. The traffic attribute of an LSP can be modified to accommodate persistent shifts in demand (traffic growth or reduction). If network congestion occurs due to some unexpected events, existing LSPs can be rerouted to alleviate the situation or network administrator can configure new LSPs to divert some traffic to alternative paths. The reservable bandwidth of the congested links can also be reduced to force some LSPs to be rerouted to other paths. A traffic matrix in an MPLS domain can also be estimated by monitoring the traffic on LSPs. Such traffic statistics can be used for a variety of purposes including network planning and network optimization.

Network management and planning systems have evolved and taken over a lot of the responsibility for determining traffic paths in TE networks. This allows a network-wide view of resources, and facilitates coordination of the use of resources for all traffic flows in the network. Initial solutions using a PCE to perform path computation on behalf of network routers have given way to an approach that follows the SDN architecture. A stateful PCE is able to track all of the LSPs in the network and can redistribute them to make better use of the available resources. Such a PCE can form part of a network orchestrator that uses PCEP or some other southbound interface to instruct the signaling protocol or directly program the routers.

Segment routing leverages a centralized TE controller and either an MPLS or IPv6 forwarding plane, but does not need to use a signaling protocol or management plane protocol to reserve resources in the routers. All resource reservation is logical within the controller, and not distributed to the routers. Packets are steered through the network using segment routing, and this may have configuration and operational scaling benefits.

As mentioned in Section 7, there is usually no direct control over the distribution of inbound traffic to a domain. Therefore, the main goal of inter-domain TE is to optimize the distribution of outbound traffic between multiple inter-domain links. When operating a global network, maintaining the ability to operate the network in a regional fashion where desired, while continuing to take advantage of the benefits of a global network, also becomes an important objective.

Inter-domain TE with BGP begins with the placement of multiple peering interconnection points that are in close proximity to traffic sources/destination, and offer lowest cost paths across the network between the peering points and the sources/destinations. Some location-decision problems that arise in association with inter-domain routing are discussed in [AWD5].

Once the locations of the peering interconnects have been determined and implemented, the network operator decides how best to handle the routes advertised by the peer, as well as how to propagate the peer's routes within their network. One way to engineer outbound traffic flows in a network with many peering interconnects is to create a hierarchy of peers. Generally, the shortest AS paths will be chosen to forward traffic but BGP metrics can be used to prefer some peers and so favor particular paths. Preferred peers are those peers attached through peering interconnects with the most available capacity. Changes may be needed, for example, to deal with a "problem peer" who is difficult to work with on upgrades or is charging high prices for connectivity to their network. In that case, the peer may be given a reduced preference. This type of change can affect a large amount of traffic, and is only used after other methods have failed to provide the desired results.

When there are multiple exit points toward a given peer, and only one of them is congested, it is not necessary to shift traffic away from the peer entirely, but only from the one congested connections. This can be achieved by using passive IGP-metrics, AS-path filtering, or prefix filtering.

## 9. Security Considerations

This document does not introduce new security issues.

Network security is, of course, an important issue. In general, TE mechanisms are security neutral: they may use tunnels which can slightly help protect traffic from inspection and which, in some cases, can be secured using encryption; they put traffic onto predictable paths within the network that may make it easier to find and attack; they increase the complexity of operation and management of the network; and they enable traffic to be steered onto more secure links or to more secure parts of the network.

The consequences of attacks on the control and management protocols used to operate TE networks can be significant: traffic can be hijacked to pass through specific nodes that perform inspection, or even to be delivered to the wrong place; traffic can be steered onto paths that deliver quality that is below the desired quality; and, networks can be congested or have resources on key links consumed. Thus, it is important to use adequate protection mechanisms on all protocols used to deliver TE.

Certain aspects of a network may be deduced from the details of the TE paths that are used. For example, the link connectivity of the network, and the quality and load on individual links may be assumed from knowing the paths of traffic and the requirements they place on

the network (for example, by seeing the control messages or through path- trace techniques). Such knowledge can be used to launch targeted attacks (for example, taking down critical links) or can reveal commercially sensitive information (for example, whether a network is close to capacity). Network operators may, therefore, choose techniques that mask or hide information from within the network.

#### 10. IANA Considerations

This draft makes no requests for IANA action.

#### 11. Acknowledgments

Much of the text in this document is derived from RFC 3272. The authors of this document would like to express their gratitude to all involved in that work. Although the source text has been edited in the production of this document, the original authors should be considered as Contributors to this work. They were:

Daniel O. Awduche  
Movaz Networks

Angela Chiu  
Celion Networks

Anwar Elwalid  
Lucent Technologies

Indra Widjaja  
Bell Labs, Lucent Technologies

XiPeng Xiao  
Redback Networks

The acknowledgements in RFC3272 were as below. All people who helped in the production of that document also need to be thanked for the carry-over into this new document.

The authors would like to thank Jim Boyle for inputs on the recommendations section, Francois Le Faucheur for inputs on Diffserv aspects, Blaine Christian for inputs on measurement, Gerald Ash for inputs on routing in telephone networks and for text on event-dependent TE methods, Steven Wright for inputs on network controllability, and Jonathan Aufderheide for inputs on inter-domain TE with BGP. Special thanks to Randy Bush for proposing the TE taxonomy based on "tactical versus strategic" methods. The subsection describing an "Overview of ITU Activities Related to Traffic Engineering" was adapted from a contribution by Waisum Lai. Useful feedback and pointers to relevant materials were provided by J. Noel Chiappa. Additional comments were provided by Glenn Grotefeld during the working last call process. Finally, the authors would like to thank Ed Kern, the TEWG co-chair, for his comments and support.

The early versions of this document were produced by the TEAS Working Group's RFC3272bis Design Team. The full list of members of this team is:

Acee Lindem  
Adrian Farrel  
Aijun Wang  
Daniele Ceccarelli  
Dieter Beller  
Jeff Tantsura  
Julien Meuric  
Liu Hua  
Loa Andersson  
Luis Miguel Contreras  
Martin Horneffer  
Tarek Saad  
Xufeng Liu

The production of this document includes a fix to the original text resulting from an Errata Report by Jean-Michel Grimaldi.

The author of this document would also like to thank Dhurv Dhody, Gyan Mishra, and Dave Taht for review comments.

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).



## 12. Contributors

The following people contributed substantive text to this document:

Gert Grammel  
EMail: ggrammel@juniper.net

Loa Andersson  
EMail: loa@pi.nu

Xufeng Liu  
EMail: xufeng.liu.ietf@gmail.com

Lou Berger  
EMail: lberger@labn.net

Jeff Tantsura  
EMail: jefftant.ietf@gmail.com

Daniel King  
EMail: daniel@olddog.co.uk

Boris Hassanov  
EMail: bhassanov@yandex-team.ru

Kiran Makhiyani  
Email: kiranm@futurewei.com

Dhruv Dhody  
Email: dhruv.ietf@gmail.com

Mohamed Boucadair  
Email: mohamed.boucadair@orange.com

## 13. Informative References

- [AJ19]     Adekitan, A., Abolade, J., and O. Shobayo, "Data mining approach for predicting the daily Internet data traffic of a smart university", Article Journal of Big Data, 2019, Volume 6, Number 1, Page 1, 1998, <<https://journalofbigdata.springeropen.com/track/pdf/10.1186/s40537-019-0176-5.pdf>>.
- [ASH2]     Ash, J., "Dynamic Routing in Telecommunications Networks", Book McGraw Hill, 1998, <<https://dl.acm.org/doi/book/10.5555/541099>>.

- [ATSSS]    "Study on access traffic steering, switch and splitting support in the 5G System (5GS) architecture", Specification 3GPP Technical Report 23.793 Release 16, December 2018, <[https://www.3gpp.org/ftp//Specs/archive/23\\_series/23.793/23793-g00.zip](https://www.3gpp.org/ftp//Specs/archive/23_series/23.793/23793-g00.zip)>.
- [AWD2]    Awduche, D., "MPLS and Traffic Engineering in IP Networks", Article IEEE Communications Magazine, December 1999, <<https://ieeexplore.ieee.org/document/809383>>.
- [AWD5]    Awduche, D., "An Approach to Optimal Peering Between Autonomous Systems in the Internet", Paper International Conference on Computer Communications and Networks (ICCCN'98), October 1998, <<https://ieeexplore.ieee.org/document/998795>>.
- [FLJA93]    Floyd, S. and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", Article IEEE/ACM Transactions on Networking, Vol. 1, p. 387-413, November 1993, <<https://www.icir.org/floyd/papers/early.twocolumn.pdf>>.
- [FLOY94]    Floyd, S., "TCP and Explicit Congestion Notification", Article ACM Computer Communication Review, V. 24, No. 5, p. 10-23, October 1994, <[https://www.icir.org/floyd/papers/tcp\\_ecn.4.pdf](https://www.icir.org/floyd/papers/tcp_ecn.4.pdf)>.
- [FT00]    Fortz, B. and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights", Article IEEE INFOCOM 2000, March 2000, <[https://www.cs.cornell.edu/courses/cs619/2004fa/documents/ospf\\_opt.pdf](https://www.cs.cornell.edu/courses/cs619/2004fa/documents/ospf_opt.pdf)>.
- [FT01]    Fortz, B. and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World", n.d., <<http://www.research.att.com/~mthorup/PAPERS/papers.html>>.
- [HUSS87]    Hurley, B.R., Seidl, C.J.R., and W.F. Sewel, "A Survey of Dynamic Routing Methods for Circuit-Switched Traffic", Article IEEE Communication Magazine, September 1987, <<https://dlnext.acm.org/doi/10.1109/MCOM.1987.1093695>>.
- [I-D.ietf-alto-performance-metrics]  
Wu, Q., Yang, Y. R., Lee, Y., Dhody, D., Randriamasy, S., and L. M. C. Murillo, "ALTO Performance Cost Metrics", Work in Progress, Internet-Draft, draft-ietf-alto-performance-metrics-28, 21 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-alto-performance-metrics-28.txt>>.

[I-D.ietf-bess-evpn-unequal-lb]

Malhotra, N., Sajassi, A., Rabadan, J., Drake, J., Lingala, A., and S. Thoria, "Weighted Multi-Path Procedures for EVPN Multi-Homing", Work in Progress, Internet-Draft, draft-ietf-bess-evpn-unequal-lb-15, 17 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-bess-evpn-unequal-lb-15.txt>>.

[I-D.ietf-idr-performance-routing]

Xu, X., Hegde, S., Talaulikar, K., Boucadair, M., and C. Jacquenet, "Performance-based BGP Routing Mechanism", Work in Progress, Internet-Draft, draft-ietf-idr-performance-routing-03, 22 December 2020, <<https://www.ietf.org/archive/id/draft-ietf-idr-performance-routing-03.txt>>.

[I-D.ietf-idr-segment-routing-te-policy]

Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", Work in Progress, Internet-Draft, draft-ietf-idr-segment-routing-te-policy-16, 7 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-idr-segment-routing-te-policy-16.txt>>.

[I-D.ietf-lsr-flex-algo]

Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", Work in Progress, Internet-Draft, draft-ietf-lsr-flex-algo-18, 25 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-lsr-flex-algo-18.txt>>.

[I-D.ietf-lsr-ip-flexalgo]

Britto, W., Hegde, S., Kaneriyi, P., Shetty, R., Bonica, R., and P. Psenak, "IGP Flexible Algorithms (Flex-Algorithm) In IP Networks", Work in Progress, Internet-Draft, draft-ietf-lsr-ip-flexalgo-04, 19 December 2021, <<https://www.ietf.org/archive/id/draft-ietf-lsr-ip-flexalgo-04.txt>>.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]

Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-08, 21 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-08.txt>>.

- [I-D.ietf-spring-segment-routing-policy]  
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-21, 19 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-21.txt>>.
- [I-D.ietf-teas-enhanced-vpn]  
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", Work in Progress, Internet-Draft, draft-ietf-teas-enhanced-vpn-10, 6 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-enhanced-vpn-10.txt>>.
- [I-D.ietf-teas-ietf-network-slices]  
Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-08, 6 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-08.txt>>.
- [I-D.ietf-tewg-qos-routing]  
Ash, G., "Traffic Engineering & QoS Methods for IP-, ATM-, & Based Multiservice Networks", Work in Progress, Internet-Draft, draft-ietf-tewg-qos-routing-04, 15 October 2001, <<https://www.ietf.org/archive/id/draft-ietf-tewg-qos-routing-04.txt>>.
- [I-D.irtf-nmrg-ibn-concepts-definitions]  
Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", Work in Progress, Internet-Draft, draft-irtf-nmrg-ibn-concepts-definitions-06, 15 December 2021, <<https://www.ietf.org/archive/id/draft-irtf-nmrg-ibn-concepts-definitions-06.txt>>.
- [ITU-E600] "Terms and Definitions of Traffic Engineering", Recommendation ITU-T Recommendation E.600, March 1993, <<https://www.itu.int/rec/T-REC-E.600/en>>.
- [ITU-E701] "Reference Connections for Traffic Engineering", Recommendation ITU-T Recommendation E.701, October 1993, <<https://www.itu.int/rec/T-REC-E.701/en>>.

- [ITU-E801] "Framework for Service Quality Agreement", Recommendation ITU-T Recommendation E.801, October 1996, <<https://www.itu.int/rec/T-REC-E.801/en>>.
- [MA] Ma, Q., "Quality of Service Routing in Integrated Services Networks", Ph.D. PhD Dissertation, CMU-CS-98-138, CMU, 1998, <<https://apps.dtic.mil/sti/pdfs/ADA352299.pdf>>.
- [MATE] Elwalid, A., Jin, C., Low, S., and I. Widjaja, "MATE - MPLS Adaptive Traffic Engineering", Proceedings INFOCOM'01, April 2001, <<https://www.yumpu.com/en/document/view/35140398/mate-mpls-adaptive-traffic-engineering-infocom-ieee-xplore/8>>.
- [MCQ80] McQuillan, J.M., Richer, I., and E.C. Rosen, "The New Routing Algorithm for the ARPANET", Transaction IEEE Transactions on Communications, vol. 28, no. 5, p. 711-719, May 1980, <The New Routing Algorithm for the ARPANET>.
- [MR99] Mitra, D. and K.G. Ramakrishnan, "A Case Study of Multiservice, Multipriority Traffic Engineering Design for Data Networks", Proceedings Globecom'99, December 1999, <<https://ieeexplore.ieee.org/document/830281>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC1102] Clark, D., "Policy routing in Internet protocols", RFC 1102, DOI 10.17487/RFC1102, May 1989, <<https://www.rfc-editor.org/info/rfc1102>>.
- [RFC1104] Braun, H., "Models of policy based routing", RFC 1104, DOI 10.17487/RFC1104, June 1989, <<https://www.rfc-editor.org/info/rfc1104>>.
- [RFC1992] Castineyra, I., Chiappa, N., and M. Steenstrup, "The Nimrod Routing Architecture", RFC 1992, DOI 10.17487/RFC1992, August 1996, <<https://www.rfc-editor.org/info/rfc1992>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.

- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<https://www.rfc-editor.org/info/rfc2330>>.
- [RFC2386] Crawley, E., Nair, R., Rajagopalan, B., and H. Sandick, "A Framework for QoS-based Routing in the Internet", RFC 2386, DOI 10.17487/RFC2386, August 1998, <<https://www.rfc-editor.org/info/rfc2386>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <<https://www.rfc-editor.org/info/rfc2475>>.
- [RFC2597] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, DOI 10.17487/RFC2597, June 1999, <<https://www.rfc-editor.org/info/rfc2597>>.
- [RFC2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, DOI 10.17487/RFC2678, September 1999, <<https://www.rfc-editor.org/info/rfc2678>>.
- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <<https://www.rfc-editor.org/info/rfc2702>>.
- [RFC2722] Brownlee, N., Mills, C., and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2722, DOI 10.17487/RFC2722, October 1999, <<https://www.rfc-editor.org/info/rfc2722>>.
- [RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, DOI 10.17487/RFC2753, January 2000, <<https://www.rfc-editor.org/info/rfc2753>>.

- [RFC2961]    Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F.,  
              and S. Molendini, "RSVP Refresh Overhead Reduction  
              Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001,  
              <<https://www.rfc-editor.org/info/rfc2961>>.
- [RFC2998]    Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L.,  
              Speer, M., Braden, R., Davie, B., Wroclawski, J., and E.  
              Felstaine, "A Framework for Integrated Services Operation  
              over Diffserv Networks", RFC 2998, DOI 10.17487/RFC2998,  
              November 2000, <<https://www.rfc-editor.org/info/rfc2998>>.
- [RFC3031]    Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol  
              Label Switching Architecture", RFC 3031,  
              DOI 10.17487/RFC3031, January 2001,  
              <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3086]    Nichols, K. and B. Carpenter, "Definition of  
              Differentiated Services Per Domain Behaviors and Rules for  
              their Specification", RFC 3086, DOI 10.17487/RFC3086,  
              April 2001, <<https://www.rfc-editor.org/info/rfc3086>>.
- [RFC3124]    Balakrishnan, H. and S. Seshan, "The Congestion Manager",  
              RFC 3124, DOI 10.17487/RFC3124, June 2001,  
              <<https://www.rfc-editor.org/info/rfc3124>>.
- [RFC3209]    Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V.,  
              and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP  
              Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001,  
              <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3270]    Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen,  
              P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-  
              Protocol Label Switching (MPLS) Support of Differentiated  
              Services", RFC 3270, DOI 10.17487/RFC3270, May 2002,  
              <<https://www.rfc-editor.org/info/rfc3270>>.
- [RFC3272]    Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X.  
              Xiao, "Overview and Principles of Internet Traffic  
              Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002,  
              <<https://www.rfc-editor.org/info/rfc3272>>.
- [RFC3469]    Sharma, V., Ed. and F. Hellstrand, Ed., "Framework for  
              Multi-Protocol Label Switching (MPLS)-based Recovery",  
              RFC 3469, DOI 10.17487/RFC3469, February 2003,  
              <<https://www.rfc-editor.org/info/rfc3469>>.

- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <<https://www.rfc-editor.org/info/rfc3945>>.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<https://www.rfc-editor.org/info/rfc4090>>.
- [RFC4124] Le Faucheur, F., Ed., "Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering", RFC 4124, DOI 10.17487/RFC4124, June 2005, <<https://www.rfc-editor.org/info/rfc4124>>.
- [RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <<https://www.rfc-editor.org/info/rfc4203>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, DOI 10.17487/RFC4594, August 2006, <<https://www.rfc-editor.org/info/rfc4594>>.
- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC4872] Lang, J.P., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <<https://www.rfc-editor.org/info/rfc4872>>.



- [RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, DOI 10.17487/RFC4873, May 2007, <<https://www.rfc-editor.org/info/rfc4873>>.
- [RFC4920] Farrel, A., Ed., Satyanarayana, A., Iwata, A., Fujita, N., and G. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE", RFC 4920, DOI 10.17487/RFC4920, July 2007, <<https://www.rfc-editor.org/info/rfc4920>>.
- [RFC5151] Farrel, A., Ed., Ayyangar, A., and JP. Vasseur, "Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 5151, DOI 10.17487/RFC5151, February 2008, <<https://www.rfc-editor.org/info/rfc5151>>.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <<https://www.rfc-editor.org/info/rfc5250>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed., "Traffic Engineering Extensions to OSPF Version 3", RFC 5329, DOI 10.17487/RFC5329, September 2008, <<https://www.rfc-editor.org/info/rfc5329>>.
- [RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <<https://www.rfc-editor.org/info/rfc5331>>.
- [RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, DOI 10.17487/RFC5394, December 2008, <<https://www.rfc-editor.org/info/rfc5394>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <<https://www.rfc-editor.org/info/rfc5541>>.

- [RFC5557] Lee, Y., Le Roux, J.L., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, DOI 10.17487/RFC5557, July 2009, <<https://www.rfc-editor.org/info/rfc5557>>.
- [RFC5623] Oki, E., Takeda, T., Le Roux, J.L., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, DOI 10.17487/RFC5623, September 2009, <<https://www.rfc-editor.org/info/rfc5623>>.
- [RFC5664] Halevy, B., Welch, B., and J. Zelenka, "Object-Based Parallel NFS (pNFS) Operations", RFC 5664, DOI 10.17487/RFC5664, January 2010, <<https://www.rfc-editor.org/info/rfc5664>>.
- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <<https://www.rfc-editor.org/info/rfc5693>>.
- [RFC6107] Shiomoto, K., Ed. and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <<https://www.rfc-editor.org/info/rfc6107>>.
- [RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, DOI 10.17487/RFC6119, February 2011, <<https://www.rfc-editor.org/info/rfc6119>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6372] Sprecher, N., Ed. and A. Farrel, Ed., "MPLS Transport Profile (MPLS-TP) Survivability Framework", RFC 6372, DOI 10.17487/RFC6372, September 2011, <<https://www.rfc-editor.org/info/rfc6372>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.

- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<https://www.rfc-editor.org/info/rfc6790>>.
- [RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <<https://www.rfc-editor.org/info/rfc6805>>.
- [RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <<https://www.rfc-editor.org/info/rfc7149>>.
- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <<https://www.rfc-editor.org/info/rfc7285>>.
- [RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <<https://www.rfc-editor.org/info/rfc7390>>.
- [RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <<https://www.rfc-editor.org/info/rfc7426>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, DOI 10.17487/RFC7491, March 2015, <<https://www.rfc-editor.org/info/rfc7491>>.

- [RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <<https://www.rfc-editor.org/info/rfc7567>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <<https://www.rfc-editor.org/info/rfc7680>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <<https://www.rfc-editor.org/info/rfc7923>>.
- [RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <<https://www.rfc-editor.org/info/rfc7926>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<https://www.rfc-editor.org/info/rfc8051>>.
- [RFC8189] Randriamasy, S., Roome, W., and N. Schwan, "Multi-Cost Application-Layer Traffic Optimization (ALTO)", RFC 8189, DOI 10.17487/RFC8189, October 2017, <<https://www.rfc-editor.org/info/rfc8189>>.
- [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <<https://www.rfc-editor.org/info/rfc8231>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.
- [RFC8283] Farrel, A., Ed., Zhao, Q., Ed., Li, Z., and C. Zhou, "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", RFC 8283, DOI 10.17487/RFC8283, December 2017, <<https://www.rfc-editor.org/info/rfc8283>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <<https://www.rfc-editor.org/info/rfc8570>>.

- [RFC8571] Ginsberg, L., Ed., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, DOI 10.17487/RFC8571, March 2019, <<https://www.rfc-editor.org/info/rfc8571>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <<https://www.rfc-editor.org/info/rfc8664>>.
- [RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March 2020, <<https://www.rfc-editor.org/info/rfc8684>>.
- [RFC8685] Zhang, F., Zhao, Q., Gonzalez de Dios, O., Casellas, R., and D. King, "Path Computation Element Communication Protocol (PCEP) Extensions for the Hierarchical Path Computation Element (H-PCE) Architecture", RFC 8685, DOI 10.17487/RFC8685, December 2019, <<https://www.rfc-editor.org/info/rfc8685>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.
- [RFC8803] Bonaventure, O., Ed., Boucadair, M., Ed., Gundavelli, S., Seo, S., and B. Hesmans, "0-RTT TCP Convert Protocol", RFC 8803, DOI 10.17487/RFC8803, July 2020, <<https://www.rfc-editor.org/info/rfc8803>>.
- [RFC8896] Randriamasy, S., Yang, R., Wu, Q., Deng, L., and N. Schwan, "Application-Layer Traffic Optimization (ALTO) Cost Calendar", RFC 8896, DOI 10.17487/RFC8896, November 2020, <<https://www.rfc-editor.org/info/rfc8896>>.

- [RFC8938] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC 8938, DOI 10.17487/RFC8938, November 2020, <<https://www.rfc-editor.org/info/rfc8938>>.
- [RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <<https://www.rfc-editor.org/info/rfc8955>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.
- [RFC9023] Varga, B., Ed., Farkas, J., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: IP over IEEE 802.1 Time-Sensitive Networking (TSN)", RFC 9023, DOI 10.17487/RFC9023, June 2021, <<https://www.rfc-editor.org/info/rfc9023>>.
- [RR94] Rodrigues, M.A. and K.G. Ramakrishnan, "Optimal Routing in Shortest Path Data Networks", Proceedings ITS'94, Rio de Janeiro, Brazil, 1994, <<https://onlinelibrary.wiley.com/doi/abs/10.1002/bltj.2267>>.
- [SLDC98] Suter, B., Lakshman, T., Stiliadis, D., and A. Choudhury, "Design Considerations for Supporting TCP with Per-flow Queueing", Proceedings INFOCOM'98, p. 299-306, 1998, <<https://ieeexplore.ieee.org/document/659666>>.
- [WANG] Wang, Y., Wang, Z., and L. Zhang, "Internet traffic engineering without full mesh overlaying", Proceedings INFOCOM'2001, April 2001, <<https://ieeexplore.ieee.org/document/916782>>.
- [XIAO] Xiao, X., Hannan, A., Bailey, B., and L. Ni, "Traffic Engineering with MPLS in the Internet", Article IEEE Network Magazine, March 2000, <<https://courses.cs.washington.edu/courses/cse561/02au/papers/xiao-mppls-net00.pdf>>.
- [YARE95] Yang, C. and A. Reddy, "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks", Article IEEE Network Magazine, p. 34-45, 1995, <[http://www.cs.uccs.edu/~zbo/teaching/CS522/Projects/Taxonomy\\_Network1995.pdf](http://www.cs.uccs.edu/~zbo/teaching/CS522/Projects/Taxonomy_Network1995.pdf)>.

## Appendix A.    Historic Overview

### A.1.    Traffic Engineering in Classical Telephone Networks

This subsection presents a brief overview of traffic engineering in telephone networks which often relates to the way user traffic is steered from an originating node to the terminating node. This subsection presents a brief overview of this topic. A detailed description of the various routing strategies applied in telephone networks is included in the book by G. Ash [ASH2].

The early telephone network relied on static hierarchical routing, whereby routing patterns remained fixed independent of the state of the network or time of day. The hierarchy was intended to accommodate overflow traffic, improve network reliability via alternate routes, and prevent call looping by employing strict hierarchical rules. The network was typically over-provisioned since a given fixed route had to be dimensioned so that it could carry user traffic during a busy hour of any busy day. Hierarchical routing in the telephony network was found to be too rigid upon the advent of digital switches and stored program control which were able to manage more complicated TE rules.

Dynamic routing was introduced to alleviate the routing inflexibility in the static hierarchical routing so that the network would operate more efficiently. This resulted in significant economic gains [HUSS87]. Dynamic routing typically reduces the overall loss probability by 10 to 20 percent (compared to static hierarchical routing). Dynamic routing can also improve network resilience by recalculating routes on a per-call basis and periodically updating routes.

There are three main types of dynamic routing in the telephone network. They are time-dependent routing, state-dependent routing (SDR), and event dependent routing (EDR).

In time-dependent routing, regular variations in traffic loads (such as time of day or day of week) are exploited in pre-planned routing tables. In state-dependent routing, routing tables are updated online according to the current state of the network (e.g., traffic demand, utilization, etc.). In event dependent routing, routing changes are triggered by events (such as call setups encountering congested or blocked links) whereupon new paths are searched out using learning models. EDR methods are real-time adaptive, but they do not require global state information as does SDR. Examples of EDR schemes include the dynamic alternate routing (DAR) from BT, the state-and-time dependent routing (STR) from NTT, and the success-to-the-top (STT) routing from AT&T.



Dynamic non-hierarchical routing (DNHR) is an example of dynamic routing that was introduced in the AT&T toll network in the 1980's to respond to time-dependent information such as regular load variations as a function of time. Time-dependent information in terms of load may be divided into three timescales: hourly, weekly, and yearly. Correspondingly, three algorithms are defined to pre-plan the routing tables. The network design algorithm operates over a year-long interval while the demand servicing algorithm operates on a weekly basis to fine tune link sizes and routing tables to correct forecast errors on the yearly basis. At the smallest timescale, the routing algorithm is used to make limited adjustments based on daily traffic variations. Network design and demand servicing are computed using offline calculations. Typically, the calculations require extensive searches on possible routes. On the other hand, routing may need online calculations to handle crankback. DNHR adopts a "two-link" approach whereby a path can consist of two links at most. The routing algorithm presents an ordered list of route choices between an originating switch and a terminating switch. If a call overflows, a via switch (a tandem exchange between the originating switch and the terminating switch) would send a crankback signal to the originating switch. This switch would then select the next route, and so on, until there are no alternative routes available in which the call is blocked. Note that the concept of crankback found its way into GMPLS as [RFC4920].

#### A.2. Evolution of Traffic Engineering in Packet Networks

This subsection reviews related prior work that was intended to improve the performance of data networks. Indeed, optimization of the performance of data networks started in the early days of the ARPANET. Other early commercial networks such as SNA also recognized the importance of performance optimization and service differentiation.

In terms of traffic management, the Internet has been a best effort service environment until recently. In particular, very limited traffic management capabilities existed in IP networks to provide differentiated queue management and scheduling services to packets belonging to different classes.

In terms of routing control, the Internet has employed distributed protocols for intra-domain routing. These protocols are highly scalable and resilient. However, they are based on simple algorithms for path selection which have very limited functionality to allow flexible control of the path selection process.

In the following subsections, the evolution of practical traffic engineering mechanisms in IP networks and its predecessors are reviewed.

#### A.2.1. Adaptive Routing in the ARPANET

The early ARPANET recognized the importance of adaptive routing where routing decisions were based on the current state of the network [MCQ80]. Early minimum delay routing approaches forwarded each packet to its destination along a path for which the total estimated transit time was the smallest. Each node maintained a table of network delays, representing the estimated delay that a packet would experience along a given path toward its destination. The minimum delay table was periodically transmitted by a node to its neighbors. The shortest path, in terms of hop count, was also propagated to give the connectivity information.

One drawback to this approach is that dynamic link metrics tend to create "traffic magnets" causing congestion to be shifted from one location of a network to another location, resulting in oscillation and network instability.

#### A.2.2. Dynamic Routing in the Internet

The Internet evolved from the ARPANET and adopted dynamic routing algorithms with distributed control to determine the paths that packets should take en-route to their destinations. The routing algorithms are adaptations of shortest path algorithms where costs are based on link metrics. The link metric can be based on static or dynamic quantities. The link metric based on static quantities may be assigned administratively according to local criteria. The link metric based on dynamic quantities may be a function of a network congestion measure such as delay or packet loss.

It was apparent early that static link metric assignment was inadequate because it can easily lead to unfavorable scenarios in which some links become congested while others remain lightly loaded. One of the many reasons for the inadequacy of static link metrics is that link metric assignment was often done without considering the traffic matrix in the network. Also, the routing protocols did not take traffic attributes and capacity constraints into account when making routing decisions. This results in traffic concentration being localized in subsets of the network infrastructure and potentially causing congestion. Even if link metrics are assigned in accordance with the traffic matrix, unbalanced loads in the network can still occur due to a number factors including:

- \* Resources may not be deployed in the most optimal locations from a routing perspective.
- \* Forecasting errors in traffic volume and/or traffic distribution.
- \* Dynamics in traffic matrix due to the temporal nature of traffic patterns, BGP policy change from peers, etc.

The inadequacy of the legacy Internet interior gateway routing system is one of the factors motivating the interest in path oriented technology with explicit routing and constraint-based routing capability such as MPLS.

#### A.2.3. ToS Routing

Type-of-Service (ToS) routing involves different routes going to the same destination with selection dependent upon the ToS field of an IP packet [RFC2474]. The ToS classes may be classified as low delay and high throughput. Each link is associated with multiple link costs and each link cost is used to compute routes for a particular ToS. A separate shortest path tree is computed for each ToS. The shortest path algorithm must be run for each ToS resulting in very expensive computation. Classical ToS-based routing is now outdated as the IP header field has been replaced by a Diffserv field. Effective TE is difficult to perform in classical ToS-based routing because each class still relies exclusively on shortest path routing which results in localization of traffic concentration within the network.

#### A.2.4. Equal Cost Multi-Path

Equal Cost Multi-Path (ECMP) is another technique that attempts to address the deficiency in the Shortest Path First (SPF) interior gateway routing systems [RFC2328]. In the classical SPF algorithm, if two or more shortest paths exist to a given destination, the algorithm will choose one of them. The algorithm is modified slightly in ECMP so that if two or more equal cost shortest paths exist between two nodes, the traffic between the nodes is distributed among the multiple equal-cost paths. Traffic distribution across the equal-cost paths is usually performed in one of two ways: (1) packet-based in a round-robin fashion, or (2) flow-based using hashing on source and destination IP addresses and possibly other fields of the IP header. The first approach can easily cause out-of-order packets while the second approach is dependent upon the number and distribution of flows. Flow-based load sharing may be unpredictable in an enterprise network where the number of flows is relatively small and less heterogeneous (for example, hashing may not be uniform), but it is generally effective in core public networks where the number of flows is large and heterogeneous.

In ECMP, link costs are static and bandwidth constraints are not considered, so ECMP attempts to distribute the traffic as equally as possible among the equal-cost paths independent of the congestion status of each path. As a result, given two equal-cost paths, it is possible that one of the paths will be more congested than the other. Another drawback of ECMP is that load sharing cannot be achieved on multiple paths which have non-identical costs.

MPLS (through the Entropy Label [RFC6790]) and IPv6 (through the Flow Label [RFC6437]) allow the ingress node to set a hash value that can be used in the network for ECMP. These mechanisms can be used to take into account other constraints and, if there is sufficient visibility into the network (perhaps through a central controller), these mechanisms can be used to achieve for more evenly distributed load balancing giving a modicum of traffic steering.

#### A.2.5. Nimrod

Nimrod was a routing system developed to provide heterogeneous service specific routing in the Internet, while taking multiple constraints into account [RFC1992]. Essentially, Nimrod was a link state routing protocol to support path oriented packet forwarding. It used the concept of maps to represent network connectivity and services at multiple levels of abstraction. Mechanisms allowed restriction of the distribution of routing information.

Even though Nimrod did not enjoy deployment in the public Internet, a number of key concepts incorporated into the Nimrod architecture, such as explicit routing which allows selection of paths at originating nodes, are beginning to find applications in some recent constraint-based routing initiatives.

### A.3. Development of Internet Traffic Engineering

#### A.3.1. Overlay Model

In the overlay model, a virtual-circuit network, such as Synchronous Optical Network / Synchronous Digital Hierarchy (SONET/SDH), Optical Transport Network (OTN), or Wavelength Division Multiplexing (WDM), provides virtual-circuit connectivity between routers that are located at the edges of a virtual-circuit cloud. In this mode, two routers that are connected through a virtual circuit see a direct adjacency between themselves independent of the physical route taken by the virtual circuit through the ATM, frame relay, or WDM network. Thus, the overlay model essentially decouples the logical topology that routers see from the physical topology that the ATM, frame relay, or WDM network manages. The overlay model based on ATM or frame relay enables a network administrator or an automaton to employ

TE concepts to perform path optimization by re-configuring or rearranging the virtual circuits so that a virtual circuit on a congested or sub-optimal physical link can be re-routed to a less congested or more optimal one. In the overlay model, TE is also employed to establish relationships between the traffic management parameters (e.g., Peak Cell Rate, Sustained Cell Rate, and Maximum Burst Size for ATM) of the virtual- circuit technology and the actual traffic that traverses each circuit. These relationships can be established based upon known or projected traffic profiles, and some other factors.

## Appendix B. Overview of Traffic Engineering Related Work in Other SDOs

### B.1. Overview of ITU Activities Related to Traffic Engineering

This section provides an overview of prior work within the ITU-T pertaining to traffic engineering in traditional telecommunications networks.

ITU-T Recommendations E.600 [ITU-E600], E.701 [ITU-E701], and E.801 [ITU-E801] address TE issues in traditional telecommunications networks. Recommendation E.600 provides a vocabulary for describing TE concepts, while E.701 defines reference connections, Grade of Service (GoS), and traffic parameters for ISDN. Recommendation E.701 uses the concept of a reference connection to identify representative cases of different types of connections without describing the specifics of their actual realizations by different physical means. As defined in Recommendation E.600, "a connection is an association of resources providing means for communication between two or more devices in, or attached to, a telecommunication network." Also, E.600 defines "a resource as any set of physically or conceptually identifiable entities within a telecommunication network, the use of which can be unambiguously determined" [ITU-E600]. There can be different types of connections as the number and types of resources in a connection may vary.

Typically, different network segments are involved in the path of a connection. For example, a connection may be local, national, or international. The purposes of reference connections are to clarify and specify traffic performance issues at various interfaces between different network domains. Each domain may consist of one or more service provider networks.

Reference connections provide a basis to define grade of service (GoS) parameters related to TE within the ITU-T framework. As defined in E.600, "GoS refers to a number of traffic engineering variables which are used to provide a measure of the adequacy of a group of resources under specified conditions." These GoS variables

may be probability of loss, dial tone, delay, etc. They are essential for network internal design and operation as well as for component performance specification.

GoS is different from quality of service (QoS) in the ITU framework. QoS is the performance perceivable by a telecommunication service user and expresses the user's degree of satisfaction of the service. QoS parameters focus on performance aspects observable at the service access points and network interfaces, rather than their causes within the network. GoS, on the other hand, is a set of network oriented measures which characterize the adequacy of a group of resources under specified conditions. For a network to be effective in serving its users, the values of both GoS and QoS parameters must be related, with GoS parameters typically making a major contribution to the QoS.

Recommendation E.600 stipulates that a set of GoS parameters must be selected and defined on an end-to-end basis for each major service category provided by a network to assist the network provider with improving efficiency and effectiveness of the network. Based on a selected set of reference connections, suitable target values are assigned to the selected GoS parameters under normal and high load conditions. These end-to-end GoS target values are then apportioned to individual resource components of the reference connections for dimensioning purposes.

## Appendix C.   Summary of Changes Since RFC 3272

The changes to this document since RFC 3272 are substantial and not easily summarized as section-by-section changes. The material in the document has been moved around considerably, some of it removed, and new text added.

The approach taken here is to list the table of content of both the previous RFC and this document saying, respectively, where the text has been placed and where the text came from.

### C.1.   RFC 3272

1.0 Introduction:   Edited in place in Section 1.

1.1 What is Internet Traffic Engineering?:   Edited in place in Section 1.1.

1.2 Scope:   Moved to Section 1.3.

1.3 Terminology:   Moved to Section 1.4 with some obsolete terms removed and a little editing.

- 2.0 Background:    Retained as Section 2 with some text removed.
- 2.1 Context of Internet Traffic Engineering:    Retained as Section 2.1.
- 2.2 Network Context:    Rewritten as Section 2.2.
- 2.3 Problem Context:    Rewritten as Section 2.3.
  - 2.3.1 Congestion and its Ramifications:    Retained as Section 2.3.1.
- 2.4 Solution Context:    Edited as Section 2.4.
  - 2.4.1 Combating the Congestion Problem:    Reformatted as Section 2.4.1.
- 2.5 Implementation and Operational Context:    Retained as Section 2.5.
- 3.0 Traffic Engineering Process Model:    Retained as Section 3.
  - 3.1 Components of the Traffic Engineering Process Model:    Retained as Section 3.1.
  - 3.2 Measurement:    Merged into Section 3.1.
  - 3.3 Modeling, Analysis, and Simulation:    Merged into Section 3.1.
  - 3.4 Optimization:    Merged into Section 3.1.
- 4.0 Historical Review and Recent Developments:    Retained as Section 5, but the very historic aspects moved to Appendix A.
  - 4.1 Traffic Engineering in Classical Telephone Networks:    Moved to Appendix A.1.
  - 4.2 Evolution of Traffic Engineering in the Internet:    Moved to Appendix A.2.
    - 4.2.1 Adaptive Routing in ARPANET:    Moved to Appendix A.2.1.
    - 4.2.2 Dynamic Routing in the Internet:    Moved to Appendix A.2.2.
    - 4.2.3 ToS Routing:    Moved to Appendix A.2.3.
    - 4.2.4 Equal Cost Multi-Path:    Moved to Appendix A.2.4.

- 4.2.5 Nimrod:    Moved to Appendix A.2.5.
- 4.3 Overlay Model:    Moved to Appendix A.3.1.
- 4.4 Constraint-Based Routing:    Retained as Section 5.1.3.1, but moved into Section 5.1.
- 4.5 Overview of Other IETF Projects Related to Traffic Engineering:    Retained as Section 5.1 with many new subsections.
  - 4.5.1 Integrated Services:    Retained as Section 5.1.1.1.
  - 4.5.2 RSVP:    Retained as Section 5.1.3.2 with some edits.
  - 4.5.3 Differentiated Services:    Retained as Section 5.1.1.2.
  - 4.5.4 MPLS:    Retained as Section 5.1.3.3.
  - 4.5.5 IP Performance Metrics:    Retained as Section 5.1.3.5.
  - 4.5.6 Flow Measurement:    Retained as Section 5.1.3.6 with some reformatting.
  - 4.5.7 Endpoint Congestion Management:    Retained as Section 5.1.3.7.
- 4.6 Overview of ITU Activities Related to Traffic Engineering:    Moved to Appendix B.1.
- 4.7 Content Distribution:    Retained as Section 5.2.
- 5.0 Taxonomy of Traffic Engineering Systems:    Retained as Section 4.
  - 5.1 Time-Dependent Versus State-Dependent:    Retained as Section 4.1.
  - 5.2 Offline Versus Online:    Retained as Section 4.2.
  - 5.3 Centralized Versus Distributed:    Retained as Section 4.3 with additions.
  - 5.4 Local Versus Global:    Retained as Section 4.4.
  - 5.5 Prescriptive Versus Descriptive:    Retained as Section 4.5 with additions.
  - 5.6 Open-Loop Versus Closed-Loop:    Retained as Section 4.6.



- 5.7 Tactical vs Strategic: Retained as Section 4.7.
  - 6.0 Recommendations for Internet Traffic Engineering: Retained as Section 6.
    - 6.1 Generic Non-functional Recommendations: Retained as Section 6.1.
    - 6.2 Routing Recommendations: Retained as Section 6.2 with edits.
    - 6.3 Traffic Mapping Recommendations: Retained as Section 6.3.
    - 6.4 Measurement Recommendations: Retained as Section 6.4.
    - 6.5 Network Survivability: Retained as Section 6.6.
      - 6.5.1 Survivability in MPLS Based Networks: Retained as Section 6.6.1.
      - 6.5.2 Protection Option: Retained as Section 6.6.2.
    - 6.6 Traffic Engineering in Diffserv Environments: Retained as Section 6.8 with edits.
    - 6.7 Network Controllability: Retained as Section 6.9.
  - 7.0 Inter-Domain Considerations: Retained as Section 7.
  - 8.0 Overview of Contemporary TE Practices in Operational IP Networks: Retained as Section 8.
  - 9.0 Conclusion: Removed.
  - 10.0 Security Considerations: Retained as Section 9 with considerable new text.
- C.2. This Document
- \* Section 1: Based on Section 1 of RFC 3272.
    - Section 1.1: Based on Section 1.1 of RFC 3272.
    - Section 1.2: New for this document.
    - Section 1.3: Based on Section 1.2 of RFC 3272.
    - Section 1.4: Based on Section 1.3 of RFC 3272.

- \* Section 2: Based on Section 2. of RFC 3272.
  - Section 2.1: Based on Section 2.1 of RFC 3272.
  - Section 2.2: Based on Section 2.2 of RFC 3272.
  - Section 2.3: Based on Section 2.3 of RFC 3272.
    - o Section 2.3.1: Based on Section 2.3.1 of RFC 3272.
  - Section 2.4: Based on Section 2.4 of RFC 3272.
    - o Section 2.4.1: Based on Section 2.4.1 of RFC 3272.
  - Section 2.5: Based on Section 2.5 of RFC 3272.
- \* Section 3: Based on Section 3 of RFC 3272.
  - Section 3.1: Based on Sections 3.1, 3.2, 3.3, and 3.4 of RFC 3272.
- \* Section 4: Based on Section 5 of RFC 3272.
  - Section 4.1: Based on Section 5.1 of RFC 3272.
  - Section 4.2: Based on Section 5.2 of RFC 3272.
  - Section 4.3: Based on Section 5.3 of RFC 3272.
    - o Section 4.3.1: New for this document.
    - o Section 4.3.2: New for this document.
  - Section 4.4: Based on Section 5.4 of RFC 3272.
  - Section 4.5: Based on Section 5.5 of RFC 3272.
    - o Section 4.5.1: New for this document.
  - Section 4.6: Based on Section 5.6 of RFC 3272.
  - Section 4.7: Based on Section 5.7 of RFC 3272.
- \* Section 5: Based on Section 4 of RFC 3272.
  - Section 5.1: Based on Section 4.5 of RFC 3272.
    - o Section 5.1.1.1: Based on Section 4.5.1 of RFC 3272.

- o Section 5.1.1.2: Based on Section 4.5.3 of RFC 3272.
- o Section 5.1.1.3: New for this document.
- o Section 5.1.1.4: New for this document.
- o Section 5.1.1.5: New for this document.
- o Section 5.1.2.1: New for this document.
- o Section 5.1.2.2: New for this document.
- o Section 5.1.2.3: New for this document.
- o Section 5.1.3.1: Based on Section 4.4 of RFC 3272.
  - + Section 5.1.3.1.1: New for this document.
- o Section 5.1.3.2: Based on Section 4.5.2 of RFC 3272.
- o Section 5.1.3.3: Based on Section 4.5.4 of RFC 3272.
- o Section 5.1.3.4: New for this document.
- o Section 5.1.3.5: Based on Section 4.5.5 of RFC 3272.
- o Section 5.1.3.6: Based on Section 4.5.6 of RFC 3272.
- o Section 5.1.3.7: Based on Section 4.5.7 of RFC 3272.
- o Section 5.1.3.8: New for this document.
- o Section 5.1.3.9: New for this document.
- o Section 5.1.3.10: New for this document.
- o Section 5.1.3.11: New for this document.
- o Section 5.1.3.12: New for this document.
- o Section 5.1.3.13: New for this document.
- Section 5.2: Based on Section 4.7 of RFC 3272.
- \* Section 6: Based on Section 6 of RFC 3272.
  - Section 6.1: Based on Section 6.1 of RFC 3272.

- Section 6.2: Based on Section 6.2 of RFC 3272.
- Section 6.3: Based on Section 6.3 of RFC 3272.
- Section 6.4: Based on Section 6.4 of RFC 3272.
- Section 6.5: New for this document.
- Section 6.6: Based on Section 6.5 of RFC 3272.
  - o Section 6.6.1: Based on Section 6.5.1 of RFC 3272.
  - o Section 6.6.2: Based on Section 6.5.2 of RFC 3272.
- Section 6.7: New for this document.
- Section 6.8: Based on Section 6.6. of RFC 3272.
- Section 6.9: Based on Section 6.7 of RFC 3272.
- \* Section 7: Based on Section 7 of RFC 3272.
- \* Section 8: Based on Section 8 of RFC 3272.
- \* Section 9: Based on Section 10 of RFC 3272.
- \* Appendix A: New for this document.
  - Appendix A.1: Based on Section 4.1 of RFC 3272.
  - Appendix A.2: Based on Section 4.2 of RFC 3272.
    - o Appendix A.2.1: Based on Section 4.2.1 of RFC 3272.
    - o Appendix A.2.2: Based on Section 4.2.2 of RFC 3272.
    - o Appendix A.2.3: Based on Section 4.2.3 of RFC 3272.
    - o Appendix A.2.4: Based on Section 4.2.4 of RFC 3272.
    - o Appendix A.2.5: Based on Section 4.2.5 of RFC 3272.
  - Appendix A.3: New for this document.
    - o Appendix A.3.1: Based on Section 4.3 of RFC 3272.
- \* Appendix B: New for this document.

- Appendix B.1: Based on Section 4.7 of RFC 3272.

Author's Address

Adrian Farrel (editor)  
Old Dog Consulting  
Email: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: October 2, 2021

D. King  
Old Dog Consulting  
J. Drake  
Juniper Networks  
H. Zheng  
Huawei Technologies  
A. Farrel  
Old Dog Consulting  
March 31, 2021

Applicability of Abstraction and Control of Traffic Engineered Networks  
(ACTN) to Network Slicing  
draft-king-teas-applicability-actn-slicing-10

Abstract

Network abstraction is a technique that can be applied to a network domain. It utilizes a set of policies to select network resources and obtain a view of potential connectivity across the network.

Network slicing is an approach to network operations that builds on the concept of network abstraction to provide programmability, flexibility, and modularity. It may use techniques such as Software Defined Networking (SDN) and Network Function Virtualization (NFV) to create multiple logical or virtual networks, each tailored for a set of services that share the same set of requirements.

Abstraction and Control of Traffic Engineered Networks (ACTN) is described in RFC 8453. It defines an SDN-based architecture that relies on the concept of network and service abstraction to detach network and service control from the underlying data plane.

This document outlines the applicability of ACTN to network slicing in a Traffic Engineering (TE) network that utilizes IETF technology. It also identifies the features of network slicing not currently within the scope of ACTN, and indicates where ACTN might be extended.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 2, 2021.

#### Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .   | 3  |
| 1.1. Terminology . . . . .  | 4  |
| 2. Requirements for Network Slicing . . . . .                                   | 5  |
| 2.1. Resource Slicing . . . . .   | 6  |
| 2.2. Network Virtualization . . . . .   | 6  |
| 2.3. Service Isolation . . . . .  | 6  |
| 2.4. Control and Orchestration . . . . .  | 7  |
| 3. Abstraction and Control of Traffic Engineered (TE) Networks (ACTN) . . . . . | 7  |
| 3.1. ACTN Virtual Network as a Network Slice . . . . .                          | 8  |
| 3.2. ACTN Virtual Network for Network Slice Aggregation . . . . .               | 9  |
| 3.3. Management Components for ACTN and Network Slicing . . . . .               | 9  |
| 3.4. Examples of ACTN Delivering Types of Network Slices . . . . .              | 10 |
| 3.4.1. ACTN Used for Virtual Private Line . . . . .                             | 10 |
| 3.4.2. ACTN Used for VPN Delivery Model . . . . .                               | 12 |
| 3.4.3. ACTN Used to Deliver a Virtual Consumer Network . . . . .                | 13 |
| 4. YANG Models . . . . .  | 15 |
| 4.1. Network Slice Service Mapping from TE to ACTN VN Models . . . . .          | 15 |
| 4.2. Interfaces and Yang Models . . . . .                                       | 16 |
| 4.3. ACTN VN Telemetry . . . . .  | 17 |
| 5. IANA Considerations . . . . .  | 18 |
| 6. Security Considerations . . . . .  | 18 |
| 7. Acknowledgements . . . . .   | 19 |
| 8. Contributors . . . . .   | 19 |

|                                     |    |
|-------------------------------------|----|
| 9. Informative References . . . . . | 19 |
| Authors' Addresses . . . . .        | 22 |

## 1. Introduction

The principles of network resource separation are not new. For years, the concept of separated overlay and logical (virtual) networking has existed, allowing multiple services to be deployed over a single physical network comprised of single or multiple layers. However, several key differences exist that differentiate overlay and virtual networking from network slicing.

A network slice is a virtual (that is, logical) network with its own network topology and a set of network resources that are used to provide connectivity that conforms to a specific Service Level Agreement (SLA) or set of Service Level Objectives (SLOs). The network resources used to realize a network slice belong to the network that is sliced. The resources may be assigned and dedicated to an individual slice, or they may be shared with other slices enabling different degrees of service guarantee and providing different levels of isolation between the traffic in each slice.

[I-D.ietf-teas-ietf-network-slice-definition] provides a number of useful definitions for network slicing in the context of IETF network technologies. In particular, that document defines the term "IETF network slice" to be the generic network slice concept applied to a network that uses IETF technologies. An IETF network slice could span multiple technologies (such as IP, MPLS, or optical) and multiple administrative domains. The logical network that is an IETF network slice may be kept separate from other concurrent logical networks each with independent control and management: each can be created or modified on demand. Since this document is focused entirely on IETF technologies, it uses the term "network slice" as a more concise expression. Further discussion on the topic of IETF network slices can be found in [I-D.ietf-teas-ietf-network-slice-framework].

At one end of the spectrum, a virtual private wire or a virtual private network (VPN) may be used to build a network slice. In these cases, the network slices do not require the service provider to isolate network resources for the provision of the service - the service is "virtual".

At the other end of the spectrum there may be a detailed description of a complex service that will meet the needs of a set of applications with connectivity and service function requirements that may include compute resource, storage capability, and access to content. Such a service may be requested dynamically (that is,



instantiated when an application needs it, and released when the application no longer needs it), and modified as the needs of the application change. This type of service is called an enhanced VPN and is described in more detail in [I-D.ietf-teas-enhanced-vpn]. It is often based on Traffic Engineering (TE) constructs in the underlay network.

Abstraction and Control of TE Networks (ACTN) [RFC8453] is a framework that facilitates the abstraction of underlying network resources to higher-layer applications and that allows network operators to create virtual networks for their customers through the abstraction of the operators' network resources.

As noted in [I-D.ietf-teas-ietf-network-slice-framework], ACTN is a toolset capable of delivering network slice functionality. This document outlines the application of ACTN and associated enabling technologies to provide network slicing in a network that utilizes IETF technologies such as IP, MPLS, or GMPLS. It describes how the ACTN functional components can be used to support model-driven partitioning of resources into variable-sized bandwidth units to facilitate network sharing and virtualization. Furthermore, the use of model-based interfaces to dynamically request the instantiation of virtual networks can be extended to encompass requesting and instantiation of specific service functions (which may be both physical or virtual), and to partition network resources such as compute resource, storage capability, and access to content. Finally, this document highlights how the ACTN approach might be extended to address the requirements of network slicing where the underlying network is TE-capable.

### 1.1. Terminology

As far as is possible, this document re-uses terminology from [I-D.ietf-teas-ietf-network-slice-definition], [I-D.ietf-teas-enhanced-vpn] and [I-D.ietf-teas-ietf-network-slice-framework]. The terms defined below are give context and meaning for use in this document only and do not force wider applicability. As other work matures, it is hoped that the terminology will converge.

**Service Provider:** A server network or collection of server networks. The persons or organization responsible for operating such networks.

**Consumer:** As defined in [I-D.ietf-teas-ietf-network-slice-definition], a consumer is the component or entity that requests and uses a network slice. This may be any application, client network, or customer of a service

provider. In the ACTN framework [RFC8453] the consumer of a network service is termed a 'customer' because it will often be the case that a VPN consumer is a customer of the operator of the core network that delivers the service. In the context of a network slice, the consumer may well be a customer, but might also be a client network of the service provider (which could also be an internal organization of the service provider), or an application that engineers traffic in the network.

**Service Functions (SFs):** Components that provide specific functions within a network. SFs are often combined in a specific sequence called a service function chain to deliver services [RFC7665].

**Resource:** Any feature including connectivity, bufferage, compute, storage, and content delivery that forms part of or can be accessed through a network. Resources may be shared between users, applications, and clients, or they may be dedicated for use by a unique consumer.

**Infrastructure Resources:** The hardware and software for hosting and connecting SFs. These resources may include computing hardware, storage capacity, network resources (e.g., links and switching/routing devices enabling network connectivity), and physical assets for radio access.

**Service Level Agreement (SLA):** Per [I-D.ietf-teas-ietf-network-slice-definition], an SLA is an explicit or implicit contract between the consumer of a network slice and the provider of the slice. The SLA is expressed in terms of a set of Service Level Objectives (SLOs) and may include commercial terms as well as the consequences of violating the SLOs. The SLA describes the quality with which features and functions are to be delivered. It may include measures of bandwidth, latency, and jitter; the types of service (such as firewalls or billing) to be provided; the location, nature, and quantities of services (such as the amount and location of compute resources and the accelerators required).

**Network Slice Service:** An agreement between a consumer and a service provider to deliver network resources according to a specific service level agreement.

## 2. Requirements for Network Slicing

According to [I-D.ietf-teas-ietf-network-slice-framework] the consumer expresses requirements for a particular IETF network slice by specifying what is required rather than how the requirement is to be fulfilled. That is, the IETF network slice consumer's view of a IETF network slice is an abstract one.

The concept of network slicing is a key capability to serve consumers with a wide variety of different service needs expressed as SLOs in term of latency, reliability, capacity, and service function specific capabilities.

This section outlines the key capabilities required to realize network slicing in a TE-enabled IETF technology network.

### 2.1. Resource Slicing

Network resources need to be allocated and dedicated for use by a specific network slice, or they may be shared among multiple slices. This allows a flexible approach that can deliver a range of services by partitioning (that is, slicing) the available network resources to make them available to meet the consumer's SLA.

### 2.2. Network Virtualization

Network virtualization enables the creation of multiple virtual networks that are operationally decoupled from the underlying physical network, and are run on top of it. Slicing enables the creation of virtual networks as consumer services.

### 2.3. Service Isolation

A consumer may request, through their SLA, that changes to the other services delivered by the service provider do not have any negative impact on the delivery of the service. This quality is referred to as "isolation" [I-D.ietf-teas-ietf-network-slice-definition] [I-D.ietf-teas-enhanced-vpn].

Delivery of such service isolation may be achieved in the underlying network by various forms of resource partitioning ranging from dedicated allocation of resources for a specific slice, to sharing or resources with safeguards.

Although multiple network slices may utilize resources from a single underlying network, isolation should be understood in terms of the following three categorisations.

- o Performance isolation requires that service delivery for one network slice does not adversely impact congestion or performance levels of other slices.
- o Security isolation means that attacks or faults occurring in one slice do not impact on other slices. Moreover, the security functions supporting each slice must operate independently so that an attack or misconfiguration of security in one slice will not

prevent proper security function in the other slices. Further, privacy concerns require that traffic from one slice is not delivered to an end point in another slice, and that it should not be possible to determine the nature or characteristics of a slice from any external point.

- o Management isolation means that each slice must be independently viewed, utilized, and managed as a separate network. Furthermore, it should be possible to prevent the operator of one slice from being able to control, view, or detect any aspect of any other network slice.

#### 2.4. Control and Orchestration

Orchestration combines and coordinates multiple control methods to provide a single mechanism to operate one or more networks to deliver services. In a network slicing environment, an orchestrator is needed to coordinate disparate processes and resources for creating, managing, and deploying the network slicing service. Two aspects of orchestration are required:

- o Multi-domain Orchestration: Managing connectivity to set up a network slice across multiple administrative domains.
- o End-to-end Orchestration: Combining resources for an end-to-end service (e.g., underlay connectivity with firewalling, and guaranteed bandwidth with minimum delay).

### 3. Abstraction and Control of Traffic Engineered (TE) Networks (ACTN)

ACTN facilitates end-to-end connectivity and provide virtual connectivity services (such as virtual links and virtual networks) to the user. The ACTN framework [RFC8453] introduces three functional components and two interfaces:

- o Customer Network Controller (CNC)
- o Multi-domain Service Coordinator (MDSC)
- o Provisioning Network Controller (PNC)
- o CNC-MDSC Interface (CMI)
- o MDSC-PNC Interface (MPI)

RFC 8453 also highlights how:

- o Abstraction of the underlying network resources is provided to higher-layer applications and consumers.
- o Virtualization is achieved by selecting resources according to criteria derived from the details and requirements of the consumer, application, or service.
- o Creation of a virtualized environment is performed to allow operators to view and control multi-domain networks as a single virtualized network.
- o A network is presented to a consumer as a single virtual network via open and programmable interfaces.

The ACTN managed infrastructure consists of traffic engineered network resources. The concept of traffic engineering is broad: it describes the planning and operation of networks using a method of reserving and partitioning of network resources in order to facilitate traffic delivery across a network (see [I-D.ietf-teas-rfc3272bis] for more details). In the context of ACTN, traffic engineering network resources may include:

- o Statistical packet bandwidth.
- o Physical forwarding plane sources, such as wavelengths and time slots.
- o Forwarding and cross-connect capabilities.

The ACTN network is "sliced" with consumers each being given a different partial and abstracted topology view of the physical underlay network.

### 3.1. ACTN Virtual Network as a Network Slice

To support multiple consumers, each with its own view of and control of a virtual network constructed using a server network, a service provider needs to partition the server network resources to create network slices assigned to each consumer.

An ACTN Virtual Network (VN) is a consumer view of a slice of the ACTN-managed infrastructure. It is a network slice that is presented to the consumer by the ACTN provider as a set of abstracted resources. See [I-D.ietf-teas-actn-vn-yang] for a detailed description of ACTN VNs and an overview of how various different types of YANG model are applicable to the ACTN framework.

Depending on the agreement between consumer and provider, various VN operations are possible:

- o Network Slice Creation: A VN could be pre-configured and created through static configuration or through dynamic request and negotiation between consumer and service provider. The VN must meet the network slice requirements specified in the SLA to satisfy the consumer's objectives.
- o Network Slice Operations: The VN may be modified and deleted based on consumer requests. The consumer can further act upon the VN to manage the consumer's traffic flows across the network slice.
- o Network Slice View: The VN topology is viewed from the consumer's perspective. This may be the entire VN topology or a collection of tunnels that are expressed as consumer end points, access links, intra domain paths and inter-domain links.

[RFC8454] describes a set of functional primitives that support these different ACTN VN operations.

### 3.2. ACTN Virtual Network for Network Slice Aggregation

Scaling considerations for IETF network slicing are an important consideration. If the service provider must manage and maintain network state for every network slice then this will quickly limit the number of customer services that can be supported.

The importance of network slice aggregation is discussed in [I-D.ietf-teas-enhanced-vpn] and further in [I-D.dong-teas-enhanced-vpn-vtn-scalability]. That work notes the importance of aggregating network slices into groups of similar slices before realizing those aggregates in the network.

The same consideration applies to ACTN VNs. But fortunately, ACTN VNs may be arranged hierarchically by recursing the MDSCs so that one VN is realised over another VN. This allows the VNs presented to the customer to be aggregated before they are instantiated in the physical network.

### 3.3. Management Components for ACTN and Network Slicing

The ACTN management components (CNC, MDSC, and PNC) and interfaces (CMI and MPI) are introduced in Section 3 and described in detail in [RFC8453]. The management components for network slicing are described in [I-D.ietf-teas-ietf-network-slice-framework] and are known as the consumer orchestration system, the IETF network slice controller (NSC), and the network controller. The network slicing

management components are separated by the network slice controller northbound interface (NSC NBI) and the network slice controller southbound interface (NSC SBI).

[I-D.ietf-teas-ietf-network-slice-framework] describes the mapping between network slicing management components and ACTN management components. This is presented visually in Figure 1 and provides a useful reference for understanding the material in Section 3.4 and Section 4.

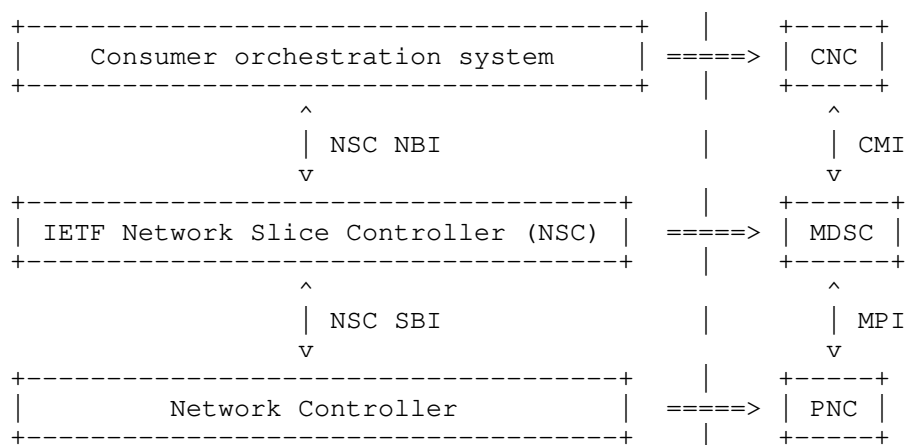


Figure 1: Mapping Between IETF Network Slice and ACTN Components

### 3.4. Examples of ACTN Delivering Types of Network Slices

The examples that follow build on the ACTN framework to provide control, management, and orchestration for the network slice life-cycle. These network slices utilize common physical infrastructure, and meet specific service-level requirements.

Three examples are shown. Each uses ACTN to achieve a different network slicing scenario. All three scenarios can be scaled up in capacity or be subject to topology changes as well as changes of consumer requirements.

#### 3.4.1. ACTN Used for Virtual Private Line

In the example shown in Figure 2, ACTN provides virtual connections between multiple consumer locations (sites accessed through Customer Edge nodes - CEs). The service is requested by the consumer (via

CNC-A) and delivered as a Virtual Private Line (VPL) service. The benefits of this model include:

- o Automated: the service set-up and operation is managed by the network provider.
- o Virtual: the private line connectivity is provided from Site A to Site C (VPL1) and from Site B to Site C (VPL2) across the ACTN-managed physical network.
- o Agile: on-demand adjustments to the connectivity and bandwidth are available according to the consumer's requests.



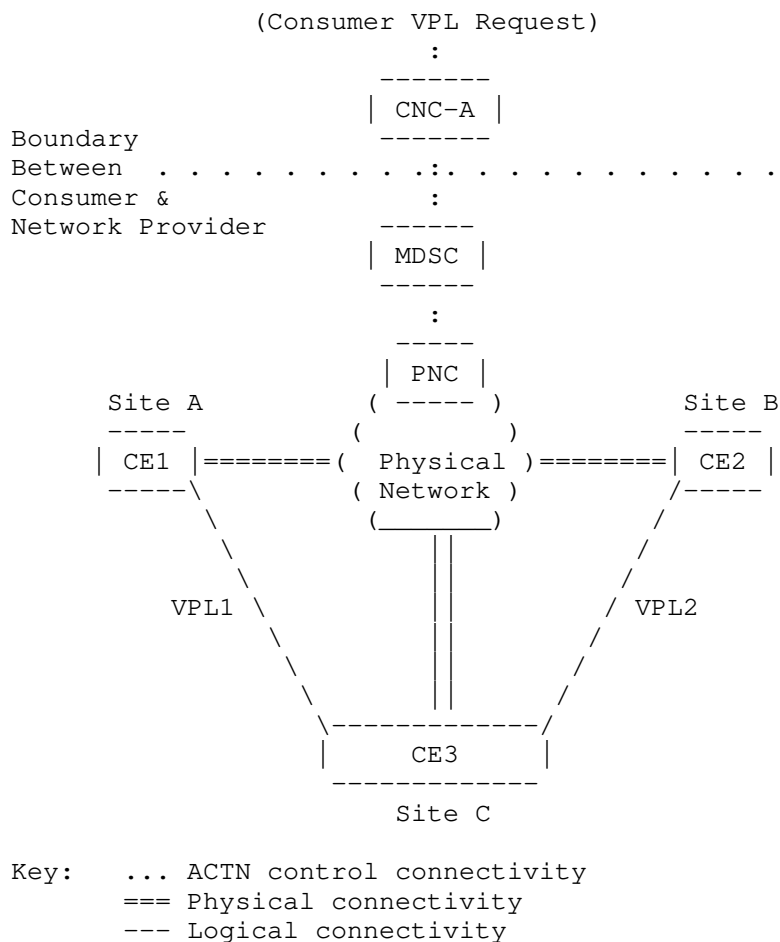


Figure 2: Virtual Private Line Model

### 3.4.2. ACTN Used for VPN Delivery Model

In the example shown in Figure 3, ACTN provides VPN connectivity between two sites across three physical networks. The requirements for the VPN are expressed by the users of the two sites who are the consumers. Their requests are directed to the CNC, and the CNC interacts with the network provider's MDSC. The benefits of this model include:

- o Provides edge-to-edge VPN multi-access connectivity.

- o Most of the function is managed by the network provider, with some flexibility delegated to the consumer-managed CNC.

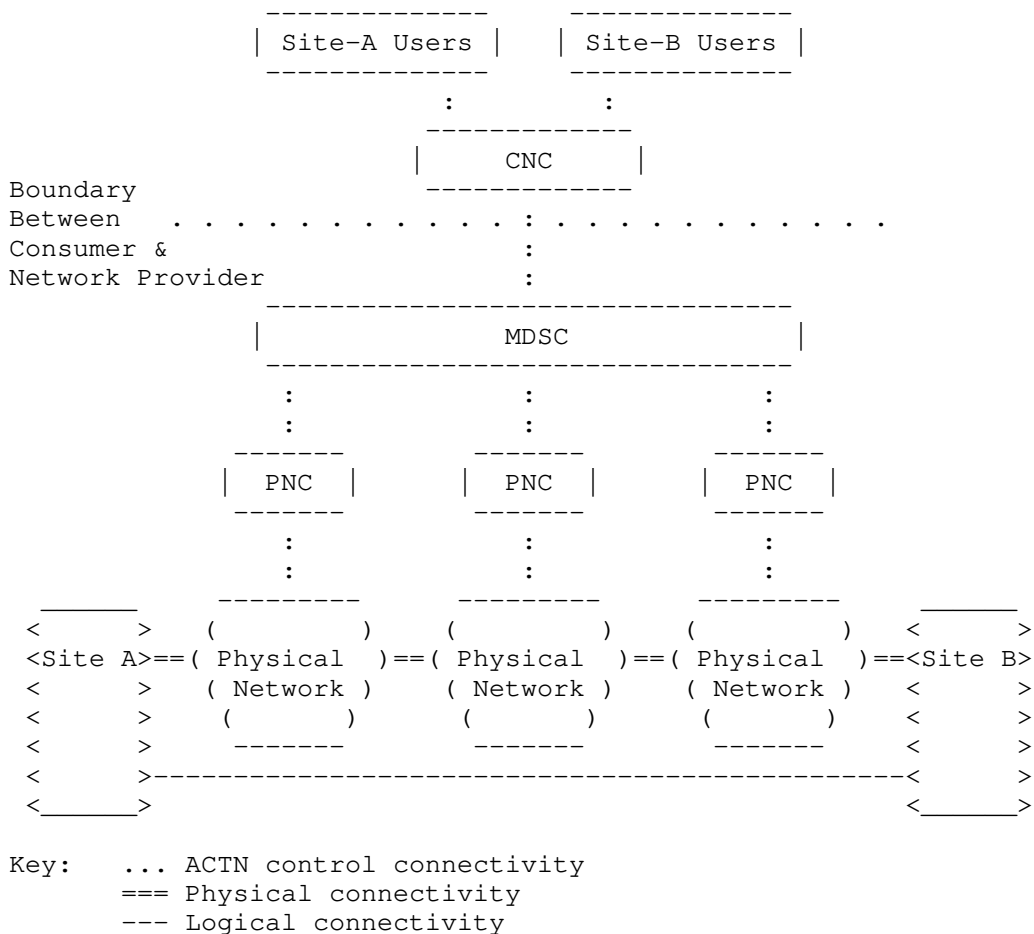


Figure 3: VPN Model

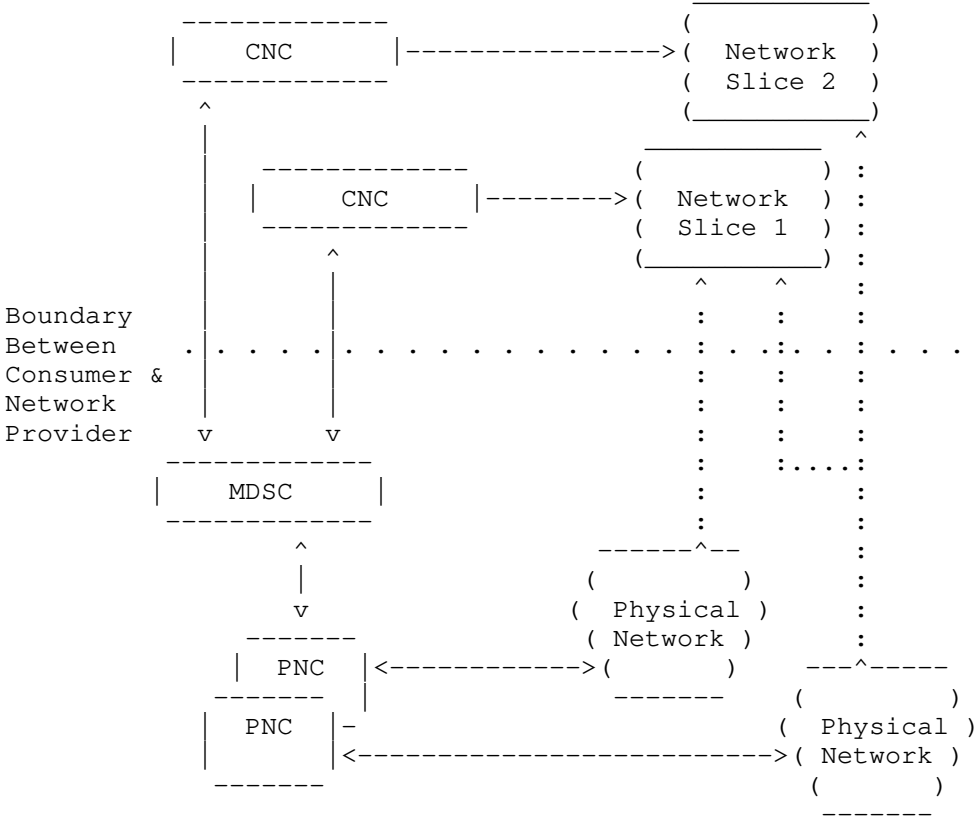
### 3.4.3. ACTN Used to Deliver a Virtual Consumer Network

In the example shown in Figure 4, ACTN provides a virtual network to the consumer. This virtual network is managed by the consumer. The figure shows two virtual networks (Network Slice 1 and Network Slice 2) each created for a different consumer under the care of a different CNC. There are two physical networks controlled by separate PNCs. Network Slice 2 is built using resources from just

one physical network, while Network Slice 1 is constructed from resources from both physical networks.

The benefits of this model include:

- o The MDSC provides the topology to the consumer so that the consumer can control their network slice to fit their needs.
- o Applications can interact with their assigned network slices directly. The consumer may implement their own network control methods and traffic prioritization, and manage their own addressing schemes.
- o Consumers may further slice their virtual networks so that this becomes a recursive model.
- o Service isolation can be provided through selection of physical networking resources through a combination of efforts of the MSDC and PNC.
- o The network slice may include nodes with specific capabilities. These can be delivered as Physical Network Functions (PNFs) or Virtual Network Functions (VNFs).



Key: --- ACTN control connection  
... Virtualization/abstraction through slicing

Figure 4: Network Slicing

4. YANG Models

4.1. Network Slice Service Mapping from TE to ACTN VN Models

The role of the TE-service mapping model [I-D.ietf-teas-te-service-mapping-yang] is to create a binding relationship across a Layer 3 Service Model (L3SM) [RFC8299], Layer 2 Service Model (L2SM) [RFC8466], and TE Tunnel model [I-D.ietf-teas-yang-te], via the generic ACTN Virtual Network (VN) model [I-D.ietf-teas-actn-vn-yang].

The ACTN VN model is a generic virtual network service model that allows consumers to specify a VN (i.e., network slice) that meets the consumer's service objectives with various constraints on how the service is delivered.

The TE-service mapping model [I-D.ietf-teas-te-service-mapping-yang] is used to bind the L3SM with TE-specific parameters. This binding facilitates seamless service operation and enables visibility of the underlay TE network. The TE-service model developed in that document can also be extended to support other services including L2SM, and the Layer 1 Connectivity Service Model (L1CSM) [I-D.ietf-ccamp-llcsm-yang] L1CSM network service models.

Figure 5 shows the relationship between the models discussed above.

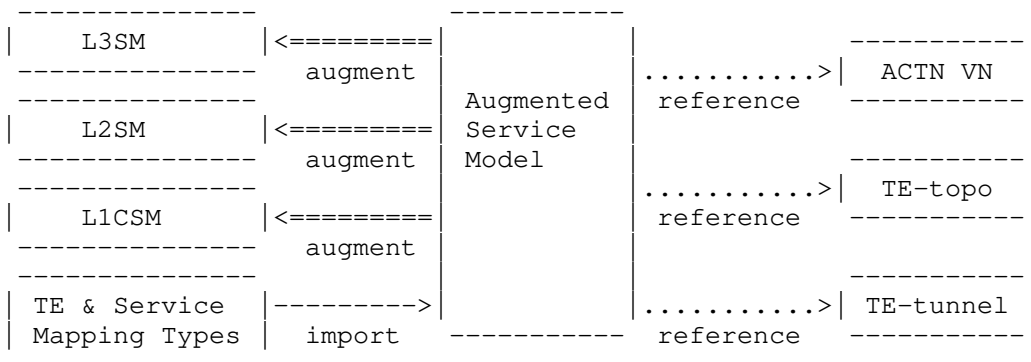


Figure 5: TE-Service Mapping

#### 4.2. Interfaces and Yang Models

Figure 6 shows the three ACTN components and two ACTN interfaces as listed in Section 3. The figure also shows the Southbound Interface (SBI) between the PNC and the devices in the physical network. That interface might be used to install state on every device in the network, or might instruct a "head-end" node if a control plane is used within the physical network. In the context of [RFC8309], the SBI uses one or more device configuration models.

The figure also shows the Network Slice Service Interface. This interface allows a consumer of a service to make requests for delivery of the service, and it facilitates the consumer modifying and monitoring the service. In the context of [RFC8309], this

"northbound interface (NBI)" is a customer service interface and uses a service model.

When an ACTN system is used to manage the delivery of network slices, a network slice resource model is needed. This model will be used for instantiation, operation, and monitoring of network and function resource slices. The YANG model defined in [I-D.wd-teas-transport-slice-yang] provides a suitable basis for requesting, controlling, and deleting, network slices.

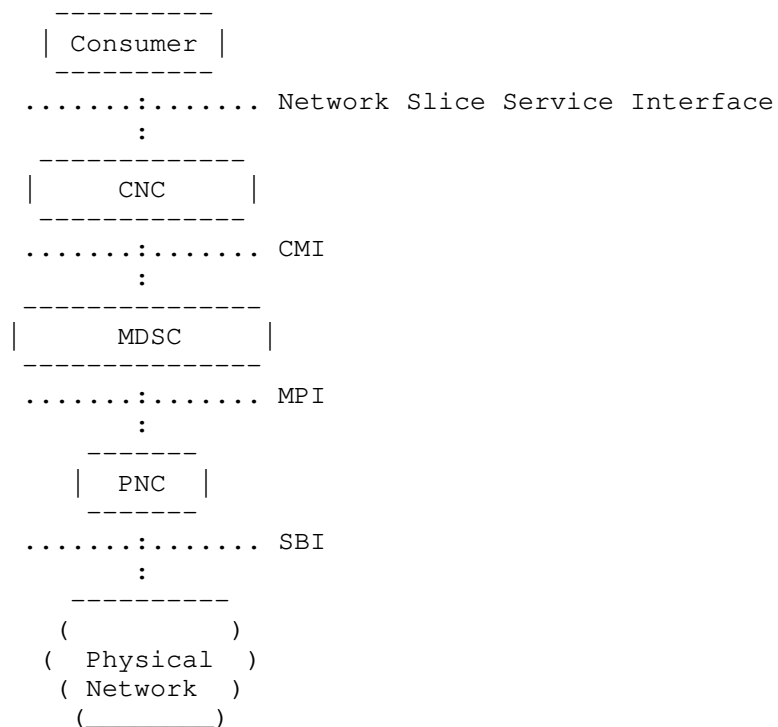


Figure 6: The Yang Interfaces in Context

#### 4.3. ACTN VN Telemetry

The ACTN VN KPI telemetry model [I-D.ietf-teas-actn-pm-telemetry-autonomics] provides a way for a consumer to define performance monitoring relevant for its VN/network slice via the NETCONF subscription mechanisms [RFC8639], [RFC8640], or using the equivalent mechanisms in RESTCONF [RFC8641], [RFC8650].

Key characteristics of [I-D.ietf-teas-actn-pm-telemetry-autonomics] include:

- o An ability to provide scalable VN-level telemetry aggregation based on consumer subscription model for key performance parameters defined by the consumer.
- o An ability to facilitate proactive re-optimization and reconfiguration of VNs/network slices based on network autonomic traffic engineering scaling configuration mechanism.

## 5. IANA Considerations

This document makes no requests for action by IANA.

## 6. Security Considerations

Network slicing involves the control of network resources in order to meet the service requirements of consumers. In some deployment models, the consumer is able to directly request modification in the behaviour of resources owned and operated by a service provider. Such changes could significantly affect the service provider's ability to provide services to other consumers. Furthermore, the resources allocated for or consumed by a consumer will normally be billable by the service provider.

Therefore, it is crucial that the mechanisms used in any network slicing system allow for authentication of requests, security of those requests, and tracking of resource allocations.

It should also be noted that while the partitioning or slicing of resources is virtual, as mentioned in Section 2.3 the consumers expect and require that there is no risk of leakage of data from one slice to another, no transfer of knowledge of the structure or even existence of other slices, and that changes to one slice (under the control of one consumer) should not have detrimental effects on the operation of other slices (whether under control of different or the same consumers) beyond the limits allowed within the SLA. Thus, slices are assumed to be private and to provide the appearance of genuine physical connectivity.

Some service providers may offer secure network slices as a service. Such services may claim to include edge-to-edge encryption for the consumer's traffic. However, a consumer should take full responsibility for the privacy and integrity of their traffic and should carefully consider using their own edge-to-edge encryption.

ACTN operates using the NETCONF [RFC6241] or RESTCONF [RFC8040] protocols and assumes the security characteristics of those protocols. Deployment models for ACTN should fully explore the authentication and other security aspects before networks start to carry live traffic.

## 7. Acknowledgements

Thanks to Qin Wu, Andy Jones, Ramon Casellas, Gert Grammel, and Kiran Makhijani for their insight and useful discussions about network slicing.

## 8. Contributors

The following people contributed text to this document.

Young Lee  
Email: [younglee.tx@gmail.com](mailto:younglee.tx@gmail.com)

Mohamed Boucadair  
Email: [mohamed.boucadair@orange.com](mailto:mohamed.boucadair@orange.com)

Sergio Belotti  
Email: [sergio.belotti@nokia.com](mailto:sergio.belotti@nokia.com)

Daniele Ceccarelli  
Email: [daniele.ceccarelli@ericsson.com](mailto:daniele.ceccarelli@ericsson.com)

## 9. Informative References

[I-D.dong-teas-enhanced-vpn-vtn-scalability]  
Dong, J., Li, Z., Qin, F., and G. Yang, "Scalability Considerations for Enhanced VPN (VPN+)", draft-dong-teas-enhanced-vpn-vtn-scalability-01 (work in progress), November 2020.

[I-D.ietf-ccamp-llcsm-yang]  
Lee, Y., Lee, K., Zheng, H., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-llcsm-yang-13 (work in progress), November 2020.



- [I-D.ietf-teas-actn-pm-telemetry-autonomics]  
Lee, Y., Dhody, D., Karunanithi, S., Vilata, R., King, D.,  
and D. Ceccarelli, "YANG models for VN/TE Performance  
Monitoring Telemetry and Scaling Intent Autonomics",  
draft-ietf-teas-actn-pm-telemetry-autonomics-04 (work in  
progress), November 2020.
- [I-D.ietf-teas-actn-vn-yang]  
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B.  
Yoon, "A YANG Data Model for VN Operation", draft-ietf-  
teas-actn-vn-yang-10 (work in progress), November 2020.
- [I-D.ietf-teas-enhanced-vpn]  
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A  
Framework for Enhanced Virtual Private Networks (VPN+)  
Service", draft-ietf-teas-enhanced-vpn-06 (work in  
progress), July 2020.
- [I-D.ietf-teas-ietf-network-slice-definition]  
Rokui, R., Homma, S., Makhiyani, K., Contreras, L., and J.  
Tantsura, "Definition of IETF Network Slices", draft-ietf-  
teas-ietf-network-slice-definition-00 (work in progress),  
January 2021.
- [I-D.ietf-teas-ietf-network-slice-framework]  
Gray, E. and J. Drake, "Framework for IETF Network  
Slices", draft-ietf-teas-ietf-network-slice-framework-00  
(work in progress), March 2021.
- [I-D.ietf-teas-rfc3272bis]  
Farrel, A., "Overview and Principles of Internet Traffic  
Engineering", draft-ietf-teas-rfc3272bis-10 (work in  
progress), December 2020.
- [I-D.ietf-teas-te-service-mapping-yang]  
Lee, Y., Dhody, D., Fioccola, G., WU, Q., Ceccarelli, D.,  
and J. Tantsura, "Traffic Engineering (TE) and Service  
Mapping Yang Model", draft-ietf-teas-te-service-mapping-  
yang-05 (work in progress), November 2020.
- [I-D.ietf-teas-yang-te]  
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin,  
"A YANG Data Model for Traffic Engineering Tunnels, Label  
Switched Paths and Interfaces", draft-ietf-teas-yang-te-25  
(work in progress), July 2020.

- [I-D.wd-teas-transport-slice-yang]  
Bo, W., Dhody, D., Han, L., and R. Rokui, "A Yang Data Model for Transport Slice NBI", draft-wd-teas-transport-slice-yang-02 (work in progress), July 2020.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.
- [RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8454] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., and B. Yoon, "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC 8454, DOI 10.17487/RFC8454, September 2018, <<https://www.rfc-editor.org/info/rfc8454>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.
- [RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <<https://www.rfc-editor.org/info/rfc8639>>.

- [RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", RFC 8640, DOI 10.17487/RFC8640, September 2019, <<https://www.rfc-editor.org/info/rfc8640>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.
- [RFC8650] Voit, E., Rahman, R., Nilsen-Nygaard, E., Clemm, A., and A. Bierman, "Dynamic Subscription to YANG Events and Datastores over RESTCONF", RFC 8650, DOI 10.17487/RFC8650, November 2019, <<https://www.rfc-editor.org/info/rfc8650>>.

#### Authors' Addresses

Daniel King  
Old Dog Consulting  
  
Email: [daniel@olddog.co.uk](mailto:daniel@olddog.co.uk)

John Drake  
Juniper Networks  
  
Email: [jdrake@juniper.net](mailto:jdrake@juniper.net)

Haomian Zheng  
Huawei Technologies  
  
Email: [zhenghaomian@huawei.com](mailto:zhenghaomian@huawei.com)

Adrian Farrel  
Old Dog Consulting  
  
Email: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 10, 2022

X. Liu  
Volta Networks  
J. Tantsura  
Microsoft  
I. Bryskin  
Individual  
L. Contreras  
Telefonica  
Q. Wu  
Huawei  
S. Belotti  
R. Rokui  
Nokia  
July 9, 2021

IETF Network Slice YANG Data Model  
draft-liu-teas-transport-network-slice-yang-04

## Abstract

This document describes a YANG data model for managing and controlling IETF network slices, defined in [I-D.liu-teas-ietf-network-slices].

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 10, 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                                     | 2  |
| 1.1. Terminology . . . . .                                    | 3  |
| 1.2. Tree Diagrams . . . . .                                  | 3  |
| 2. Modeling Considerations . . . . .                          | 3  |
| 2.1. Relationships to Related Topology Models . . . . .       | 3  |
| 2.2. Network Slice with TE . . . . .                          | 4  |
| 2.3. ACTN for Network Slicing . . . . .                       | 5  |
| 3. Model Applicability . . . . .                              | 6  |
| 3.1. Network Slicing by Virtualization . . . . .              | 6  |
| 3.2. Network Slicing by TE Overlay . . . . .                  | 8  |
| 4. Model Tree Structure . . . . .                             | 10 |
| 5. YANG Module . . . . .                                      | 10 |
| 6. IANA Considerations . . . . .                              | 16 |
| 7. Security Considerations . . . . .                          | 17 |
| 8. Acknowledgements . . . . .                                 | 18 |
| 9. References . . . . .                                       | 18 |
| 9.1. Normative References . . . . .                           | 18 |
| 9.2. Informative References . . . . .                         | 20 |
| Appendix A. Data Tree for the Example in Section 3.1. . . . . | 22 |
| A.1. Native Topology . . . . .                                | 22 |
| A.2. Network Slice Blue . . . . .                             | 26 |
| Authors' Addresses . . . . .                                  | 32 |

## 1. Introduction

This document defines a YANG [RFC7950] data model for for representing, managing, and controlling IETF network slices, defined in [I-D.ietf-teas-ietf-network-slices]

The defined data model is an interface between clients and providers for configurations and state retrievals, so as to support network slicing as a service. Through this model, a client can learn the slicing capabilities and the available resources of the provider. A client can request or negotiate with a network slicing provider to create an instance. The client can incrementally update its requirements on individual topology elements in the slice instance,

and retrieve the operational states of these elements. With the help of other mechanisms and data models defined in IETF, the telemetry information can be published to the client.

The YANG data model in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

### 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC7950] and are not redefined here:

- o augment
- o data model
- o data node

### 1.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

## 2. Modeling Considerations

An IETF network slice is modeled as network topology defined in [RFC8345], with augmentations. A new network type "network-slice" is defined in this document. When a network topology data instance contains the network-slice network type, it represents an instance of an IETF network slice.

### 2.1. Relationships to Related Topology Models

There are several related YANG data models that have been defined in IETF. Some of these are:

Network Topology Model:  
Defined in [RFC8345].

OTN Topology Model:  
Defined in [I-D.ietf-ccamp-otn-topo-yang].

## L2 Topology Model:

Defined in [I-D.ietf-i2rs-yang-l2-network-topology].

### L3 Topology Model:

Defined in [RFC8346].

### TE Topology Model:

Defined in [RFC8795].

Figure 1 shows the relationships among these models. The box of dotted lines denotes the model defined in this document.

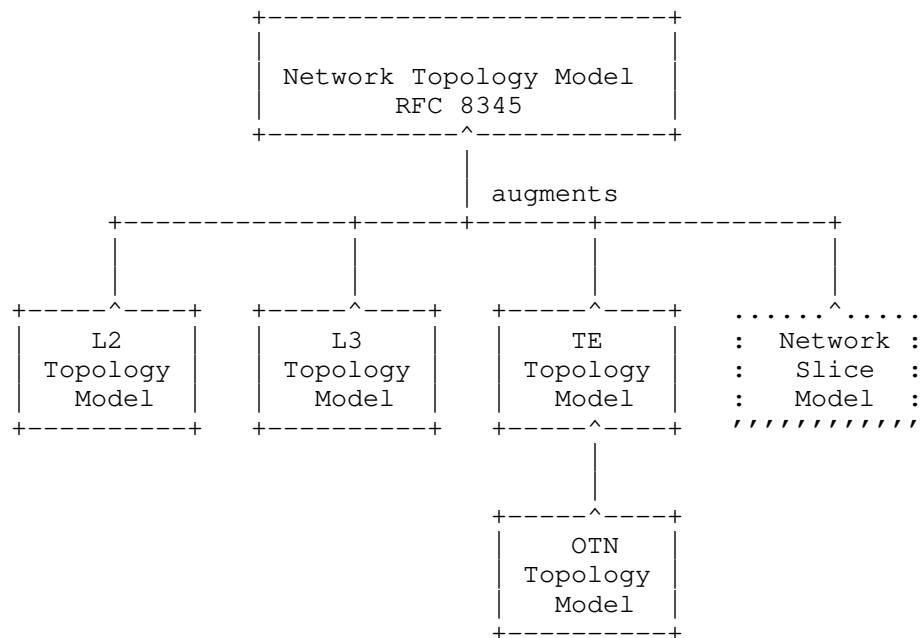


Figure 1: Model Relationships

## 2.2. Network Slice with TE

In many situations, an IETF network slice needs to have TE (Traffic Engineering) capabilities to achieve certain network characteristics. The TE Topology Model defined in [RFC8795] can be used to make an IETF network slice TE capable. To achieve this, an IETF network slice instance will be configured to have both "network-slice" and "te-topology" network types, taking advantage of the multiple

inheritance capability featured by the network topology model [RFC8345]. The following diagram shows their relations.

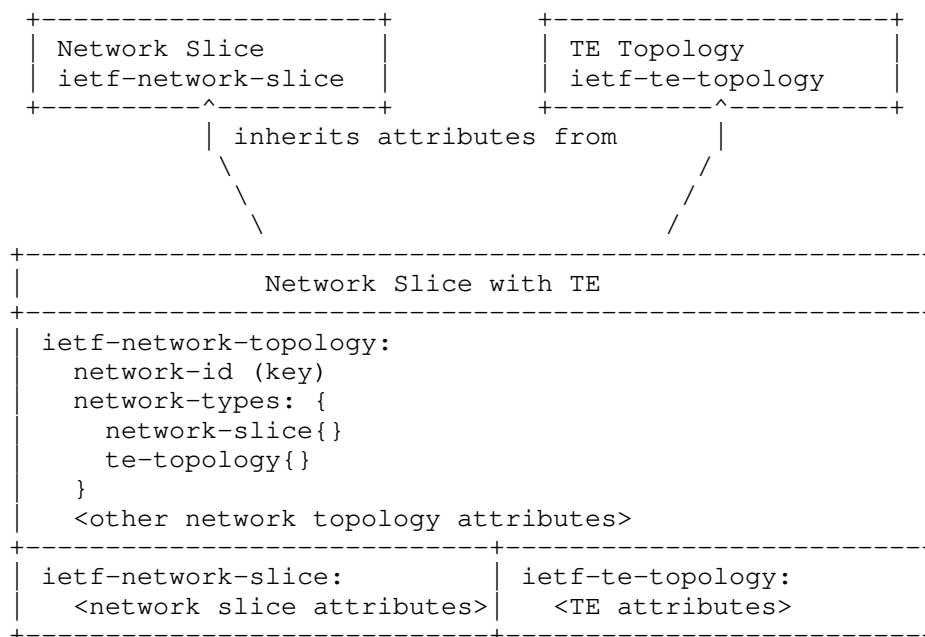


Figure 2: Network Slice with TE

This method can be applied to other types of network topology models too. For example, when a network topology instance is configured to have the types of "network-slice" defined in this document, "te-topology" defined in [RFC8795], and "l3-unicast-topology" defined in [RFC8346], this network topology instance becomes an IETF network slice instance that can perform layer 3 traffic engineering.

### 2.3. ACTN for Network Slicing

Since ACTN topology data models are based on the network topology model defined in [RFC8345], the augmentations defined in this document are effective augmentations to the ACTN topology data models, resulting in making the ACTN framework [RFC8453] and data models [I-D.ietf-teas-actn-yang] capable of slicing networks with the required network characteristics.



### 3. Model Applicability

There are many technologies to achieve network slicing. The data model defined in this document can be applied to a wide ranges of cases. This section describes how this data model is applied to a few cases.

#### 3.1. Network Slicing by Virtualization

In the case shown in Figure 3, node virtualization is used to separate and allocate resources in physical devices. Two virtual routers VR1 and VR2 are created over physical router R1. Each of the virtual routers takes a portion of the resources such as ports and memory in the physical router. Depending on the requirements and the implementations, they may share certain resources such as processors, ASICs, and switch fabric.

As an example, Appendix A. shows the JSON encoded data instances of the native topology and the customized topology for Network Slice Blue.

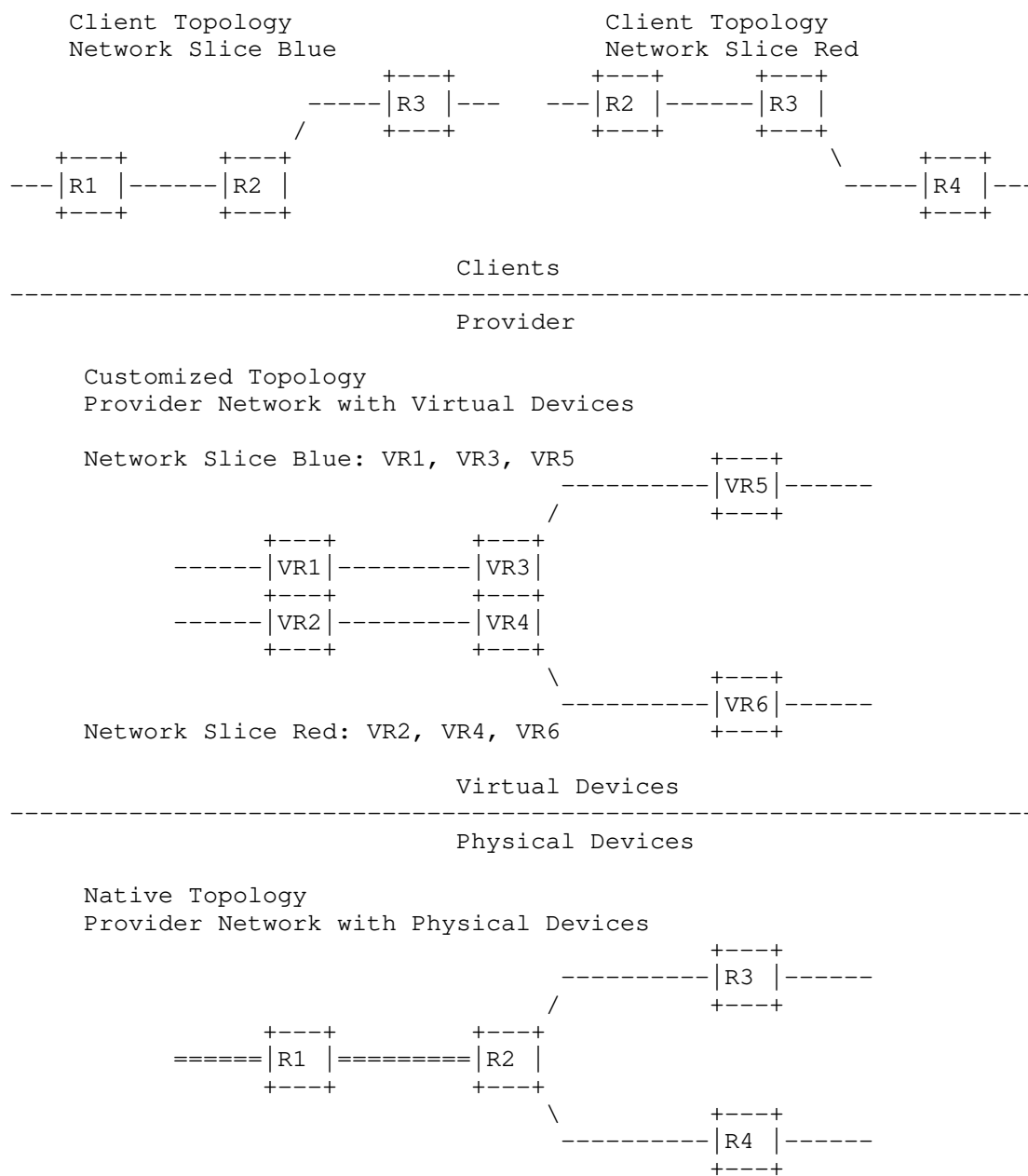


Figure 3: Network Slicing by Virtualization

### 3.2. Network Slicing by TE Overlay

Figure 4 shows a case where TE (Traffic Engineering) overlay is applied to achieve logically separated client IETF network slices. In the underlay TE capable network, TE tunnels are established to support the TE links in the overlay network. These links and tunnels maintain the characteristics required by the clients. The provider selects the proper logical nodes and links in the overlay network, assigns them to specific IETF network slices, and uses the data model defined in this document to send the results to the clients.

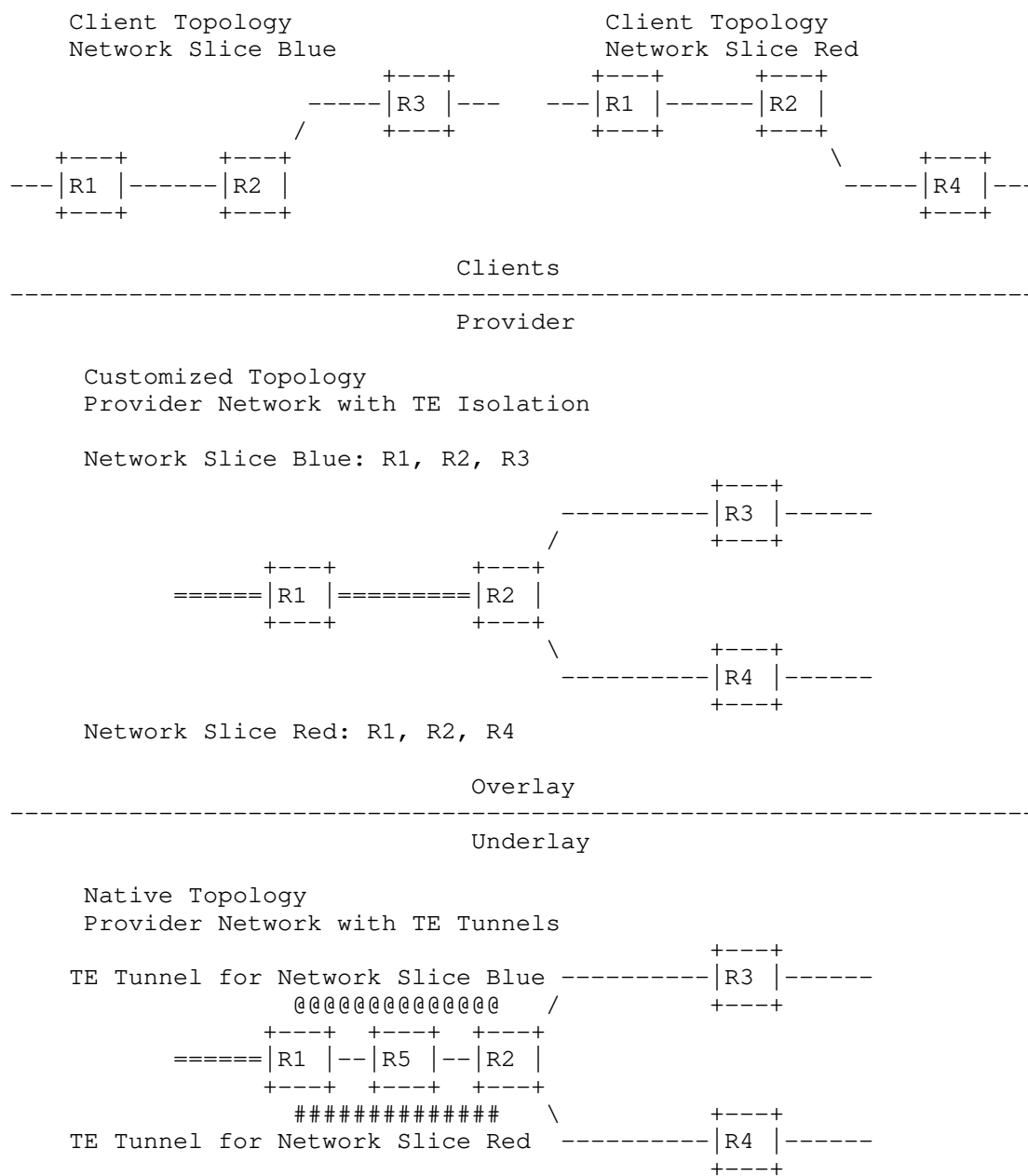


Figure 4: Network Slicing by TE Overlay

#### 4. Model Tree Structure

TODO - Complete IETF network slice attributes that are technology-agnostic and common to all use cases.

```
module: ietf-network-slice
  augment /nw:networks/nw:network/nw:network-types:
    +--rw network-slice!
  augment /nw:networks/nw:network:
    +--rw network-slice
      +--rw optimization-criterion?  identityref
      +--rw delay-tolerance?         boolean
      +--rw periodicity*             uint64
      +--rw isolation-level?         identityref
  augment /nw:networks/nw:network/nw:node:
    +--rw network-slice
      +--rw isolation-level?         identityref
      +--rw compute-node-id?        string
      +--rw storage-id?             string
  augment /nw:networks/nw:network/nt:link:
    +--rw network-slice
      +--rw delay-tolerance?        boolean
      +--rw periodicity*            uint64
      +--rw isolation-level?        identityref
```

#### 5. YANG Module

This module references [RFC8345], [RFC8776], and [GSMA-NS-Template]

```
<CODE BEGINS> file "ietf-network-slice@2020-11-01.yang"
module ietf-network-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-slice";
  prefix "ns";

  import ietf-network {
    prefix "nw";
    reference "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-network-topology {
    prefix "nt";
    reference "RFC 8345: A YANG Data Model for Network Topologies";
  }
}
```

```
import ietf-te-types {  
  prefix "te-types";  
  reference  
    "RFC 8776: Traffic Engineering Common YANG Types";  
}  
  
organization  
  "IETF Traffic Engineering Architecture and Signaling (TEAS)  
  Working Group";  
  
contact  
  "WG Web:    <http://tools.ietf.org/wg/teas/>  
  WG List:    <mailto:teas@ietf.org>  
  
  Editor:     Xufeng Liu  
              <mailto:xufeng.liu.ietf@gmail.com>  
  
  Editor:     Jeff Tantsura  
              <mailto:jefftant.ietf@gmail.com>  
  
  Editor:     Igor Bryskin  
              <mailto:i_bryskin@yahoo.com>  
  
  Editor:     Luis Miguel Contreras Murillo  
              <mailto:luismiguel.contrerasmurillo@telefonica.com>  
  
  Editor:     Qin Wu  
              <mailto:bill.wu@huawei.com>  
  
  Editor:     Sergio Belotti  
              <mailto:sergio.belotti@nokia.com>  
  
  Editor:     Reza Rokui  
              <mailto:reza.rokui@nokia.com>  
  ";  
  
description  
  "YANG data model for representing and managing network  
  slices.  
  
  Copyright (c) 2020 IETF Trust and the persons identified as  
  authors of the code. All rights reserved.  
  
  Redistribution and use in source and binary forms, with or  
  without modification, is permitted pursuant to, and subject to  
  the license terms contained in, the Simplified BSD License set  
  forth in Section 4.c of the IETF Trust's Legal Provisions  
  Relating to IETF Documents
```

(<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2020-11-01 {
  description "Initial revision";
  reference
    "RFC XXXX: YANG Data Model for Network Slices";
}

/*
 * Identities
 */
identity isolation-level {
  description
    "Base identity for the isolation-level.";
  reference
    "GSMA-NS-Template: Generic Network Slice Template,
    Version 3.0.";
}
identity no-isolation {
  base isolation-level;
  description
    "Network slices are not separated.";
}
identity physical-isolation {
  base isolation-level;
  description
    "Network slices are physically separated (e.g. different rack,
    different hardware, different location, etc.).";
}
identity logical-isolation {
  base isolation-level;
  description
    "Network slices are logically separated.";
}
identity process-isolation {
  base physical-isolation;
  description
    "Process and threads isolation.";
}
identity physical-memory-isolation {
  base physical-isolation;
  description
    "Process and threads isolation.";
}
identity physical-network-isolation {
```

```
    base physical-isolation;
    description
      "Process and threads isolation.";
  }
  identity virtual-resource-isolation {
    base logical-isolation;
    description
      "A network slice has access to specific range of resources
       that do not overlap with other network slices
       (e.g. VM isolation).";
  }
  identity network-functions-isolation {
    base logical-isolation;
    description
      "NF (Network Function) is dedicated to the network slice, but
       virtual resources are shared.";
  }
  identity service-isolation {
    base logical-isolation;
    description
      "NSC data are isolated from other NSCs, but virtual
       resources and NFs are shared.";
  }
}

/*
 * Groupings
 */
grouping network-slice-topology-attributes {
  description "Network Slice topology scope attributes.";
  container network-slice {
    description
      "Containing Network Slice attributes.";
    leaf optimization-criterion {
      type identityref {
        base te-types:objective-function-type;
      }
      description
        "Optimization criterion applied to this topology.";
    }
    leaf delay-tolerance {
      type boolean;
      description
        "'true' if is not too critical how long it takes to deliver
         the amount of data.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
         Version 3.0.";
    }
  }
}
```



```
    leaf-list periodicity {
      type uint64;
      units seconds;
      description
        "A list of periodicities supported by the network slice.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
    }
    leaf isolation-level {
      type identityref {
        base isolation-level;
      }
      description
        "A network slice instance may be fully or partly, logically
        and/or physically, isolated from another network slice
        instance. This attribute describes different types of
        isolation:";
    }
  } // network-slice
} // network-slice-topology-attributes

grouping network-slice-node-attributes {
  description "Network Slice node scope attributes.";
  container network-slice {
    description
      "Containing Network Slice attributes.";
    leaf isolation-level {
      type identityref {
        base isolation-level;
      }
      description
        "A network slice instance may be fully or partly, logically
        and/or physically, isolated from another network slice
        instance. This attribute describes different types of
        isolation:";
    }
    leaf compute-node-id {
      type string;
      description
        "Reference to a compute node instance specified in
        a data model specifying the computing resources.";
    }
    leaf storage-id {
      type string;
      description
        "Reference to a storage instance specified in
        a data model specifying the storage resources.";
    }
  }
}
```

```
    }
  } // network-slice
} // network-slice-node-attributes

grouping network-slice-link-attributes {
  description "Network Slice link scope attributes";
  container network-slice {
    description
      "Containing Network Slice attributes.";
    leaf delay-tolerance {
      type boolean;
      description
        "'true' if is not too critical how long it takes to deliver
        the amount of data.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
    }
    leaf-list periodicity {
      type uint64;
      units seconds;
      description
        "A list of periodicities supported by the network slice.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
    }
    leaf isolation-level {
      type identityref {
        base isolation-level;
      }
      description
        "A network slice instance may be fully or partly, logically
        and/or physically, isolated from another network slice
        instance. This attribute describes different types of
        isolation:";
    }
  } // network-slice
} // network-slice-link-attributes

/*
 * Data nodes
 */
augment "/nw:networks/nw:network/nw:network-types" {
  description
    "Defines the Network Slice topology type.";
  container network-slice {
    presence "Indicates Network Slice topology";
  }
}
```

```
        description
          "Its presence identifies the Network Slice type.";
      }
    }

    augment "/nw:networks/nw:network" {
      when "nw:network-types/ns:network-slice" {
        description "Augment only for Network Slice topology.";
      }
      description "Augment topology configuration and state.";
      uses network-slice-topology-attributes;
    }

    augment "/nw:networks/nw:network/nw:node" {
      when "../nw:network-types/ns:network-slice" {
        description "Augment only for Network Slice topology.";
      }
      description "Augment node configuration and state.";
      uses network-slice-node-attributes;
    }

    augment "/nw:networks/nw:network/nt:link" {
      when "../nw:network-types/ns:network-slice" {
        description "Augment only for Network Slice topology.";
      }
      description "Augment link configuration and state.";
      uses network-slice-link-attributes;
    }
  }
}
<CODE ENDS>
```

## 6. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```
-----
URI: urn:ietf:params:xml:ns:yang:ietf-network-slice
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
-----
```

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

```
-----
name:          ietf-l3-te-topology
namespace:     urn:ietf:params:xml:ns:yang:ietf-network-slice
prefix:        ns
reference:     RFC XXXX
-----
```

## 7. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/nw:networks/nw:network/nw:network-types/ns:network-slice
  This subtree specifies the network slice type. Modifying the
  configurations can make network slice type invalid and cause
  interruption to IETF network slices.
```

```
/nw:networks/nw:network/ns:network-slice
  This subtree specifies the topology-wide configurations.
  Modifying the configurations here can cause traffic
  characteristics changed in this IETF network slice and related
  networks.
```

```
/nw:networks/nw:network/nw:node/ns:network-slice
  This subtree specifies the configurations of the nodes in a IETF
  network slice. Modifying the configurations in this subtree can
```

change the traffic characteristics on this node and the related networks.

/nw:networks/nw:network/nt:link/ns:network-slice

This subtree specifies the configurations of the links in a IETF network slice. Modifying the configurations in this subtree can change the traffic characteristics on this link and the related networks.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/nw:networks/nw:network/nw:network-types/ns:network-slice

Unauthorized access to this subtree can disclose the network slice type.

/nw:networks/nw:network/ns:network-slice

Unauthorized access to this subtree can disclose the topology-wide states.

/nw:networks/nw:network/nw:node/ns:network-slice

Unauthorized access to this subtree can disclose the operational state information of the nodes in a IETF network slice.

/nw:networks/nw:network/nt:link/ns:network-slic

Unauthorized access to this subtree can disclose the operational state information of the links in a IETF network slice.

## 8. Acknowledgements

The TEAS Network Slicing Design Team (NSDT) members included Aijun Wang, Dong Jie, Eric Gray, Jari Arkko, Jeff Tantsura, John E Drake, Luis M. Contreras, Rakesh Gandhi, Ran Chen, Reza Rokui, Ricard Vilalta, Ron Bonica, Sergio Belotti, Tomonobu Niwa, Xuesong Geng, and Xufeng Liu.

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.

- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.
- [GSMA-NS-Template]  
GSM Association, "Generic Network Slice Template, Version 3.0", NG.116, May 2020.
- [I-D.ietf-teas-ietf-network-slices]  
Farrel, A., Gray, E., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", draft-ietf-teas-ietf-network-slices-00 (work in progress), April 2021.

## 9.2. Informative References

- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.

[I-D.ietf-ccamp-otn-topo-yang]

Zheng, H., Busi, I., Liu, X., Belotti, S., and O. G. D. Dios, "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang-12 (work in progress), February 2021.

[I-D.ietf-i2rs-yang-l2-network-topology]

Dong, J., Wei, X., Wu, Q., Boucadair, M., and A. Liu, "A YANG Data Model for Layer 2 Network Topologies", draft-ietf-i2rs-yang-l2-network-topology-18 (work in progress), September 2020.

[I-D.ietf-teas-actn-yang]

Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B. Y., and S. Belotti, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-yang-07 (work in progress), February 2021.



## Appendix A. Data Tree for the Example in Section 3.1.

## A.1. Native Topology

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network" for the native topology depicted in Figure 3.

```
{
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "example-native-topology",
        "network-types": {
        },
        "node": [
          {
            "node-id": "R1",
            "ietf-network-topology:termination-point": [
              {
                "tp-id": "1-0-1"
              },
              {
                "tp-id": "1-0-2"
              },
              {
                "tp-id": "1-2-1"
              },
              {
                "tp-id": "1-2-2"
              }
            ]
          },
          {
            "node-id": "R2",
            "ietf-network-topology:termination-point": [
              {
                "tp-id": "2-1-1"
              },
              {
                "tp-id": "2-1-2"
              },
              {
                "tp-id": "2-3-1"
              },
              {
                "tp-id": "2-4-1"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ]
  },
  {
    "node-id": "R3",
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "3-0-1"
      },
      {
        "tp-id": "3-2-1"
      }
    ]
  },
  {
    "node-id": "R4",
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "4-0-1"
      },
      {
        "tp-id": "4-2-1"
      }
    ]
  }
],
"ietf-network-topology:link": [
  {
    "link-id": "R1,1-0-1,,",
    "source": {
      "source-node": "R1",
      "source-tp": "1-0-1"
    }
  },
  {
    "link-id": ",,R1,1-0-1",
    "destination": {
      "dest-node": "R1",
      "dest-tp": "1-0-1"
    }
  },
  {
    "link-id": "R1,1-0-2,,",
    "source": {
      "source-node": "R1",
      "source-tp": "1-0-2"
    }
  },
  {

```

```
    "link-id":",,R1,1-0-2",
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-0-2"
    }
  },
  {
    "link-id":"R1,1-2-1,R2,2-1-1",
    "source": {
      "source-node":"R1",
      "source-tp":"1-2-1"
    },
    "destination": {
      "dest-node":"R2",
      "dest-tp":"2-1-1"
    }
  },
  {
    "link-id":"R2,2-1-1,R1,1-2-1",
    "source": {
      "source-node":"R2",
      "source-tp":"2-1-1"
    },
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-2-1"
    }
  },
  {
    "link-id":"R1,1-2-2,R2,2-1-2",
    "source": {
      "source-node":"R1",
      "source-tp":"1-2-2"
    },
    "destination": {
      "dest-node":"R2",
      "dest-tp":"2-1-2"
    }
  },
  {
    "link-id":"R2,2-1-2,R1,1-2-2",
    "source": {
      "source-node":"R2",
      "source-tp":"2-1-2"
    },
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-2-2"
    }
  }
}
```

```
    }  
  },  
  {  
    "link-id": "R2,2-3-1,R3,3-2-1",  
    "source": {  
      "source-node": "R2",  
      "source-tp": "2-3-1"  
    },  
    "destination": {  
      "dest-node": "R3",  
      "dest-tp": "3-2-1"  
    }  
  },  
  {  
    "link-id": "R3,3-2-1,R2,2-3-1",  
    "source": {  
      "source-node": "R3",  
      "source-tp": "3-2-1"  
    },  
    "destination": {  
      "dest-node": "R2",  
      "dest-tp": "2-3-1"  
    }  
  },  
  {  
    "link-id": "R2,2-4-1,R4,4-2-1",  
    "source": {  
      "source-node": "R2",  
      "source-tp": "2-4-1"  
    },  
    "destination": {  
      "dest-node": "R4",  
      "dest-tp": "4-2-1"  
    }  
  },  
  {  
    "link-id": "R4,4-2-1,R2,2-4-1",  
    "source": {  
      "source-node": "R4",  
      "source-tp": "4-2-1"  
    },  
    "destination": {  
      "dest-node": "R2",  
      "dest-tp": "2-4-1"  
    }  
  },  
  {  
    "link-id": "R3,3-0-1,,",
```

```

        "source": {
          "source-node": "R3",
          "source-tp": "3-0-1"
        }
      },
      {
        "link-id": ",,R3,3-0-1",
        "destination": {
          "dest-node": "R3",
          "dest-tp": "3-0-1"
        }
      },
      {
        "link-id": "R4,4-0-1,,",
        "source": {
          "source-node": "R4",
          "source-tp": "4-0-1"
        }
      },
      {
        "link-id": ",,R4,4-0-1",
        "destination": {
          "dest-node": "R4",
          "dest-tp": "4-0-1"
        }
      }
    ]
  }
}

```

#### A.2. Network Slice Blue

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network-slice" for the topology customized for Network Slice Blue depicted in Figure 3.

```

{
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "example-customized-blue-topology",
        "network-types": {
          "ietf-network-slice:network-slice": {

```

```
    }
  },
  "supporting-network": [
    {
      "network-ref": "example-native-topology"
    }
  ],
  "node": [
    {
      "node-id": "VR1",
      "supporting-node": [
        {
          "network-ref": "example-native-topology",
          "node-ref": "R1"
        }
      ],
      "ietf-network-slice:network-slice": {
        "isolation-level":
          "ietf-network-slice:physical-memory-isolation"
      },
      "ietf-network-topology:termination-point": [
        {
          "tp-id": "1-0-1"
        },
        {
          "tp-id": "1-3-1"
        }
      ]
    },
    {
      "node-id": "VR3",
      "supporting-node": [
        {
          "network-ref": "example-native-topology",
          "node-ref": "R2"
        }
      ],
      "ietf-network-slice:network-slice": {
        "isolation-level":
          "ietf-network-slice:physical-memory-isolation"
      },
      "ietf-network-topology:termination-point": [
        {
          "tp-id": "3-1-1"
        },
        {
          "tp-id": "3-5-1"
        }
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "node-id": "VR5",
    "supporting-node": [
      {
        "network-ref": "example-native-topology",
        "node-ref": "R3"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-memory-isolation"
    },
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "5-3-1"
      },
      {
        "tp-id": "5-0-1"
      }
    ]
  }
],
"ietf-network-topology:link": [
  {
    "link-id": "VR1,1-0-1,,",
    "source": {
      "source-node": "VR1",
      "source-tp": "1-0-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R1,1-0-1,,"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": ",,VR1,1-0-1",
    "destination": {
      "dest-node": "VR1",
      "dest-tp": "1-0-1"
    }
  },

```

```
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": ",,R1,1-0-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR1,1-3-1,VR3,3-1-1",
    "source": {
      "source-node": "VR1",
      "source-tp": "1-3-1"
    },
    "destination": {
      "dest-node": "VR3",
      "dest-tp": "3-1-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R1,1-2-1,R2,2-1-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR3,3-1-1,VR1,1-3-1",
    "source": {
      "source-node": "VR3",
      "source-tp": "3-1-1"
    },
    "destination": {
      "dest-node": "R1",
      "dest-tp": "1-3-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R2,2-1-1,R1,1-2-1"
      }
    ],
  },
```



```

    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR3,3-5-1,VR5,5-3-1",
    "source": {
      "source-node": "VR3",
      "source-tp": "3-5-1"
    },
    "destination": {
      "dest-node": "VR5",
      "dest-tp": "5-3-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R2,2-3-1,R3,3-2-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR5,5-3-1,VR3,3-5-1",
    "source": {
      "source-node": "VR5",
      "source-tp": "5-3-1"
    },
    "destination": {
      "dest-node": "VR3",
      "dest-tp": "3-5-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R3,3-2-1,R2,2-3-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  }
},
{

```

```

        "link-id": "VR5,5-0-1,,",
        "source": {
            "source-node": "VR5",
            "source-tp": "5-0-1"
        },
        "supporting-link": [
            {
                "network-ref": "example-native-topology",
                "link-ref": "R3,3-0-1,,",
            }
        ],
        "ietf-network-slice:network-slice": {
            "isolation-level":
                "ietf-network-slice:physical-network-isolation"
        }
    },
    {
        "link-id": ",,VR5,5-0-1",
        "destination": {
            "dest-node": "VR5",
            "dest-tp": "5-0-1"
        },
        "supporting-link": [
            {
                "network-ref": "example-native-topology",
                "link-ref": ",,R3,3-0-1"
            }
        ],
        "ietf-network-slice:network-slice": {
            "isolation-level":
                "ietf-network-slice:physical-network-isolation"
        }
    }
],
"ietf-network-slice:network-slice": {
    "optimization-criterion":
        "ietf-te-types:of-minimize-cost-path",
    "isolation-level":
        "ietf-network-slice:physical-isolation"
}
}
]
}
}

```

Authors' Addresses

Xufeng Liu  
Volta Networks

EMail: xufeng.liu.ietf@gmail.com

Jeff Tantsura  
Microsoft

EMail: jefftant.ietf@gmail.com

Igor Bryskin  
Individual

EMail: i\_bryskin@yahoo.com

Luis Miguel Contreras Murillo  
Telefonica

EMail: luismiguel.contrerasmurillo@telefonica.com

Qin Wu  
Huawei

EMail: bill.wu@huawei.com

Sergio Belotti  
Nokia

EMail: sergio.belotti@nokia.com

Reza Rokui  
Nokia  
Canada

EMail: reza.rokui@nokia.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 7, 2022

X. Liu  
IBM Corporation  
J. Tantsura  
Microsoft  
I. Bryskin  
Individual  
L. Contreras  
Telefonica  
Q. Wu  
Huawei  
S. Belotti  
Nokia  
R. Rokui  
Ciena  
March 6, 2022

IETF Network Slice YANG Data Model  
draft-liu-teas-transport-network-slice-yang-05

Abstract

This document describes a YANG data model for managing and controlling IETF network slices, defined in [I-D.liu-teas-ietf-network-slices].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 7, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                                     | 3  |
| 1.1. Terminology . . . . .                                    | 3  |
| 1.2. Tree Diagrams . . . . .                                  | 4  |
| 2. Modeling Considerations . . . . .                          | 4  |
| 2.1. Relationships to Related Topology Models . . . . .       | 4  |
| 2.2. Network Slice with TE . . . . .                          | 5  |
| 2.3. ACTN for Network Slicing . . . . .                       | 6  |
| 3. Model Applicability . . . . .                              | 6  |
| 3.1. Network Slicing by Virtualization . . . . .              | 7  |
| 3.2. Network Slicing by TE Overlay . . . . .                  | 9  |
| 4. Model Communication Types . . . . .                        | 11 |
| 4.1. P2P . . . . .  | 11 |
| 4.2. P2MP . . . . .   | 12 |
| 4.3. MP2MP . . . . .  | 13 |
| 4.4. A2A . . . . .  | 15 |
| 5. Model Tree Structure . . . . .                             | 16 |
| 5.1. Module ietf-network-slice . . . . .                      | 16 |
| 5.2. Module ietf-network-slice-connectivity . . . . .         | 17 |
| 6. YANG Modules . . . . .                                     | 17 |
| 6.1. Module ietf-network-slice . . . . .                      | 17 |
| 6.2. Module ietf-network-slice-connectivity . . . . .         | 23 |
| 7. IANA Considerations . . . . .                              | 26 |
| 8. Security Considerations . . . . .                          | 27 |
| 9. Acknowledgements . . . . .                                 | 28 |
| 10. References . . . . .                                      | 29 |
| 10.1. Normative References . . . . .                          | 29 |
| 10.2. Informative References . . . . .                        | 30 |
| Appendix A. Data Tree for the Example in Section 3.1. . . . . | 32 |
| A.1. Native Topology . . . . .                                | 32 |
| A.2. Network Slice Blue . . . . .                             | 36 |
| Authors' Addresses . . . . .                                  | 42 |

## 1. Introduction

This document defines a YANG [RFC7950] data model for representing, managing, and controlling IETF network slices, defined in [I-D.ietf-teas-ietf-network-slices]

The defined data model is an interface between customers and providers for configurations and state retrievals, so as to support network slicing as a service. Through this model, a customer can learn the slicing capabilities and the available resources of the provider. A customer can request or negotiate with a network slicing provider to create an instance. The customer can incrementally update its requirements on individual topology elements in the slice instance, and retrieve the operational states of these elements. With the help of other mechanisms and data models defined in IETF, the telemetry information can be published to the customer.

As described in Section 3 of [I-D.contreras-teas-slice-controller-models], the data model defined in this document complements the data model defined in [I-D.ietf-teas-ietf-network-slice-nbi-yang]. In addition to the provider's view, the data model defined in this document models the Type 2 service defined in [RFC8453].

The YANG data model in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

### 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC7950] and are not redefined here:

- o augment
- o data model
- o data node

## 1.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

## 2. Modeling Considerations

An IETF network slice is modeled as network topology defined in [RFC8345], with augmentations. A new network type "network-slice" is defined in this document. When a network topology data instance contains the network-slice network type, it represents an instance of an IETF network slice.

### 2.1. Relationships to Related Topology Models

There are several related YANG data models that have been defined in IETF. Some of these are:

Network Topology Model:  
Defined in [RFC8345].

OTN Topology Model:  
Defined in [I-D.ietf-ccamp-otn-topo-yang].

L2 Topology Model:  
Defined in [I-D.ietf-i2rs-yang-l2-network-topology].

L3 Topology Model:  
Defined in [RFC8346].

TE Topology Model:  
Defined in [RFC8795].

Figure 1 shows the relationships among these models. The box of dotted lines denotes the model defined in this document.

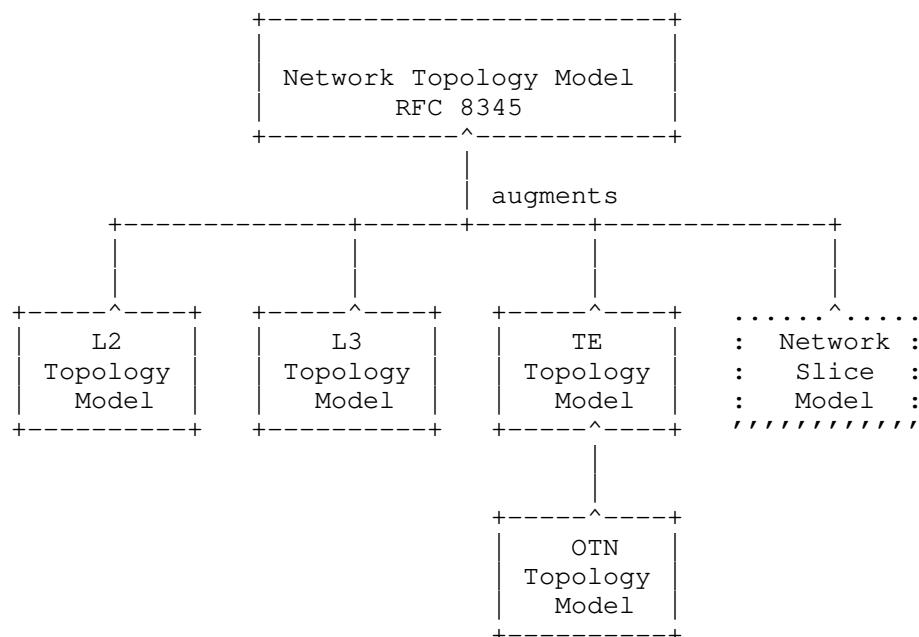


Figure 1: Model Relationships

## 2.2. Network Slice with TE

In many situations, an IETF network slice needs to have TE (Traffic Engineering) capabilities to achieve certain network characteristics. The TE Topology Model defined in [RFC8795] can be used to make an IETF network slice TE capable. To achieve this, an IETF network slice instance will be configured to have both "network-slice" and "te-topology" network types, taking advantage of the multiple inheritance capability featured by the network topology model [RFC8345]. The following diagram shows their relations.



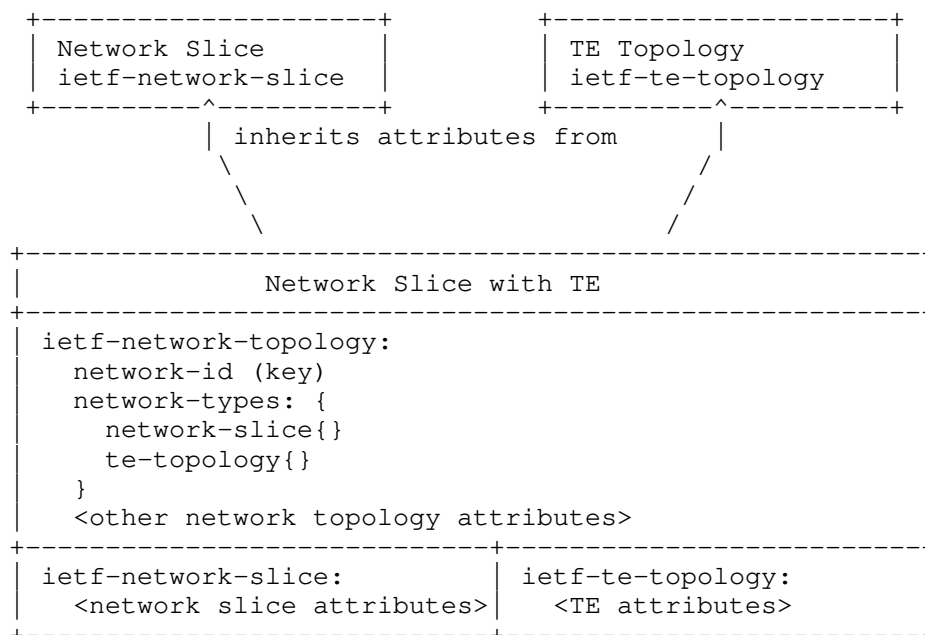


Figure 2: Network Slice with TE

This method can be applied to other types of network topology models too. For example, when a network topology instance is configured to have the types of "network-slice" defined in this document, "te-topology" defined in [RFC8795], and "l3-unicast-topology" defined in [RFC8346], this network topology instance becomes an IETF network slice instance that can perform layer 3 traffic engineering.

### 2.3. ACTN for Network Slicing

Since ACTN topology data models are based on the network topology model defined in [RFC8345], the augmentations defined in this document are effective augmentations to the ACTN topology data models, resulting in making the ACTN framework [RFC8453] and data models [I-D.ietf-teas-actn-yang] capable of slicing networks with the required network characteristics.

## 3. Model Applicability

There are many technologies to achieve network slicing. The data model defined in this document can be applied to a wide ranges of cases. This section describes how this data model is applied to a few cases.

### 3.1. Network Slicing by Virtualization

In the case shown in Figure 3, node virtualization is used to separate and allocate resources in physical devices. Two virtual routers VR1 and VR2 are created over physical router R1. Each of the virtual routers takes a portion of the resources such as ports and memory in the physical router. Depending on the requirements and the implementations, they may share certain resources such as processors, ASICs, and switch fabric.

As an example, Appendix A. shows the JSON encoded data instances of the native topology and the customized topology for Network Slice Blue.

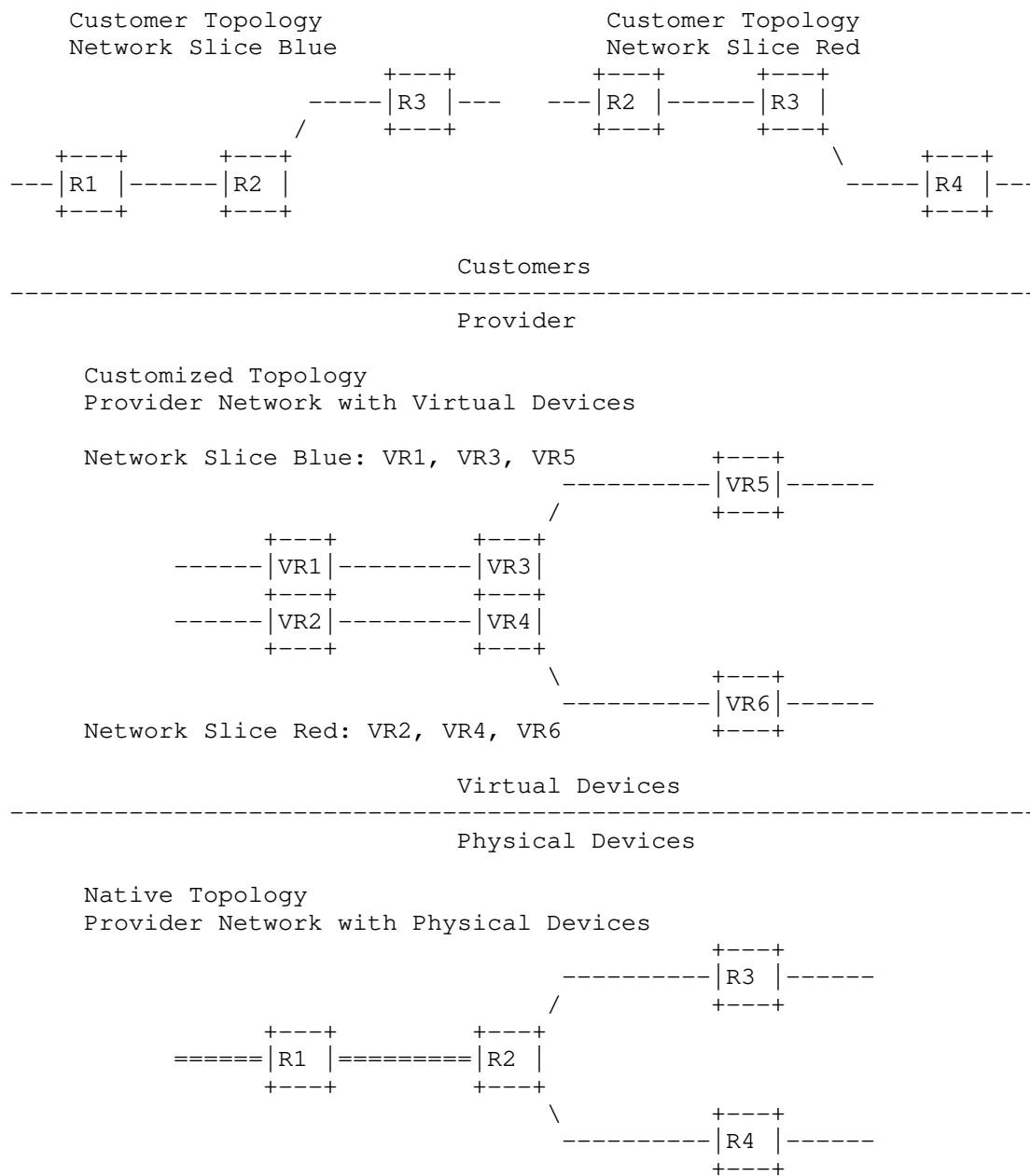


Figure 3: Network Slicing by Virtualization

### 3.2. Network Slicing by TE Overlay

Figure 4 shows a case where TE (Traffic Engineering) overlay is applied to achieve logically separated customer IETF network slices. In the underlay TE capable network, TE tunnels are established to support the TE links in the overlay network. These links and tunnels maintain the characteristics required by the customers. The provider selects the proper logical nodes and links in the overlay network, assigns them to specific IETF network slices, and uses the data model defined in this document to send the results to the customers.

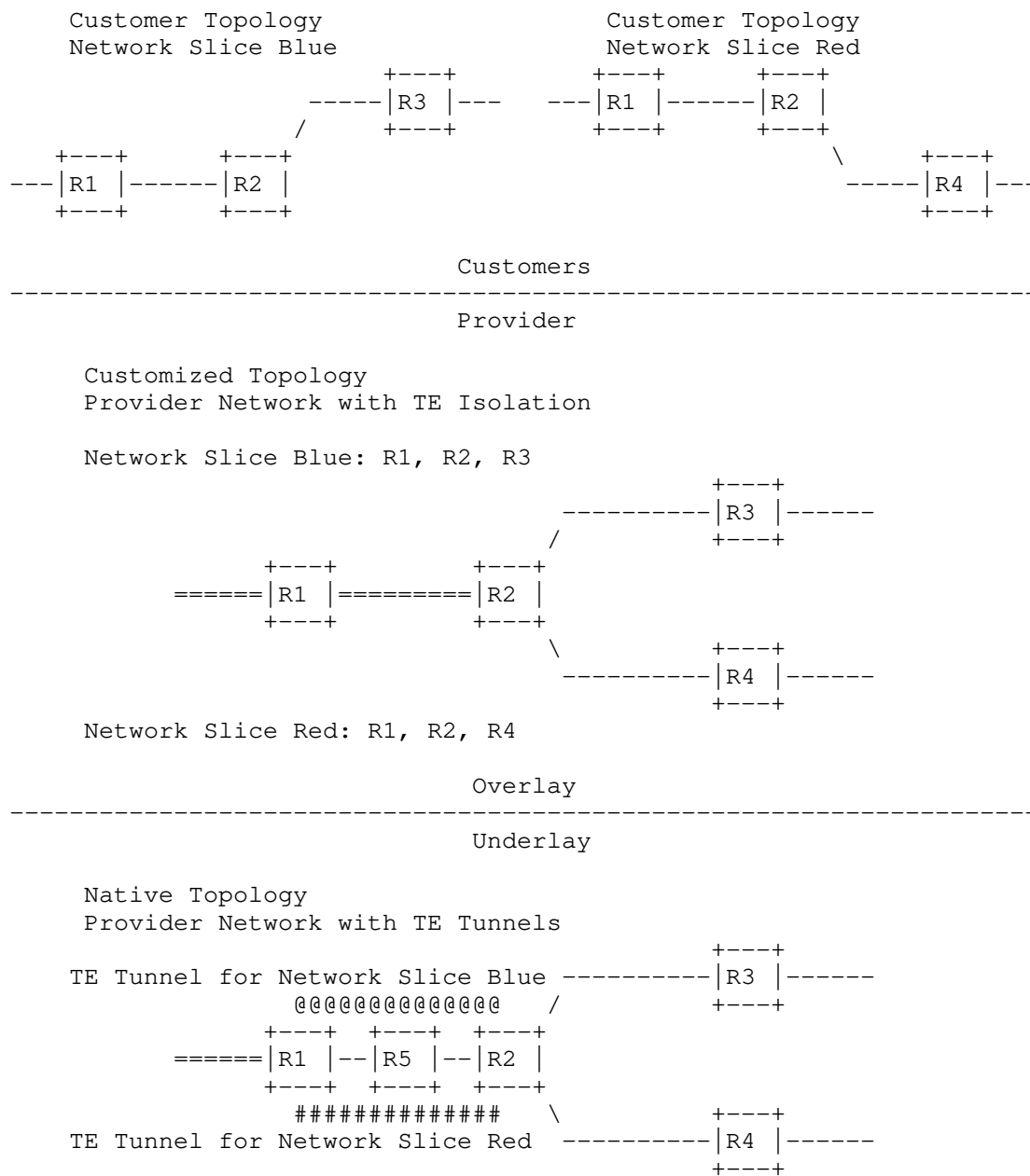


Figure 4: Network Slicing by TE Overlay

#### 4. Model Communication Types

Section 3.2 of [I-D.ietf-teas-ietf-network-slices] describes various communication types that an IETF network slice may serve, including P2P, P2MP, MP2P, MP2MP, and A2A. The data models specified in [RFC8345] and [RFC8795] support only P2P and A2A. In this document, the YANG module `ietf-network-slice-connectivity` is defined to extend the capabilities to cover P2MP, MP2P, and MP2MP.

The YANG module `ietf-network-slice-connectivity` is defined in Section 6.2 of this document, with its structure shown in Section 5.2 of this document. This YANG module introduces two modeling constructs in each connectivity construct (that is called connectivity matrix entries in [RFC8795]):

##### Replication Group:

A replication group contains a list of connectivity constructs (that are called connectivity matrix entries in RFC 8795). When traffic is sent to one entry in this replication group, the traffic is replicated to all other entries in the same replication group.

##### Receiver Constraint Group:

A receiver constraint group contains a list of connectivity constructs (that are called connectivity matrix entries in RFC 8795). When traffic is sent to one or more entries in this receiver constraint group, the constraints specified in this receiver constraint group are applied to the receiver-side termination points referenced by all entries in this receiver constraint group.

The following sections describe some data examples:

##### 4.1. P2P

NSE3 <-> NSC7 : Bidirectional P2P connectivity  
NSE4 -> NSE8 : Unidirectional P2P connectivity

```
{  
  "connectivity-matrices": {  
    "connectivity-matrix": [  
      {  
        "id": 1,  
        "from": {  
          "tp-ref": "NSE3"  
        },  
        "to": {  
          "tp-ref": "NSE7"  
        }  
      },  
      {  
        "id": 2,  
        "from": {  
          "tp-ref": "NSE7"  
        },  
        "to": {  
          "tp-ref": "NSE3"  
        }  
      },  
      {  
        "id": 3,  
        "from": {  
          "tp-ref": "NSE4"  
        },  
        "to": {  
          "tp-ref": "NSE8"  
        }  
      }  
    ]  
  }  
}
```

#### 4.2. P2MP

```
    NSE5 -> {NSC9, NSE10}
  {
    "connectivity-matrices": {
      "connectivity-matrix": [
        {
          "id": 1,
          "from": {
            "tp-ref": "NSE5"
          },
          "to": {
            "tp-ref": "NSE9"
          }
        },
        {
          "id": 2,
          "from": {
            "tp-ref": "NSE5"
          },
          "to": {
            "tp-ref": "NSE10"
          }
        }
      ],
      "replication-group": [
        {
          "id": 1,
          "entry": [1, 2]
        }
      ]
    }
  }
```

#### 4.3. MP2MP

```
    {NSE14, NSE15} -> {NSE16, NSE17}
  {
    "connectivity-matrices": {
      "connectivity-matrix": [
        {
          "id": 1,
          "from": {
            "tp-ref": "NSE14"
          },
          "to": {
            "tp-ref": "NSE16"
          }
        },
        {
          "id": 2,
          "from": {

```



```
        "tp-ref": "NSE14"
      },
      "to": {
        "tp-ref": "NSE17"
      }
    ],
    "connectivity-matrix": [
      "id": 3,
      "from": {
        "tp-ref": "NSE15"
      },
      "to": {
        "tp-ref": "NSE16"
      }
    ],
    "connectivity-matrix": [
      "id": 4,
      "from": {
        "tp-ref": "NSE15"
      },
      "to": {
        "tp-ref": "NSE17"
      }
    ],
    "replication-group": [
      "id": 1,
      "entry": [1, 2]
    ],
    "replication-group": [
      "id": 2,
      "entry": [3, 4]
    ],
    "receiver-constraint-group": [
      "id": 1,
      "entry": [1, 3]
    ],
    "receiver-constraint-group": [
      "id": 2,
      "entry": [2, 4]
    ]
  ]
}
```

## 4.4. A2A

```
{NSE1, NSE2, NSE6} -> {NSE1, NSE2, NSE6}

{
  "connectivity-matrices": {
    "connectivity-matrix": [
      {
        "id": 1,
        "from": {
          "tp-ref": "NSE1"
        },
        "to": {
          "tp-ref": "NSE2"
        }
      },
      {
        "id": 2,
        "from": {
          "tp-ref": "NSE1"
        },
        "to": {
          "tp-ref": "NSE6"
        }
      },
      {
        "id": 3,
        "from": {
          "tp-ref": "NSE2"
        },
        "to": {
          "tp-ref": "NSE1"
        }
      },
      {
        "id": 4,
        "from": {
          "tp-ref": "NSE2"
        },
        "to": {
          "tp-ref": "NSE6"
        }
      },
      {
        "id": 5,
        "from": {
          "tp-ref": "NSE6"
        }
      }
    ]
  }
}
```

```

        "to": {
          "tp-ref": "NSE1"
        }
      ],
      "connectivity-matrix": [
        {
          "id": 6,
          "from": {
            "tp-ref": "NSE6"
          },
          "to": {
            "tp-ref": "NSE2"
          }
        }
      ]
    }
  }
}

```

## 5. Model Tree Structure

### 5.1. Module ietf-network-slice

TODO - Complete IETF network slice attributes that are technology-agnostic and common to all use cases.

```

module: ietf-network-slice
  augment /nw:networks/nw:network/nw:network-types:
    +--rw network-slice!
  augment /nw:networks/nw:network:
    +--rw network-slice
      +--rw optimization-criterion?  identityref
      +--rw delay-tolerance?          boolean
      +--rw periodicity*              uint64
      +--rw isolation-level?          identityref
  augment /nw:networks/nw:network/nw:node:
    +--rw network-slice
      +--rw isolation-level?          identityref
      +--rw compute-node-id?         string
      +--rw storage-id?              string
  augment /nw:networks/nw:network/nt:link:
    +--rw network-slice
      +--rw delay-tolerance?          boolean
      +--rw periodicity*              uint64
      +--rw isolation-level?          identityref

```

## 5.2. Module ietf-network-slice-connectivity

```

module: ietf-network-slice-connectivity
  augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:connectivity-matrix:
  +--rw replication-group* [id]
  |   +--rw id          uint32
  |   +--rw entry*      -> ../../tet:id
  +--rw receiver-constraint-group* [id]
  |   +--rw id          uint32
  |   +--rw entry*      -> ../../tet:id
  |   +--rw te-bandwidth
  |   |   +--rw (technology)?
  |   |   |   +--:(generic)
  |   |   |   +--rw generic?   te-bandwidth
  augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:connectivity-matrix:
  +--ro replication-group* [id]
  |   +--ro id          uint32
  |   +--ro entry*      -> ../../tet:id
  +--ro receiver-constraint-group* [id]
  |   +--ro id          uint32
  |   +--ro entry*      -> ../../tet:id
  |   +--ro te-bandwidth
  |   |   +--ro (technology)?
  |   |   |   +--:(generic)
  |   |   |   +--ro generic?   te-bandwidth

```

## 6. YANG Modules

### 6.1. Module ietf-network-slice

This module references [RFC8345], [RFC8776], and [GSMA-NS-Template]

```

<CODE BEGINS> file "ietf-network-slice@2020-11-01.yang"
module ietf-network-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-slice";
  prefix "ns";

  import ietf-network {
    prefix "nw";
    reference "RFC 8345: A YANG Data Model for Network Topologies";

```

```
}
import ietf-network-topology {
  prefix "nt";
  reference "RFC 8345: A YANG Data Model for Network Topologies";
}
import ietf-te-types {
  prefix "te-types";
  reference
    "RFC 8776: Traffic Engineering Common YANG Types";
}
```

organization

"IETF Traffic Engineering Architecture and Signaling (TEAS)  
Working Group";

contact

"WG Web: <<http://tools.ietf.org/wg/teas/>>  
WG List: <<mailto:teas@ietf.org>>

Editor: Xufeng Liu  
<<mailto:xufeng.liu.ietf@gmail.com>>

Editor: Jeff Tantsura  
<<mailto:jefftant.ietf@gmail.com>>

Editor: Igor Bryskin  
<[mailto:i\\_bryskin@yahoo.com](mailto:i_bryskin@yahoo.com)>

Editor: Luis Miguel Contreras Murillo  
<<mailto:luismiguel.contrerasmurillo@telefonica.com>>

Editor: Qin Wu  
<<mailto:bill.wu@huawei.com>>

Editor: Sergio Belotti  
<<mailto:sergio.belotti@nokia.com>>

Editor: Reza Rokui  
<<mailto:reza.rokui@nokia.com>>

";

description

"YANG data model for representing and managing network  
slices.

Copyright (c) 2020 IETF Trust and the persons identified as  
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2020-11-01 {
  description "Initial revision";
  reference
    "RFC XXXX: YANG Data Model for Network Slices";
}

/*
 * Identities
 */
identity isolation-level {
  description
    "Base identity for the isolation-level.";
  reference
    "GSMA-NS-Template: Generic Network Slice Template,
    Version 3.0.";
}
identity no-isolation {
  base isolation-level;
  description
    "Network slices are not separated.";
}
identity physical-isolation {
  base isolation-level;
  description
    "Network slices are physically separated (e.g. different rack,
    different hardware, different location, etc.).";
}
identity logical-isolation {
  base isolation-level;
  description
    "Network slices are logically separated.";
}
identity process-isolation {
  base physical-isolation;
  description
    "Process and threads isolation.";
}
identity physical-memory-isolation {
```

```
    base physical-isolation;
    description
      "Process and threads isolation.";
  }
  identity physical-network-isolation {
    base physical-isolation;
    description
      "Process and threads isolation.";
  }
  identity virtual-resource-isolation {
    base logical-isolation;
    description
      "A network slice has access to specific range of resources
       that do not overlap with other network slices
       (e.g. VM isolation).";
  }
  identity network-functions-isolation {
    base logical-isolation;
    description
      "NF (Network Function) is dedicated to the network slice, but
       virtual resources are shared.";
  }
  identity service-isolation {
    base logical-isolation;
    description
      "NSC data are isolated from other NSCs, but virtual
       resources and NFs are shared.";
  }
}

/*
 * Groupings
 */
grouping network-slice-topology-attributes {
  description "Network Slice topology scope attributes.";
  container network-slice {
    description
      "Containing Network Slice attributes.";
    leaf optimization-criterion {
      type identityref {
        base te-types:objective-function-type;
      }
      description
        "Optimization criterion applied to this topology.";
    }
    leaf delay-tolerance {
      type boolean;
      description
        "'true' if is not too critical how long it takes to deliver
```

```
        the amount of data.";
    reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
}
leaf-list periodicity {
    type uint64;
    units seconds;
    description
        "A list of periodicities supported by the network slice.";
    reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
}
leaf isolation-level {
    type identityref {
        base isolation-level;
    }
    description
        "A network slice instance may be fully or partly, logically
        and/or physically, isolated from another network slice
        instance. This attribute describes different types of
        isolation:";
}
} // network-slice
} // network-slice-topology-attributes

grouping network-slice-node-attributes {
    description "Network Slice node scope attributes.";
    container network-slice {
        description
            "Containing Network Slice attributes.";
        leaf isolation-level {
            type identityref {
                base isolation-level;
            }
            description
                "A network slice instance may be fully or partly, logically
                and/or physically, isolated from another network slice
                instance. This attribute describes different types of
                isolation:";
        }
        leaf compute-node-id {
            type string;
            description
                "Reference to a compute node instance specified in
                a data model specifying the computing resources.";
        }
    }
}
```



```
    leaf storage-id {
        type string;
        description
            "Reference to a storage instance specified in
             a data model specifying the storage resources.";
    }
} // network-slice
} // network-slice-node-attributes

grouping network-slice-link-attributes {
    description "Network Slice link scope attributes";
    container network-slice {
        description
            "Containing Network Slice attributes.";
        leaf delay-tolerance {
            type boolean;
            description
                "'true' if is not too critical how long it takes to deliver
                 the amount of data.";
            reference
                "GSMA-NS-Template: Generic Network Slice Template,
                 Version 3.0.";
        }
        leaf-list periodicity {
            type uint64;
            units seconds;
            description
                "A list of periodicities supported by the network slice.";
            reference
                "GSMA-NS-Template: Generic Network Slice Template,
                 Version 3.0.";
        }
        leaf isolation-level {
            type identityref {
                base isolation-level;
            }
            description
                "A network slice instance may be fully or partly, logically
                 and/or physically, isolated from another network slice
                 instance. This attribute describes different types of
                 isolation:";
        }
    } // network-slice
} // network-slice-link-attributes

/*
 * Data nodes
 */
```

```
augment "/nw:networks/nw:network/nw:network-types" {
  description
    "Defines the Network Slice topology type.";
  container network-slice {
    presence "Indicates Network Slice topology";
    description
      "Its presence identifies the Network Slice type.";
  }
}

augment "/nw:networks/nw:network" {
  when "nw:network-types/ns:network-slice" {
    description "Augment only for Network Slice topology.";
  }
  description "Augment topology configuration and state.";
  uses network-slice-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node" {
  when "../nw:network-types/ns:network-slice" {
    description "Augment only for Network Slice topology.";
  }
  description "Augment node configuration and state.";
  uses network-slice-node-attributes;
}

augment "/nw:networks/nw:network/nt:link" {
  when "../nw:network-types/ns:network-slice" {
    description "Augment only for Network Slice topology.";
  }
  description "Augment link configuration and state.";
  uses network-slice-link-attributes;
}
}
<CODE ENDS>
```

## 6.2. Module ietf-network-slice-connectivity

This module references [RFC8345], [RFC8776], and [RFC8795]

```
<CODE BEGINS> file "ietf-network-slice-connectivity@2022-03-04.yang"
module ietf-network-slice-connectivity {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:"
```

```
+ "ietf-network-slice-connectivity";
prefix "ns-con-types";

import ietf-network {
  prefix "nw";
  reference "RFC 8345: A YANG Data Model for Network Topologies";
}
import ietf-te-topology {
  prefix "tet";
  reference
    "RFC 8795: YANG Data Model for Traffic Engineering (TE)
    Topologies";
}
import ietf-te-types {
  prefix "te-types";
  reference
    "RFC 8776: Traffic Engineering Common YANG Types";
}
```

```
organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
```

```
contact
  "WG Web:  <http://tools.ietf.org/wg/teas/>
  WG List:  <mailto:teas@ietf.org>

  Editor:   Xufeng Liu
            <mailto:xufeng.liu.ietf@gmail.com>

  Editor:   Luis Miguel Contreras Murillo
            <mailto:luismiguel.contrerasmurillo@telefonica.com>

  Editor:   Sergio Belotti
            <mailto:sergio.belotti@nokia.com>
  ";
```

```
description
  "YANG augmentations to support various connectivity types for
  IETF network slices.
```

Copyright (c) 2022 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions

Relating to IETF Documents  
(<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2022-03-04 {
  description "Initial revision";
  reference
    "RFC XXXX: YANG Data Model for Network Slices";
}

/*
 * Groupings
 */
grouping network-slice-connectivity-types {
  description "Network Slice topology scope attributes.";
  list replication-group {
    key "id";
    description
      "A list of replication groups. Each replication group
       contains a list of connectivity constructs
       (that are called connectivity matrix entries in RFC 8795).
       When traffic is sent to one entry in this replication group,
       the traffic is replicated to all other entries in the same
       replication group.";
    leaf id {
      type uint32;
      description
        "Identifies the replication group.";
    }
    leaf-list entry {
      type leafref {
        path "../..tet:id";
      }
      description
        "References a connectivity matrix entry that belongs to
         this replication group.";
    }
  }
}
list receiver-constraint-group {
  key "id";
  description
    "A list of receiver constraint groups. Each receiver
     constraint group contains a list of connectivity constructs
     (that are called connectivity matrix entries in RFC 8795).
     When traffic is sent to one or more entries in this
     receiver constraint group, the constraints specified in this
```

```
        receiver constraint group are applied to the receiver-side
        termination points referenced by all entries in this
        receiver constraint group.";
    leaf id {
        type uint32;
        description
            "Identifies the receiver constraint group.";
    }
    leaf-list entry {
        type leafref {
            path "../tet:id";
        }
        description
            "References a connectivity matrix entry that belongs to
            this receiver constraint group..";
    }
    uses te-types:te-bandwidth;
}

/*
 * Data nodes
 */
augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix" {
    description "Augment node configuration and state.";
    uses network-slice-connectivity-types;
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix" {
    description "Augment node configuration and state.";
    uses network-slice-connectivity-types;
}
}
<CODE ENDS>
```

## 7. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

---

URI: urn:ietf:params:xml:ns:yang:ietf-network-slice  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

---

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

---

|            |  |
|------------|--|
| name:      | ietf-l3-te-topology                            |
| namespace: | urn:ietf:params:xml:ns:yang:ietf-network-slice |
| prefix:    | ns   |
| reference: | RFC XXXX                                       |

---

## 8. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/nw:networks/nw:network/nw:network-types/ns:network-slice  
This subtree specifies the network slice type. Modifying the configurations can make network slice type invalid and cause interruption to IETF network slices.

/nw:networks/nw:network/ns:network-slice  
This subtree specifies the topology-wide configurations.  
Modifying the configurations here can cause traffic

characteristics changed in this IETF network slice and related networks.

/nw:networks/nw:network/nw:node/ns:network-slice

This subtree specifies the configurations of the nodes in a IETF network slice. Modifying the configurations in this subtree can change the traffic characteristics on this node and the related networks.

/nw:networks/nw:network/nt:link/ns:network-slice

This subtree specifies the configurations of the links in a IETF network slice. Modifying the configurations in this subtree can change the traffic characteristics on this link and the related networks.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/nw:networks/nw:network/nw:network-types/ns:network-slice

Unauthorized access to this subtree can disclose the network slice type.

/nw:networks/nw:network/ns:network-slice

Unauthorized access to this subtree can disclose the topology-wide states.

/nw:networks/nw:network/nw:node/ns:network-slice

Unauthorized access to this subtree can disclose the operational state information of the nodes in a IETF network slice.

/nw:networks/nw:network/nt:link/ns:network-slice

Unauthorized access to this subtree can disclose the operational state information of the links in a IETF network slice.

## 9. Acknowledgements

The TEAS Network Slicing Design Team (NSDT) members included Aijun Wang, Dong Jie, Eric Gray, Jari Arkko, Jeff Tantsura, John E Drake, Luis M. Contreras, Rakesh Gandhi, Ran Chen, Reza Rokui, Ricard Vilalta, Ron Bonica, Sergio Belotti, Tomonobu Niwa, Xuesong Geng, and Xufeng Liu.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.



- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.
- [GSMA-NS-Template] GSM Association, "Generic Network Slice Template, Version 3.0", NG.116, May 2020.
- [I-D.ietf-teas-ietf-network-slices] Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", draft-ietf-teas-ietf-network-slices-07 (work in progress), March 2022.

## 10.2. Informative References

- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [I-D.ietf-ccamp-otn-topo-yang]  
Zheng, H., Busi, I., Liu, X., Belotti, S., and O. G. D. Dios, "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang-13 (work in progress), July 2021.
- [I-D.ietf-i2rs-yang-l2-network-topology]  
Dong, J., Wei, X., Wu, Q., Boucadair, M., and A. Liu, "A YANG Data Model for Layer 2 Network Topologies", draft-ietf-i2rs-yang-l2-network-topology-18 (work in progress), September 2020.
- [I-D.ietf-teas-actn-yang]  
Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B. Y., and S. Belotti, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-yang-08 (work in progress), September 2021.
- [I-D.ietf-teas-ietf-network-slice-nbi-yang]  
Wu, B., Dhody, D., Rokui, R., Saad, T., and L. Han, "IETF Network Slice Service YANG Model", draft-ietf-teas-ietf-network-slice-nbi-yang-01 (work in progress), March 2022.
- [I-D.contreras-teas-slice-controller-models]  
Contreras, L. M., Rokui, R., Tantsura, J., Wu, B., Liu, X., Dhody, D., and S. Bellotti, "IETF Network Slice Controller and its associated data models", draft-contreras-teas-slice-controller-models-02 (work in progress), March 2022.

## Appendix A. Data Tree for the Example in Section 3.1.

## A.1. Native Topology

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network" for the native topology depicted in Figure 3.

```
{
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "example-native-topology",
        "network-types": {
        },
        "node": [
          {
            "node-id": "R1",
            "ietf-network-topology:termination-point": [
              {
                "tp-id": "1-0-1"
              },
              {
                "tp-id": "1-0-2"
              },
              {
                "tp-id": "1-2-1"
              },
              {
                "tp-id": "1-2-2"
              }
            ]
          },
          {
            "node-id": "R2",
            "ietf-network-topology:termination-point": [
              {
                "tp-id": "2-1-1"
              },
              {
                "tp-id": "2-1-2"
              },
              {
                "tp-id": "2-3-1"
              },
              {
                "tp-id": "2-4-1"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ]
  },
  {
    "node-id": "R3",
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "3-0-1"
      },
      {
        "tp-id": "3-2-1"
      }
    ]
  },
  {
    "node-id": "R4",
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "4-0-1"
      },
      {
        "tp-id": "4-2-1"
      }
    ]
  }
],
"ietf-network-topology:link": [
  {
    "link-id": "R1,1-0-1,,",
    "source": {
      "source-node": "R1",
      "source-tp": "1-0-1"
    }
  },
  {
    "link-id": ",,R1,1-0-1",
    "destination": {
      "dest-node": "R1",
      "dest-tp": "1-0-1"
    }
  },
  {
    "link-id": "R1,1-0-2,,",
    "source": {
      "source-node": "R1",
      "source-tp": "1-0-2"
    }
  },
  {

```

```
    "link-id":",,R1,1-0-2",
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-0-2"
    }
  },
  {
    "link-id":"R1,1-2-1,R2,2-1-1",
    "source": {
      "source-node":"R1",
      "source-tp":"1-2-1"
    },
    "destination": {
      "dest-node":"R2",
      "dest-tp":"2-1-1"
    }
  },
  {
    "link-id":"R2,2-1-1,R1,1-2-1",
    "source": {
      "source-node":"R2",
      "source-tp":"2-1-1"
    },
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-2-1"
    }
  },
  {
    "link-id":"R1,1-2-2,R2,2-1-2",
    "source": {
      "source-node":"R1",
      "source-tp":"1-2-2"
    },
    "destination": {
      "dest-node":"R2",
      "dest-tp":"2-1-2"
    }
  },
  {
    "link-id":"R2,2-1-2,R1,1-2-2",
    "source": {
      "source-node":"R2",
      "source-tp":"2-1-2"
    },
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-2-2"
    }
  }
}
```

```
    }
  },
  {
    "link-id": "R2,2-3-1,R3,3-2-1",
    "source": {
      "source-node": "R2",
      "source-tp": "2-3-1"
    },
    "destination": {
      "dest-node": "R3",
      "dest-tp": "3-2-1"
    }
  },
  {
    "link-id": "R3,3-2-1,R2,2-3-1",
    "source": {
      "source-node": "R3",
      "source-tp": "3-2-1"
    },
    "destination": {
      "dest-node": "R2",
      "dest-tp": "2-3-1"
    }
  },
  {
    "link-id": "R2,2-4-1,R4,4-2-1",
    "source": {
      "source-node": "R2",
      "source-tp": "2-4-1"
    },
    "destination": {
      "dest-node": "R4",
      "dest-tp": "4-2-1"
    }
  },
  {
    "link-id": "R4,4-2-1,R2,2-4-1",
    "source": {
      "source-node": "R4",
      "source-tp": "4-2-1"
    },
    "destination": {
      "dest-node": "R2",
      "dest-tp": "2-4-1"
    }
  },
  {
    "link-id": "R3,3-0-1,,",
```

```

        "source": {
          "source-node": "R3",
          "source-tp": "3-0-1"
        }
      },
      {
        "link-id": ",,R3,3-0-1",
        "destination": {
          "dest-node": "R3",
          "dest-tp": "3-0-1"
        }
      },
      {
        "link-id": "R4,4-0-1,,",
        "source": {
          "source-node": "R4",
          "source-tp": "4-0-1"
        }
      },
      {
        "link-id": ",,R4,4-0-1",
        "destination": {
          "dest-node": "R4",
          "dest-tp": "4-0-1"
        }
      }
    ]
  }
}

```

#### A.2. Network Slice Blue

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network-slice" for the topology customized for Network Slice Blue depicted in Figure 3.

```

{
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "example-customized-blue-topology",
        "network-types": {
          "ietf-network-slice:network-slice": {

```

```
    }
  },
  "supporting-network": [
    {
      "network-ref": "example-native-topology"
    }
  ],
  "node": [
    {
      "node-id": "VR1",
      "supporting-node": [
        {
          "network-ref": "example-native-topology",
          "node-ref": "R1"
        }
      ],
      "ietf-network-slice:network-slice": {
        "isolation-level":
          "ietf-network-slice:physical-memory-isolation"
      },
      "ietf-network-topology:termination-point": [
        {
          "tp-id": "1-0-1"
        },
        {
          "tp-id": "1-3-1"
        }
      ]
    },
    {
      "node-id": "VR3",
      "supporting-node": [
        {
          "network-ref": "example-native-topology",
          "node-ref": "R2"
        }
      ],
      "ietf-network-slice:network-slice": {
        "isolation-level":
          "ietf-network-slice:physical-memory-isolation"
      },
      "ietf-network-topology:termination-point": [
        {
          "tp-id": "3-1-1"
        },
        {
          "tp-id": "3-5-1"
        }
      ]
    }
  ]
}
```



```

    ]
  },
  {
    "node-id": "VR5",
    "supporting-node": [
      {
        "network-ref": "example-native-topology",
        "node-ref": "R3"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-memory-isolation"
    },
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "5-3-1"
      },
      {
        "tp-id": "5-0-1"
      }
    ]
  }
],
"ietf-network-topology:link": [
  {
    "link-id": "VR1,1-0-1,,",
    "source": {
      "source-node": "VR1",
      "source-tp": "1-0-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R1,1-0-1,,"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": ",,VR1,1-0-1",
    "destination": {
      "dest-node": "VR1",
      "dest-tp": "1-0-1"
    }
  },

```

```
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": ",,R1,1-0-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR1,1-3-1,VR3,3-1-1",
    "source": {
      "source-node": "VR1",
      "source-tp": "1-3-1"
    },
    "destination": {
      "dest-node": "VR3",
      "dest-tp": "3-1-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R1,1-2-1,R2,2-1-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR3,3-1-1,VR1,1-3-1",
    "source": {
      "source-node": "VR3",
      "source-tp": "3-1-1"
    },
    "destination": {
      "dest-node": "R1",
      "dest-tp": "1-3-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R2,2-1-1,R1,1-2-1"
      }
    ],
  },
```

```
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR3,3-5-1,VR5,5-3-1",
    "source": {
      "source-node": "VR3",
      "source-tp": "3-5-1"
    },
    "destination": {
      "dest-node": "VR5",
      "dest-tp": "5-3-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R2,2-3-1,R3,3-2-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR5,5-3-1,VR3,3-5-1",
    "source": {
      "source-node": "VR5",
      "source-tp": "5-3-1"
    },
    "destination": {
      "dest-node": "VR3",
      "dest-tp": "3-5-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R3,3-2-1,R2,2-3-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  }
],
{
```

```

        "link-id": "VR5,5-0-1,,",
        "source": {
            "source-node": "VR5",
            "source-tp": "5-0-1"
        },
        "supporting-link": [
            {
                "network-ref": "example-native-topology",
                "link-ref": "R3,3-0-1,,",
            }
        ],
        "ietf-network-slice:network-slice": {
            "isolation-level":
                "ietf-network-slice:physical-network-isolation"
        }
    },
    {
        "link-id": ",,VR5,5-0-1",
        "destination": {
            "dest-node": "VR5",
            "dest-tp": "5-0-1"
        },
        "supporting-link": [
            {
                "network-ref": "example-native-topology",
                "link-ref": ",,R3,3-0-1"
            }
        ],
        "ietf-network-slice:network-slice": {
            "isolation-level":
                "ietf-network-slice:physical-network-isolation"
        }
    }
],
"ietf-network-slice:network-slice": {
    "optimization-criterion":
        "ietf-te-types:of-minimize-cost-path",
    "isolation-level":
        "ietf-network-slice:physical-isolation"
}
}
]
}
}

```

Authors' Addresses

Xufeng Liu  
IBM Corporation

EMail: xufeng.liu.ietf@gmail.com

Jeff Tantsura  
Microsoft

EMail: jefftant.ietf@gmail.com

Igor Bryskin  
Individual

EMail: i\_bryskin@yahoo.com

Luis Miguel Contreras Murillo  
Telefonica

EMail: luismiguel.contrerasmurillo@telefonica.com

Qin Wu  
Huawei

EMail: bill.wu@huawei.com

Sergio Belotti  
Nokia

EMail: sergio.belotti@nokia.com

Reza Rokui  
Ciena  
Canada

EMail: rrokui@Ciena.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 10, 2022

B. Wu  
D. Dhody  
Huawei Technologies  
R. Rokui  
Nokia  
T. Saad  
Juniper Networks  
L. Han  
China Mobile  
July 9, 2021

A Yang Data Model for IETF Network Slice NBI  
draft-wd-teas-ietf-network-slice-nbi-yang-03

## Abstract

This document provides a YANG data model for the IETF Network Slice Controller (NSC) Northbound Interface (NBI). The model can be used by a IETF Network Slice customer to request configuration, and management IETF Network Slice services from the IETF NSC.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 10, 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .   | 2  |
| 2. Conventions used in this document . . . . .  | 3  |
| 2.1. Tree Diagrams . . . . .  | 4  |
| 3. IETF Network Slice NBI Model Usage . . . . .   | 4  |
| 4. IETF Network Slice NBI Model Overview . . . . .  | 5  |
| 5. IETF Network Slice Templates . . . . .   | 9  |
| 6. IETF Network Slice Modeling Description . . . . .  | 10 |
| 6.1. IETF Network Slice Connectivity Type . . . . .   | 11 |
| 6.2. IETF Network Slice SLO and SLE Policy . . . . .  | 11 |
| 6.3. IETF Network Slice Endpoint (NSE) . . . . .  | 13 |
| 7. IETF Network Slice Monitoring . . . . .  | 16 |
| 8. IETF Network Slice NBI Module . . . . .  | 17 |
| 9. Security Considerations . . . . .  | 35 |
| 10. IANA Considerations . . . . .   | 36 |
| 11. Acknowledgments . . . . .   | 36 |
| 12. References . . . . .  | 37 |
| 12.1. Normative References . . . . .  | 37 |
| 12.2. Informative References . . . . .  | 38 |
| Appendix A. IETF Network Slice NBI Model Usage Example . . . . .                                  | 39 |
| Appendix B. Comparison with Other Possible Design choices for<br>IETF Network Slice NBI . . . . . | 42 |
| B.1. ACTN VN Model Augmentation . . . . .   | 42 |
| B.2. RFC8345 Augmentation Model . . . . .   | 43 |
| Appendix C. Appendix B IETF Network Slice Match Criteria . . . . .                                | 43 |
| Authors' Addresses . . . . .  | 45 |

## 1. Introduction

This document provides a YANG [RFC7950] data model for the IETF Network Slice NBI.

The YANG model discussed in this document is defined based on the description of the IETF Network Slice in [I-D.ietf-teas-ietf-network-slices], which is used to operate IETF Network Slice during the IETF Network Slice instantiation. This YANG model supports various operations on IETF Network Slices such as creation, modification, deletion, and monitoring of IETF Network Slices.

The IETF Network Slice Controller (NSC) provides a Northbound Interface (NBI) that allows customers of network slices to request and monitor IETF network slices.

The NBI carries information that the IETF network slice customer provides, describing generic requirements of connectivity, service level objectives (SLO), etc. and also monitoring and reporting requirements that may apply. It is an abstract interface that hides excessive technology-related information which may then be realized using some technology-specific Southbound Interface (SBI) by the NSC.

The YANG model discussed in this document describes the requirements of an IETF Network Slice from the point of view of the customer, which is classified as Customer Service Model in [RFC8309].

It will be up to the management system or NSC to take this model as an input and use other management system or specific configuration models to configure the different network elements to deliver an IETF Network Slice. The YANG models can be used with network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The details of how the IETF network slices are realized by the NSC is out of scope for this document.

The IETF Network Slice operational state is included in the same tree as the configuration consistent with Network Management Datastore Architecture [RFC8342].

## 2. Conventions used in this document

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC6241] and are used in this specification:

- o client
- o configuration data
- o state data

This document makes use of the following terminology introduced in the YANG 1.1 Data Modeling Language [RFC7950]:

- o augment



- o data model
- o data node

This document also makes use of the terms introduced in the Framework for IETF Network Slices [I-D.ietf-teas-ietf-network-slices]:

- o NBI: Northbound Interface
- o NS: IETF Network Slice
- o NSC: IETF Network Slice Controller
- o NSE: Network Slice Endpoint
- o SLO: Service Level Objective
- o SLE: Service Level Expectation

This document defines the following term:

- o IETF Network Slice Connection (NS-Connection): In the context of an IETF Network Slice, an IETF NS-Connection is an abstract entity which represents a particular connection between a pair of NSEs. An IETF Network Slice can has one or multiple NS-Connections.

## 2.1. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

## 3. IETF Network Slice NBI Model Usage

The intention of the IETF Network Slice NBI model is to allow the customer, e.g. a higher-level management system, to request and monitor IETF Network Slices. In particular, the model allows customers to operate on abstract and technology-agnostic manner, with details of the IETF Network Slices realization hidden.

According to the [I-D.ietf-teas-ietf-network-slices] description, the NBI model is applicable to use cases such as (but not limited to) network wholesale services, network infrastructure sharing among operators, NFV connectivity, Data Center Interconnect, and 5G E2E network slice.

As shown in Figure 1, in all these use-cases, the NBI model is used by the higher management system to communicate with IETF Network Slice controller for life cycle manage of IETF Network Slices

including both enablement and monitoring. For example, in 5G E2E network slicing use-case the E2E network slice orchestrator acts as the higher layer system to request the IETF Network Slices. The interface is used to support dynamic IETF Network Slice creation and its lifecycle management to facilitate end-to-end network slice services.

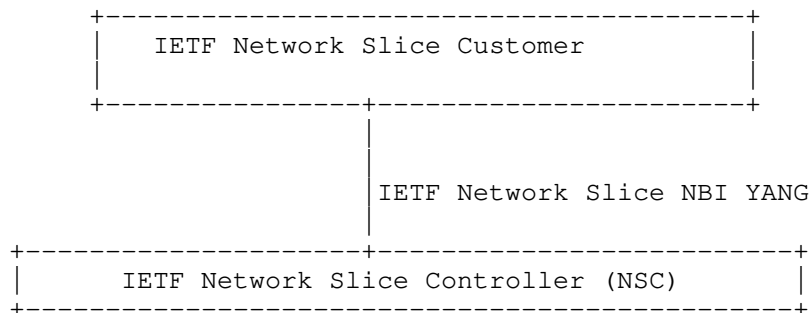
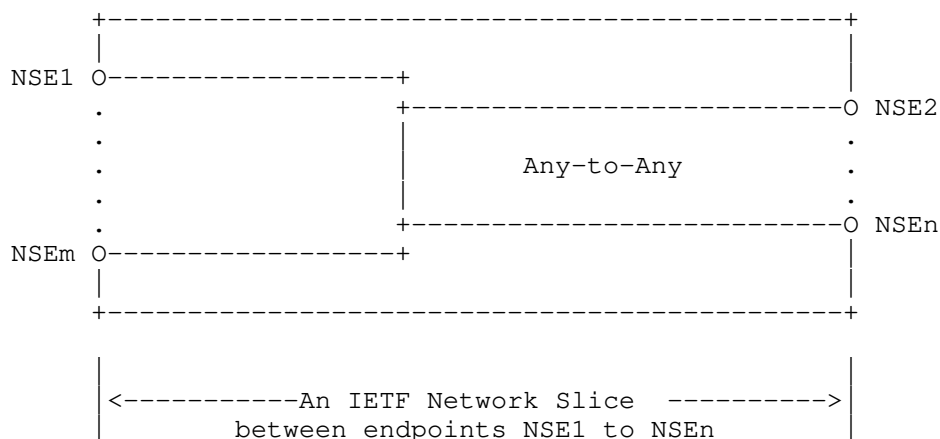


Figure 1: IETF Network Slice NBI Model Context

#### 4. IETF Network Slice NBI Model Overview

As defined in [I-D.ietf-teas-ietf-network-slices], an IETF network slice is a logical network connecting a number of endpoints with specified SLOs. The connectivity type can be Hub-and-Spoke, any-to-any, or custom connectivity type. In addition, a minimum set of SLOs is defined, including but not limited to bandwidth, latency, and etc. An example of an IETF network slice is shown in Figure 2 .



Legend:

NSE: IETF Network Slice Endpoint

O: Represents IETF Network Slice Endpoints

Figure 2: An IETF Network Slice Example

[I-D.ietf-teas-ietf-network-slices] introduces the IETF network slice endpoints (NSEs) which are conceptual points of connection to IETF network slice. As such, they are ingress/egress point where the traffic enters/exits the IETF network slice. In other words, they are the edge of the IETF network slices.

When IETF network slice controller (NSC) receives a message via its NBI for creation/modification of an IETF network slice, it uses the provided IETF network slice endpoints to map them to appropriate services/tunnels/paths endpoints in the underlay IETF network. It then uses services/tunnels/paths endpoints to realize the IETF network slice.

The IETF Network Slice ("ietf-network-slice") is defined to manage network slices in the IETF network. In particular, the 'ietf-network-slice' module can be used to create, modify, and monitor network slices of an IETF network.

The 'ietf-network-slice' module uses two main nodes: list 'ietf-network-slice' and container 'ns-templates' (see Figure 3).

The 'ietf-network-slice' list includes the set of IETF Network slices managed within IETF network. 'ietf-network-slice' is the data structure that abstracts an IETF Network Slice. Under the "ietf-network-slice", list "ns-endpoint" is used to abstract the NSEs, e.g. NSEs in the example above.

The 'ns-templates' container is used by the NSC to maintain a set of common network slice templates that apply to one or several IETF Network Slices.

The figure below describes the overall structure of the YANG module:

```

module: ietf-network-slice
  +--rw network-slices
    +--rw ns-slo-sle-templates
      |   +--rw ns-slo-sle-template* [id]
      |   |   +--rw id                string
      |   |   +--rw template-description?  string
      |   +--rw network-slice* [ns-id]
      |   |   +--rw ns-id                string
      |   |   +--rw ns-description?      string
      |   |   +--rw ns-tag*              string
      |   |   +--rw ns-connectivity-type? identityref
      |   |   +--rw (ns-slo-sle-policy)?
      |   |   |   +--:(standard)
      |   |   |   |   +--rw slo-sle-template?  leafref
      |   |   |   +--:(custom)
      |   |   |   |   +--rw slo-policy
      |   |   |   |   |   +--rw policy-description?  string
      |   |   |   |   |   +--rw ns-metric-bounds
      |   |   |   |   |   |   +--rw ns-metric-bound* [metric-type]
      |   |   |   |   |   |   |   +--rw metric-type      identityref
      |   |   |   |   |   |   |   +--rw metric-unit      string
      |   |   |   |   |   |   |   +--rw value-description? string
      |   |   |   |   |   |   |   +--rw bound?          uint64
      |   |   |   |   +--rw sle-policies
      |   |   |   |   |   +--rw security-sle*      identityref
      |   |   |   |   |   +--rw isolation?          identityref
      |   |   |   |   |   +--rw max-occupancy-level? uint8
      |   +--rw status
      |   |   +--rw admin-enabled?  boolean
      |   |   +--ro oper-status?    operational-type
      +--rw ns-endpoints
        +--rw ns-endpoint* [ep-id]
        |   +--rw ep-id                string
        |   +--rw ep-description?      string
        |   +--rw ep-role?              identityref
        |   +--rw location
        |   |   +--rw altitude?        int64
        |   |   +--rw latitude?        decimal64
        |   |   +--rw longitude?       decimal64
        |   +--rw node-id?              string
        |   +--rw ep-ip?                 inet:host
        +--rw ns-match-criteria
  
```

```

    +--rw ns-match-criterion* [match-type]
      +--rw match-type      identityref
      +--rw values* [index]
        +--rw index        uint8
        +--rw value?       string
    +--rw ep-network-access-points
      +--rw ep-network-access-point* [network-access-id]
        +--rw network-access-id      string
        +--rw network-access-description?  string
        +--rw network-access-node-id?      string
        +--rw network-access-tp-id?        string
        +--rw network-access-tp-ip?        inet:host
        +--rw ep-rate-limit
          +--rw incoming-rate-limit?
            |         te-types:te-bandwidth
          +--rw outgoing-rate-limit?
            |         te-types:te-bandwidth
    +--rw ep-rate-limit
      +--rw incoming-rate-limit?  te-types:te-bandwidth
      +--rw outgoing-rate-limit?  te-types:te-bandwidth
    +--rw ep-protocol
    +--rw status
      +--rw admin-enabled?  boolean
      +--ro oper-status?    operational-type
    +--ro ep-monitoring
      +--ro incoming-utilized-bandwidth?
        |         te-types:te-bandwidth
      +--ro incoming-bw-utilization      decimal64
      +--ro outgoing-utilized-bandwidth?
        |         te-types:te-bandwidth
      +--ro outgoing-bw-utilization      decimal64
    +--rw ns-connections
      +--rw ns-connection* [ns-connection-id]
        +--rw ns-connection-id      uint32
        +--rw ns-connection-description?  string
        +--rw src
          | +--rw src-ep-id?  leafref
        +--rw dest
          | +--rw dest-ep-id?  leafref
        +--rw (ns-slo-sle-policy)?
          +--:(standard)
          | +--rw slo-sle-template?  leafref
          +--:(custom)
            +--rw slo-policy
              +--rw policy-description?  string
              +--rw ns-metric-bounds
                +--rw ns-metric-bound* [metric-type]
                  +--rw metric-type      identityref

```

```

|           |           +--rw metric-unit           string
|           |           +--rw value-description?    string
|           |           +--rw bound?                uint64
|           +--rw sle-policies
|           |   +--rw security-sle*                identityref
|           |   +--rw isolation?                    identityref
|           |   +--rw max-occupancy-level?          uint8
+--rw monitoring-type?                               ns-monitoring-type
+--ro ns-connection-monitoring
|   +--ro latency?          yang:gauge64
|   +--ro jitter?           yang:gauge32
|   +--ro loss-ratio?        decimal64

```

Figure 3

## 5. IETF Network Slice Templates

The 'ns-templates' container (Figure 3) is used by service provider of the NSC to define and maintain a set of common IETF Network Slice templates that apply to one or several IETF Network Slices. The exact definition of the templates is deployment specific to each network provider.

The model includes only the identifiers of SLO and SLE templates. When creation of IETF Network slice, the SLO and SLE policies can be easily identified.

The following shows an example where two network slice templates can be retrieved by the upper layer management system:

```
{
  "ietf-network-slices": {
    "ns-templates": {
      "slo-sle-template": [
        {
          "id": "GOLD-template",
          "template-description": "Two-way bandwidth: 1 Gbps,
            one-way latency 100ms "
          "sle-isolation": "ns-isolation-shared",
        },
        {
          "id": "PLATINUM-template",
          "template-description": "Two-way bandwidth: 1 Gbps,
            one-way latency 50ms "
          "sle-isolation": "ns-isolation-dedicated",
        },
      ],
    },
  }
}
```

## 6. IETF Network Slice Modeling Description

The 'ietf-network-slice' is the data structure that abstracts an IETF Network Slice of the IETF network. Each 'ietf-network-slice' is uniquely identified by an identifier: 'ns-id'.

An IETF Network Slice has the following main parameters:

- o "ns-id": Is an identifier that is used to uniquely identify the IETF Network Slice within NSC.
- o "ns-description": Gives some description of an IETF Network Slice service.
- o "ns-connectivity-type": Indicates the network connectivity type for the IETF Network Slice: Hub-and-Spoke, any-to-any, or custom type.
- o "status": Is used to show the operative and administrative status of the IETF Network Slice, and can be used as indicator to detect network slice anomalies.
- o "ns-tag": Is used to show the correlation between higher level function and the IETF network slices. If provided, this parameter may be used by IETF Network Slice Controller (NSC) during the realization. It may also be used by NSC for monitoring and assurance of the IETF network slices where NSC can notify the

higher system by issuing the notifications. It is noted that a single higher level customer might have multiple IETF Network Slices for a single application. This attribute may be used by NSC to also correlated multiple IETF network slices for a single application.

- o "ns-slo-sle-policy": Defines SLO and SLE policies for the "ietf-network-slice". More description are provided in Section 6.2

The "ns-endpoint" is an abstrac entity that represents a set of matching rules applied to an IETF network edge device or a customer network edge device involved in the IETF Network Slice and each 'ns-endpoint' belongs to a single 'ietf-network-slice'. More description are provided in Section 6.3

### 6.1. IETF Network Slice Connectivity Type

Based on the customer's traffic pattern requirements, an IETF Network Slice connection type could be point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), or multipoint-to-multipoint (MP2MP). The "ns-connectivity-type" under the node "ietf-network-slice" is used for this.

For the connectivity requirements, the model proposes to support any-to-any, Hub-and-Spoke (where Hubs can exchange traffic), and the custom. By default, the any-to-any is used. New connectivity type could be added via augmentation or by list of 'ns-connection' specified.

In addition, "ep-role" under the node "ns-endpoint" also needs to be defined, which specifies the role of the NSE in a particular Network Slice connectivity type. In the any-to-any, all NSEs MUST have the same role, which will be "any-to-any-role". In the Hub-and-Spoke, NSEs MUST have a Hub role or a Spoke role.

### 6.2. IETF Network Slice SLO and SLE Policy

As defined in [I-D.ietf-teas-ietf-network-slices], the SLO policy of an IETF Network Slice defines the minimum IETF Network Slice SLO attributes, and additional attributes can be added as needed.

"ns-slo-sle-policy" is used to represent specific SLO and SLE policies. During the creation of an IETF Network Slice, the policy can be specified either by a standard SLO and SLO template or a customized SLO and SLE policy.

The policy could both apply one per Network Slice or per connection 'ns-connection'.



The model allows multiple SLO and SLE attributes to be combined to meet different SLO and SLE requirements. For example, some NSs are used for video services and require high bandwidth, some NSs are used for key business services and request low latency and reliability, and some NSs need to provide connections for a large number of NSEs. That is, not all SLO or SLE attributes must be specified to meet the particular requirements of a slice.

"ns-metric-bounds" contains all these variations, which includes a list of "ns-metric-bound" and each "ns-metric-bound" could specify a particular "metric-type". "metric-type" is defined with YANG identity and the YANG module supports the following options:

"ns-slo-one-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two NSE. And the bandwidth is unidirectional.

"ns-slo-two-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two NSE. And the bandwidth is bidirectional.

"network-slice-slo-one-way-latency": Indicates the maximum one-way latency between two NSE.

"network-slice-slo-two-way-latency": Indicates the maximum round-trip latency between two NSE.

"ns-slo-one-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the one-way latency between sequential packets in a flow.

"ns-slo-two-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the two-way latency between sequential packets in a flow.

"ns-slo-one-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

"ns-slo-two-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

"ns-slo-availability": Is defined as the ratio of up-time to total\_time(up-time+down-time), where up-time is the time the IETF

Network Slice is available in accordance with the SLOs associated with it.

Some other Network Slice SLOs or SLEs could be extended when needed.

The following shows an example where a network slice policy can be configured:

```
{
  "ietf-network-slices": {
    "ietf-network-slice": {
      "slo-policy": {
        "policy-description": "video-service-policy",
        "ns-metric-bounds": {
          "ns-metric-bound": [
            {
              "metric-type": "ns-slo-one-way-bandwidth",
              "metric-unit": "mbps",
              "bound": "1000"
            },
            {
              "metric-type": "ns-slo-availability",
              "bound": "99.9%"
            }
          ],
        }
      }
    }
  }
}
```

### 6.3. IETF Network Slice Endpoint (NSE)

An IETF Network Slice Endpoint has several characteristics:

- o "ep-id": Uniquely identifies the NSE within Network Slice Controller (NSC). The identifier is a string that allows any encoding for the local administration of the IETF Network Slice.
- o "location": Indicates NSE location information that facilitates NSC easy identification of a NSE.
- o "ep-role": Represents a connectivity type role of a NSE belonging to an IETF network slice, as described in Section 6.1. The "ep-role" leaf defines the role of the endpoint in a particular NS connectivity type. In the any-to-any, all NSEs MUST have the same role, which will be "any-to-any-role".

- o "node-id": The NSE node information facilities NSC with easy identification of a NSE.
- o "ep-ip": The NSE IP information facilities NSC with easy identification of a NSE.
- o "ns-match-criteria": A matching policies to apply on a given NSE.
- o "ep-network-access-points": The list of the interfaces attached to an edge device of the IETF Network Slice by which the customer traffic is received.
- o "ep-rate-limit": Set the rate-limiting policies to apply on a given NSE, including ingress and egress traffic to ensure access security. When applied in the incoming direction, the rate-limit is applicable to the traffic from the NSE to the IETF scope Network that passes through the external interface. When Bandwidth is applied to the outgoing direction, it is applied to the traffic from the IETF Network to the NSE of that particular NS.
- o "ep-protocol": Specify the protocol for a NSE for exchanging control-plane information, e.g. L1 signaling protocol or L3 routing protocols, etc.
- o "status": Enable the control of the operative and administrative status of the NSE, can be used as indicator to detect NSE anomalies.

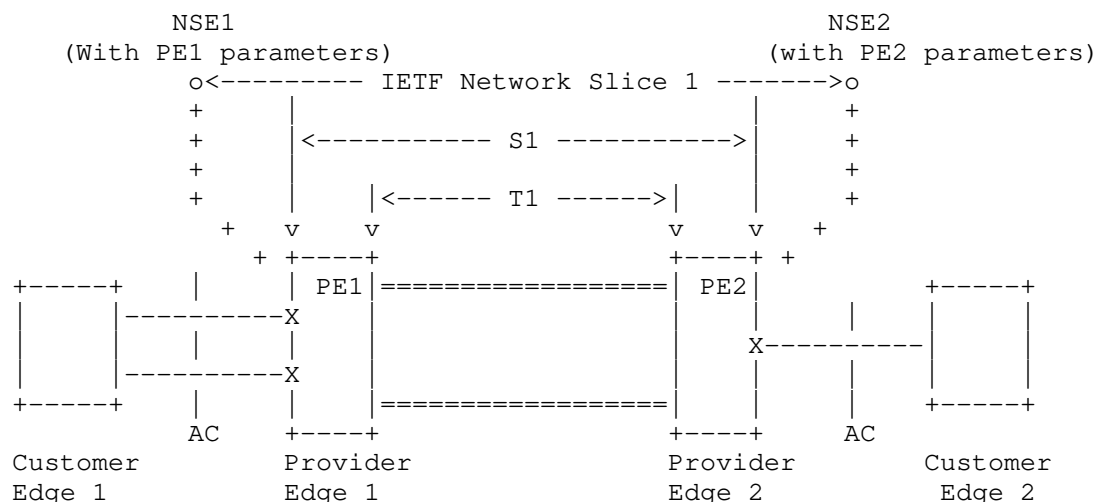
An NSE belong to a single IETF Network Slice. An IETF Network Slice involves two or more NSEs. An IETF Network Slice can be modified by adding new "ns-endpoint" or removing existing "ns-endpoint".

A NSE is used to define the matching rule on the customer traffic that can be injected to an IETF Network Slice. "network-slice-match-criteria" is defined to support different options. Classification can be based on many criteria, such as:

- o Physical interface: Indicates all the traffic received from the interface belongs to the IETF Network Slice.
- o Logical interface: For example, a given VLAN ID is used to identify an IETF Network Slice.
- o Encapsulation in the traffic header: For example, a source IP address is used to identify an IETF Network Slice.

To illustrate the use of NSE parameters, the below are two examples. How the NSC realize the mapping is out of scope for this document.

- o NSE mapping to PE example: As shown in Figure 4 , customer of the IETF network slice would like to connect two NSEs to satisfy specific service, e.g., Network wholesale services. In this case, the IETF network slice endpoints are mapped to physical interfaces of PE nodes. The IETF network slice controller (NSC) uses 'node-id' (PE device ID), 'ep-network-access-points' (Two PE interfaces ) to map the interfaces and corresponding services/tunnels/paths.



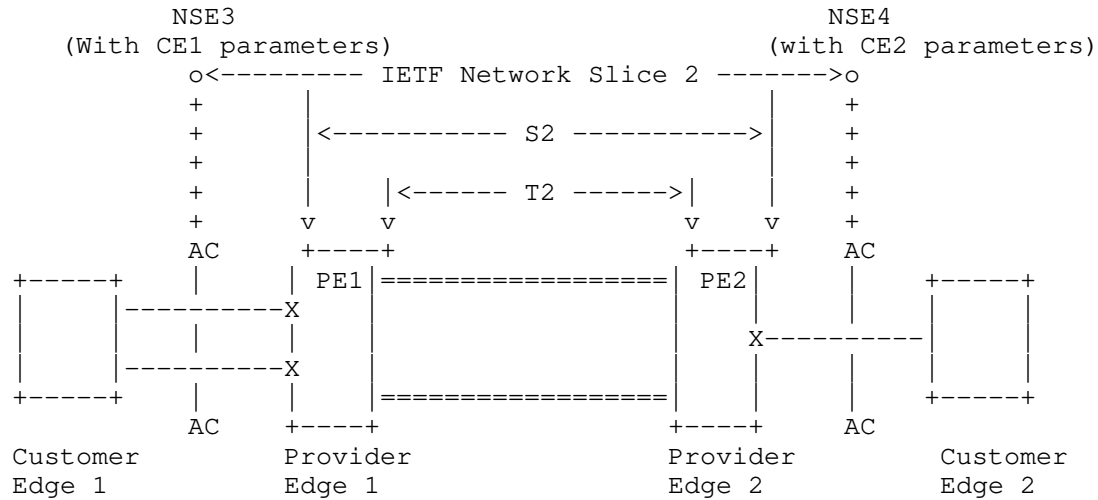
#### Legend:

- O: Representation of the IETF network slice endpoints (NSE)
- +: Mapping of NES to PE or CE nodes on IETF network
- X: Physical interfaces used for realization of IETF network slice
- S1: L0/L1/L2/L3 services used for realization of IETF network slice
- T1: Tunnels used for realization of IETF network slice

Figure 4

- o NSE mapping to CE-PE interface example: As shown in Figure 5 , customer of the IETF network slice would like to connect two NSEs to provide connectivity between transport portion of 5G RAN to 5G Core network functions. In this scenario, the IETF network slice endpoints (NSE) might be mapped to the respective PE-CE interface (see 3GPP TS 28.541 V17.1.0 section 6.3.17 EP\_Transport). The IETF network slice controller (NSC) uses 'node-id' (CE device ID), 'ep-ip' (CE tunnel endpoint IP), 'network-slice-match-criteria'

(VLAN interface), 'ep-network-access-points' (Two nexthop interfaces ) to map underlay services/tunnels/paths.



#### Legend:

- O: Representation of the IETF network slice endpoints (NSE)
- +: Mapping of NSE to PE or CE-PE interfaces on IETF network
- X: Physical interfaces used for realization of IETF network slice
- S2: L0/L1/L2/L3 services used for realization of IETF network slice
- T2: Tunnels used for realization of IETF network slice

Figure 5

## 7. IETF Network Slice Monitoring

An IETF Network Slice is a connectivity with specific SLO characteristics, including bandwidth, latency, etc. The connectivity is a combination of logical unidirectional connections, represented by 'ns-connection'.

This model also describes performance status of an IETF Network Slice. The statistics are described in the following granularity:

- o Per NS connection: specified in 'ns-connection-monitoring' under the "ns-connection"
- o Per NS Endpoint: specified in 'ep-monitoring' under the "ns-endpoint"

This model does not define monitoring enabling methods. The mechanism defined in [RFC8640] and [RFC8641] can be used for either periodic or on-demand subscription.

By specifying subtree filters or xpath filters to 'ns-connection' or 'ns-endpoint', so that only interested contents will be sent. These mechanisms can be used for monitoring the IETF Network Slice performance status so that the customer management system could initiate modification based on the IETF Network Slice running status.

## 8. IETF Network Slice NBI Module

The "ietf-network-slice" module uses types defined in [RFC6991], [RFC8776].

```
<CODE BEGINS> file "ietf-network-slice@2021-07-06.yang"
module ietf-network-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-slice";
  prefix ietf-ns;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Types.";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Types.";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering.";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group";
  contact
    "WG Web:  <https://tools.ietf.org/wg/teas/>
     WG List:  <mailto:teas@ietf.org>
     Editor: Bo Wu <lane.wubo@huawei.com>
           : Dhruv Dhody <dhruv.ietf@gmail.com>
           : Reza Rokui <reza.rokui@nokia.com>
           : Tarek Saad <tsaad@juniper.net>";
  description
```

"This module contains a YANG module for the IETF Network Slice.

Copyright (c) 2021 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2021-07-06 {
  description
    "initial version.";
  reference
    "RFC XXXX: A Yang Data Model for IETF Network Slice Operation";
}

/* Features */
/* Identities */

identity ns-isolation-type {
  description
    "Base identity for IETF Network slice isolation level.";
}

identity ns-isolation-shared {
  base ns-isolation-type;
  description
    "Shared resources (e.g. queues) are associated with the Network
    Slice traffic. Hence, the IETF network slice traffic can be
    impacted by effects of other services traffic sharing
    the same resources.";
}

identity ns-isolation-dedicated {
  base ns-isolation-type;
  description
    "Dedicated resources (e.g. queues) are associated with the Network
    Slice traffic. Hence, the IETF network slice traffic is isolated
    from other servceis traffic sharing the same resources.";
}

identity ns-security-type {
```

```
    description
      "Base identity for for IETF Network security level.";
  }

  identity ns-security-authenticate {
    base ns-security-type;
    description
      "IETF Network Slice requires authentication.";
  }

  identity ns-security-integrity {
    base ns-security-type;
    description
      "IETF Network Slice requires data integrity.";
  }

  identity ns-security-encryption {
    base ns-security-type;
    description
      "IETF Network Slice requires data encryption.";
  }

  identity ns-connectivity-type {
    description
      "Base identity for IETF Network Slice topology.";
  }

  identity any-to-any {
    base ns-connectivity-type;
    description
      "Identity for any-to-any IETF Network Slice topology.";
  }

  identity hub-spoke {
    base ns-connectivity-type;
    description
      "Identity for Hub-and-Spoke IETF Network Slice topology.";
  }

  identity custom {
    base ns-connectivity-type;
    description
      "Identity of a custom NS topology where Hubs can act as
        Spoke for certain parts of the network or Spokes as Hubs.";
  }

  identity endpoint-role {
    description
```



```
    "Base identity of a NSE role in an IETF Network Slice topology.";
}

identity any-to-any-role {
    base endpoint-role;
    description
        "Identity of any-to-any NS.";
}

identity spoke-role {
    base endpoint-role;
    description
        "A NSE is acting as a Spoke.";
}

identity hub-role {
    base endpoint-role;
    description
        "A NSE is acting as a Hub.";
}

identity custom-role {
    base endpoint-role;
    description
        "A NSE is custom role in the NS.";
}

identity ns-slo-metric-type {
    description
        "Base identity for IETF Network Slice SLO metric type.";
}

identity ns-slo-one-way-bandwidth {
    base ns-slo-metric-type;
    description
        "SLO bandwidth metric. Minimum guaranteed bandwidth between
        two endpoints at any time and is measured unidirectionally";
}

identity ns-slo-two-way-bandwidth {
    base ns-slo-metric-type;
    description
        "SLO bandwidth metric. Minimum guaranteed bandwidth between
        two endpoints at any time";
}

identity ns-slo-one-way-latency {
    base ns-slo-metric-type;
```

```
    description
      "SLO one-way latency is upper bound of network latency when
      transmitting between two endpoints. The metric is defined in
      RFC7679";
  }

  identity ns-slo-two-way-latency {
    base ns-slo-metric-type;
    description
      "SLO two-way latency is upper bound of network latency when
      transmitting between two endpoints. The metric is defined in
      RFC2681";
  }

  identity ns-slo-one-way-delay-variation {
    base ns-slo-metric-type;
    description
      "SLO one-way delay variation is defined by RFC3393, is the
      difference in the one-way delay between sequential packets
      between two endpoints.";
  }

  identity ns-slo-two-way-delay-variation {
    base ns-slo-metric-type;
    description
      "SLO two-way delay variation is defined by RFC5481, is the
      difference in the round-trip delay between sequential packets
      between two endpoints.";
  }

  identity ns-slo-one-way-packet-loss {
    base ns-slo-metric-type;
    description
      "SLO loss metric. The ratio of packets dropped to packets
      transmitted between two endpoints in one-way
      over a period of time as specified in RFC7680";
  }

  identity ns-slo-two-way-packet-loss {
    base ns-slo-metric-type;
    description
      "SLO loss metric. The ratio of packets dropped to packets
      transmitted between two endpoints in two-way
      over a period of time as specified in RFC7680";
  }

  identity ns-slo-availability {
    base ns-slo-metric-type;
```

```
    description
      "SLO availability level.";
  }

  identity ns-match-type {
    description
      "Base identity for IETF Network Slice traffic match type.";
  }

  identity ns-phy-interface-match {
    base ns-match-type;
    description
      "Use the physical interface as match criteria for the IETF
      Network Slice traffic.";
  }

  identity ns-vlan-match {
    base ns-match-type;
    description
      "Use the VLAN ID as match criteria for the IETF Network Slice
      traffic.";
  }

  identity ns-label-match {
    base ns-match-type;
    description
      "Use the MPLS label as match criteria for the IETF Network
      Slice traffic.";
  }

  /*
   * Identity for availability-type
   */

  identity availability-type {
    description
      "Base identity from which specific availability types are
      derived.";
  }

  identity level-1 {
    base availability-type;
    description
      "level 1: 99.9999%";
  }

  identity level-2 {
    base availability-type;
```

```
    description
      "level 2: 99.999%";
  }

  identity level-3 {
    base availability-type;
    description
      "level 3: 99.99%";
  }

  identity level-4 {
    base availability-type;
    description
      "level 4: 99.9%";
  }

  identity level-5 {
    base availability-type;
    description
      "level 5: 99%";
  }

  /* typedef */

  typedef operational-type {
    type enumeration {
      enum up {
        value 0;
        description
          "Operational status UP.";
      }
      enum down {
        value 1;
        description
          "Operational status DOWN.";
      }
      enum unknown {
        value 2;
        description
          "Operational status UNKNOWN.";
      }
    }
    description
      "This is a read-only attribute used to determine the
       status of a particular element.";
  }

  typedef ns-monitoring-type {
```

```
type enumeration {
  enum one-way {
    description
      "Represents one-way measurments monitoring type.";
  }
  enum two-way {
    description
      "represents two-way measurements monitoring type.";
  }
}
description
  "An enumerated type for monitoring on a IETF Network Slice
  connection.";
}

/* Groupings */

grouping status-params {
  description
    "A grouping used to join operational and administrative status.";
  container status {
    description
      "A container for the administrative and operational state.";
    leaf admin-enabled {
      type boolean;
      description
        "The administrative status.";
    }
    leaf oper-status {
      type operational-type;
      config false;
      description
        "The operational status.";
    }
  }
}

grouping ns-match-criteria {
  description
    "A grouping for the IETF Network Slice match definition.";
  container ns-match-criteria {
    description
      "Describes the IETF Network Slice match criteria.";
    list ns-match-criterion {
      key "match-type";
      description
        "List of the IETF Network Slice traffic match criteria.";
      leaf match-type {
```

```
    type identityref {
      base ns-match-type;
    }
    description
      "Identifies an entry in the list of the IETF Network Slice
      match criteria.";
  }
  list values {
    key "index";
    description
      "List of match criteria values.";
    leaf index {
      type uint8;
      description
        "Index of an entry in the list.";
    }
    leaf value {
      type string;
      description
        "Describes the IETF Network Slice match criteria, e.g.
        IP address, VLAN, etc.";
    }
  }
}

grouping ns-connection-group-metric-bounds {
  description
    "Grouping of Network Slice metric bounds that
    are shared amongst multiple connections of a Network
    Slice.";
  leaf ns-slo-shared-bandwidth {
    type te-types:te-bandwidth;
    description
      "A limit on the bandwidth that is shared amongst
      multiple connections of an IETF Network Slice.";
  }
}

grouping ns-sles {
  description
    "Indirectly Measurable Objectives of a IETF Network
    Slice.";
  container sle-policies {
    description
      "Container for the policy of SLEs applicable to
      IETF Network Slice.";
  }
}
```

```
    leaf-list security-sle {
      type identityref {
        base ns-security-type;
      }
      description
        "The IETF Network Slice security SLE(s)";
    }
    leaf isolation {
      type identityref {
        base ns-isolation-type;
      }
      default "ns-isolation-shared";
      description
        "The IETF Network Slice isolation SLE requirement.";
    }
    leaf max-occupancy-level {
      type uint8 {
        range "1..100";
      }
      description
        "The maximal occupancy level specifies the number of flows to
        be admitted.";
    }
  }
}

grouping ns-metric-bounds {
  description
    "IETF Network Slice metric bounds grouping.";
  container ns-metric-bounds {
    description
      "IETF Network Slice metric bounds container.";
    list ns-metric-bound {
      key "metric-type";
      description
        "List of IETF Network Slice metric bounds.";
      leaf metric-type {
        type identityref {
          base ns-slo-metric-type;
        }
        description
          "Identifies an entry in the list of metric type
          bounds for the IETF Network Slice.";
      }
      leaf metric-unit {
        type string;
        mandatory true;
        description

```

```
        "The metric unit of the parameter. For example,
          s, ms, ns, and so on.";
    }
    leaf value-description {
        type string;
        description
            "The description of previous value. ";
    }
    leaf bound {
        type uint64;
        default "0";
        description
            "The Bound on the Network Slice connection metric. A
             zero indicate an unbounded upper limit for the
             specific metric-type.";
    }
}
}
}

grouping ep-network-access-points {
    description
        "Grouping for the endpoint network access definition.";
    container ep-network-access-points {
        description
            "List of network access points.";
        list ep-network-access-point {
            key "network-access-id";
            description
                "The IETF Network Slice network access points
                 related parameters.";
            leaf network-access-id {
                type string;
                description
                    "Uniquely identifier a network access point.";
            }
            leaf network-access-description {
                type string;
                description
                    "The network access point description.";
            }
            leaf network-access-node-id {
                type string;
                description
                    "The network access point node ID in the case of
                     multi-homing.";
            }
            leaf network-access-tp-id {
```



```
        type string;
        description
            "The termination port ID of the EP network access
            point.";
    }
    leaf network-access-tp-ip {
        type inet:host;
        description
            "The IP address of the EP network access point.";
    }
    /* Per ep-network-access-point rate limits */
    uses ns-rate-limit;
}
}

grouping endpoint-monitoring-parameters {
    description
        "Grouping for the endpoint monitoring parameters.";
    container ep-monitoring {
        config false;
        description
            "Container for endpoint monitoring parameters.";
        leaf incoming-utilized-bandwidth {
            type te-types:te-bandwidth;
            description
                "Incoming bandwidth utilization at an endpoint.";
        }
        leaf incoming-bw-utilization {
            type decimal64 {
                fraction-digits 5;
                range "0..100";
            }
            units "percent";
            mandatory true;
            description
                "To be used to define the bandwidth utilization
                as a percentage of the available bandwidth.";
        }
        leaf outgoing-utilized-bandwidth {
            type te-types:te-bandwidth;
            description
                "Outgoing bandwidth utilization at an endpoint.";
        }
        leaf outgoing-bw-utilization {
            type decimal64 {
                fraction-digits 5;
                range "0..100";
            }
        }
    }
}
```

```
    }
    units "percent";
    mandatory true;
    description
        "To be used to define the bandwidth utilization
         as a percentage of the available bandwidth.";
    }
}

grouping common-monitoring-parameters {
    description
        "Grouping for link-monitoring-parameters.";
    leaf latency {
        type yang:gauge64;
        units "usec";
        description
            "The latency statistics per Network Slice connection.
             RFC2681 and RFC7679 discuss round trip times and one-way
             metrics, respectively";
    }
    leaf jitter {
        type yang:gauge32;
        description
            "The jitter statistics per Network Slice member
             as defined by RFC3393.";
    }
    leaf loss-ratio {
        type decimal64 {
            fraction-digits 6;
            range "0 .. 50.331642";
        }
        description
            "Packet loss as a percentage of the total traffic
             sent over a configurable interval. The finest precision is
             0.000003%. where the maximum 50.331642%.";
        reference
            "RFC 7810, section-4.4";
    }
}

grouping geolocation-container {
    description
        "A grouping containing a GPS location.";
    container location {
        description
            "A container containing a GPS location.";
        leaf altitude {
```

```
        type int64;
        units "millimeter";
        description
            "Distance above the sea level.";
    }
    leaf latitude {
        type decimal64 {
            fraction-digits 8;
            range "-90..90";
        }
        description
            "Relative position north or south on the Earth's surface.";
    }
    leaf longitude {
        type decimal64 {
            fraction-digits 8;
            range "-180..180";
        }
        description
            "Angular distance east or west on the Earth's surface.";
    }
}
// gps-location
}

// geolocation-container
grouping ns-rate-limit {
    description
        "The Network Slice rate limit grouping.";
    container ep-rate-limit {
        description
            "Container for the asymmetric traffic control";
        leaf incoming-rate-limit {
            type te-types:te-bandwidth;
            description
                "The rate-limit imposed on incoming traffic.";
        }
        leaf outgoing-rate-limit {
            type te-types:te-bandwidth;
            description
                "The rate-limit imposed on outgoing traffic.";
        }
    }
}

grouping endpoint {
    description
```

```
    "IETF Network Slice endpoint related information";
  leaf ep-id {
    type string;
    description
      "unique identifier for the referred IETF Network
       Slice endpoint";
  }
  leaf ep-description {
    type string;
    description
      "endpoint name";
  }
  leaf ep-role {
    type identityref {
      base endpoint-role;
    }
    default "any-to-any-role";
    description
      "Role of the endpoint in the IETF Network Slice.";
  }
  uses geolocation-container;
  leaf node-id {
    type string;
    description
      "Uniquely identifies an edge node within the IETF slice
       network.";
  }
  leaf ep-ip {
    type inet:host;
    description
      "The address of the endpoint IP address.";
  }
  uses ns-match-criteria;
  uses ep-network-access-points;
  uses ns-rate-limit;
  /* Per NSE rate limits */
  container ep-protocol {
    description
      "Describes protocol for the Network Slice Endpoint.";
  }
  uses status-params;
  uses endpoint-monitoring-parameters;
}

//ns-endpoint

grouping ns-connection {
  description
```

```
    "The Network Slice connection is described in this container.";
  leaf ns-connection-id {
    type uint32;
    description
      "The Network Slice connection identifier";
  }
  leaf ns-connection-description {
    type string;
    description
      "The Network Slice connection description";
  }
  container src {
    description
      "the source of Network Slice link";
    leaf src-ep-id {
      type leafref {
        path "/network-slices/network-slice"
          + "/ns-endpoints/ns-endpoint/ep-id";
      }
      description
        "reference to source Network Slice endpoint";
    }
  }
  container dest {
    description
      "the destination of Network Slice link ";
    leaf dest-ep-id {
      type leafref {
        path "/network-slices/network-slice"
          + "/ns-endpoints/ns-endpoint/ep-id";
      }
      description
        "reference to dest Network Slice endpoint";
    }
  }
  uses ns-slo-sle-policy;
  /* Per connection ns-slo-sle-policy overrides
   * the per network slice ns-slo-sle-policy.
   */
  leaf monitoring-type {
    type ns-monitoring-type;
    description
      "One way or two way monitoring type.";
  }
  container ns-connection-monitoring {
    config false;
    description
      "SLO status Per network-slice endpoint to endpoint ";
  }
```

```
    uses common-monitoring-parameters;
  }
}

//ns-connection

grouping slice-template {
  description
    "Grouping for slice-templates.";
  container ns-slo-sle-templates {
    description
      "Contains a set of network slice templates to
       reference in the IETF network slice.";
    list ns-slo-sle-template {
      key "id";
      leaf id {
        type string;
        description
          "Identification of the Service Level Objective (SLO)
           and Service Level Expectation (SLE) template to be used.
           Local administration meaning.";
      }
      leaf template-description {
        type string;
        description
          "Description of the SLO & SLE policy template.";
      }
      description
        "List for SLO and SLE template identifiers.";
    }
  }
}

/* Configuration data nodes */

grouping ns-slo-sle-policy {
  description
    "Network Slice policy grouping.";
  choice ns-slo-sle-policy {
    description
      "Choice for SLO and SLE policy template.
       Can be standard template or customized template.";
    case standard {
      description
        "Standard SLO template.";
      leaf slo-sle-template {
        type leafref {
          path "/network-slices"
        }
      }
    }
  }
}
```

```
        + "/ns-slo-sle-templates/ns-slo-sle-template/id";
    }
    description
        "Standard SLO and SLE template to be used.";
}
}
case custom {
    description
        "Customized SLO template.";
    container slo-policy {
        description
            "Contains the SLO policy.";
        leaf policy-description {
            type string;
            description
                "Description of the SLO policy.";
        }
        uses ns-metric-bounds;
    }
    uses ns-sles;
}
}
}

container network-slices {
    description
        "IETF network-slice configurations";
    uses slice-template;
    list network-slice {
        key "ns-id";
        description
            "a network-slice is identified by a ns-id";
        leaf ns-id {
            type string;
            description
                "A unique network-slice identifier across an IETF NSC ";
        }
        leaf ns-description {
            type string;
            description
                "Give more description of the network slice";
        }
        leaf-list ns-tag {
            type string;
            description
                "Network Slice tag for operational management";
        }
        leaf ns-connectivity-type {
```

```
    type identityref {
      base ns-connectivity-type;
    }
    default "any-to-any";
    description
      "Network Slice topology.";
  }
  uses ns-slo-sle-policy;
  uses status-params;
  container ns-endpoints {
    description
      "Endpoints";
    list ns-endpoint {
      key "ep-id";
      uses endpoint;
      description
        "list of endpoints in this slice";
    }
  }
  container ns-connections {
    description
      "Connections container";
    list ns-connection {
      key "ns-connection-id";
      description
        "List of Network Slice connections.";
      uses ns-connection;
    }
  }
}
//ietf-network-slice list
}
```

<CODE ENDS>

## 9. Security Considerations

The YANG module defined in this document is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a



preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

o /ietf-network-slice/network-slices/network-slice

The entries in the list above include the whole network configurations corresponding with the slice which the higher management system requests, and indirectly create or modify the PE or P device configurations. Unexpected changes to these entries could lead to service disruption and/or network misbehavior.

## 10. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-network-slice  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

This document requests to register a YANG module in the YANG Module Names registry [RFC7950].

Name: ietf-network-slice  
Namespace: urn:ietf:params:xml:ns:yang:ietf-network-slice  
Prefix: ietf-ns  
Reference: RFC XXXX

## 11. Acknowledgments

The authors wish to thank Sergio Belotti, Qin Wu, Susan Hares, Eric Grey, and many other NS DT members for their helpful comments and suggestions.

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.

- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", RFC 8640, DOI 10.17487/RFC8640, September 2019, <<https://www.rfc-editor.org/info/rfc8640>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.

## 12.2. Informative References

- [I-D.geng-teas-network-slice-mapping]  
Geng, X., Dong, J., Pang, R., Han, L., Niwa, T., Jin, J., Liu, C., and N. Nageshar, "5G End-to-end Network Slice Mapping from the view of Transport Network", draft-geng-teas-network-slice-mapping-03 (work in progress), February 2021.
- [I-D.ietf-teas-actn-vn-yang]  
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Y. Yoon, "A YANG Data Model for VN Operation", draft-ietf-teas-actn-vn-yang-11 (work in progress), February 2021.
- [I-D.ietf-teas-ietf-network-slices]  
Farrel, A., Gray, E., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", draft-ietf-teas-ietf-network-slices-00 (work in progress), April 2021.

[I-D.liu-teas-transport-network-slice-yang]

Liu, X., Tantsura, J., Bryskin, I., Contreras, L. M., Wu, Q., Belotti, S., and R. Rokui, "IETF Network Slice YANG Data Model", draft-liu-teas-transport-network-slice-yang-02 (work in progress), November 2020.

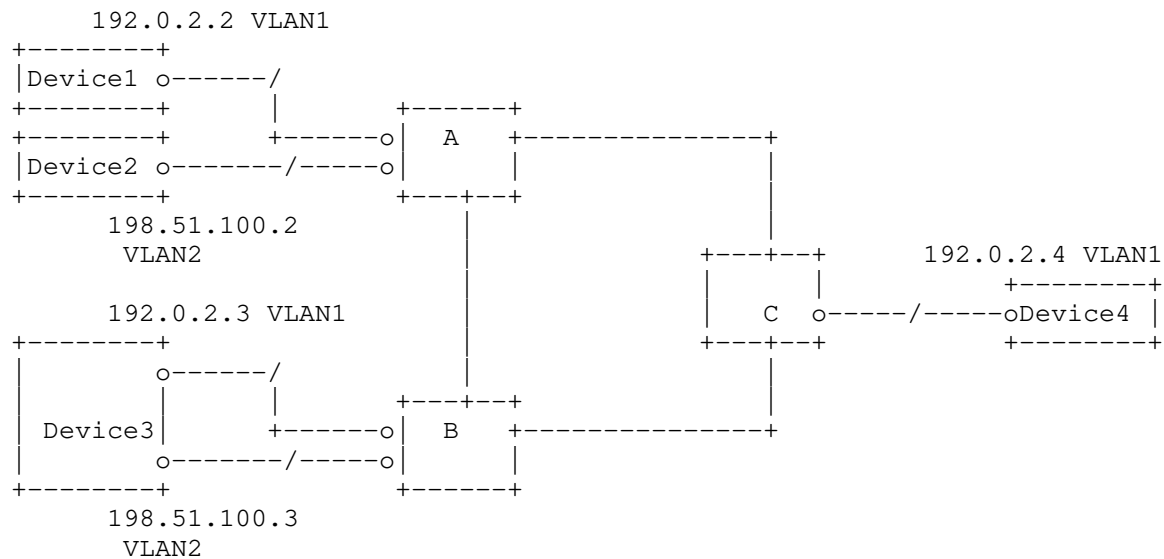
[RFC8309]

Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.

## Appendix A. IETF Network Slice NBI Model Usage Example

The following example describes a simplified service configuration of two IETF Network slice instances:

- o IETF Network Slice 1 on Device1, Device3, and Device4, with any-to-any connectivity type
- o IETF Network Slice 2 on Device2, Device3, with any-to-any connectivity type



POST: /restconf/data/ietf-network-slice:ietf-network-slices

Host: example.com

Content-Type: application/yang-data+json

```
{
  "network-slices":{
    "network-slice":[
      {
```

```
"ns-id":"1",
"ns-description":"slice1",
"ns-connectivity-type":"any-to-any",
"ns-endpoints":{
  "ns-endpoint":[
    {
      "ep-id":"11",
      "ep-description":"slice1 ep1 connected to device 1",
      "ep-role":"any-to-any-role",
      "ns-match-criteria":[
        {
          "match-type":"ns-vlan-match",
          "value":[
            {
              "index":"1",
              "value":"1"
            }
          ]
        }
      ]
    }
  ],
  {
    "ep-id":"12",
    "ep-description":"slice1 ep2 connected to device 3",
    "ep-role":"any-to-any-role",
    "ns-match-criteria":[
      {
        "match-type":"ns-vlan-match",
        "value":[
          {
            "index":"1",
            "value":"20"
          }
        ]
      }
    ]
  }
],
  {
    "ep-id":"13",
    "ep-description":"slice1 ep3 connected to device 4",
    "ep-role":"any-to-any-role",
    "ns-match-criteria":[
      {
        "match-type":"ns-vlan-match",
        "value":[
          {
            "index":"1",
            "value":"1"
          }
        ]
      }
    ]
  }
]
```

```

    }
  ]
}
],
{
  "ns-id":"ns2",
  "ns-description":"slice2",
  "ns-connectivity-type":"any-to-any",
  "ns-endpoints":{
    "ns-endpoint":[
      {
        "ep-id":"21",
        "ep-description":"slice2 ep1 connected to device 2",
        "ep-role":"any-to-any-role",
        "ns-match-criteria":[
          {
            "match-type":"ns-vlan-match",
            "value":[
              {
                "index":"1",
                "value":"2"
              }
            ]
          }
        ]
      }
    ]
  },
  {
    "ep-id":"22",
    "ep-description":"slice2 ep2 connected to device 3",
    "ep-role":"any-to-any-role",
    "ns-match-criteria":[
      {
        "match-type":"ns-vlan-match",
        "value":[
          {
            "index":"1",
            "value":"2"
          }
        ]
      }
    ]
  }
]
}

```

```

    }
  ]
}

```

## Appendix B. Comparison with Other Possible Design choices for IETF Network Slice NBI

According to the 3.3.1. Northbound Interface (NBI) [I-D.ietf-teas-ietf-network-slices], the IETF Network Slice NBI is a technology-agnostic interface, which is used for a customer to express requirements for a particular IETF Network Slice. Customers operate on abstract IETF Network Slices, with details related to their realization hidden. As classified by [RFC8309], the IETF Network Slice NBI is classified as Customer Service Model.

This draft analyzes the following existing IETF models to identify the gap between the IETF Network Slice NBI requirements.

### B.1. ACTN VN Model Augmentation

The difference between the ACTN VN model and the IETF Network Slice NBI requirements is that the IETF Network Slice NBI is a technology-agnostic interface, whereas the VN model is bound to the IETF TE Topologies. The realization of the IETF Network Slice does not necessarily require the slice network to support the TE technology.

The ACTN VN (Virtual Network) model introduced in [I-D.ietf-teas-actn-vn-yang] is the abstract customer view of the TE network. Its YANG structure includes four components:

- o VN: A Virtual Network (VN) is a network provided by a service provider to a customer for use and two types of VN has defined. The Type 1 VN can be seen as a set of edge-to-edge abstract links. Each link is an abstraction of the underlying network which can encompass edge points of the customer's network, access links, intra-domain paths, and inter-domain links.
- o AP: An AP is a logical identifier used to identify the access link which is shared between the customer and the IETF scoped Network.
- o VN-AP: A VN-AP is a logical binding between an AP and a given VN.
- o VN-member: A VN-member is an abstract edge-to-edge link between any two APs or VN-APs. Each link is formed as an E2E tunnel across the underlying networks.

The Type 1 VN can be used to describe IETF Network Slice connection requirements. However, the Network Slice SLO and Network Slice Endpoint are not clearly defined and there's no direct equivalent. For example, the SLO requirement of the VN is defined through the IETF TE Topologies YANG model, but the TE Topologies model is related to a specific implementation technology. Also, VN-AP does not define "network-slice-match-criteria" to specify a specific NSE belonging to an IETF Network Slice.

## B.2. RFC8345 Augmentation Model

The difference between the IETF Network Slice NBI requirements and the IETF basic network model is that the IETF Network Slice NBI requests abstract customer IETF Network Slices, with details related to the slice Network hidden. But the IETF network model is used to describe the interconnection details of a Network. The customer service model does not need to provide details on the Network.

For example, IETF Network Topologies YANG data model extension introduced in Transport Network Slice YANG Data Model [I-D.liu-teas-transport-network-slice-yang] includes three major parts:

- o Network: a transport network list and an list of nodes contained in the network
- o Link: "links" list and "termination points" list describe how nodes in a network are connected to each other
- o Support network: vertical layering relationships between IETF Network Slice networks and underlay networks

Based on this structure, the IETF Network Slice-specific SLO attributes nodes are augmented on the Network Topologies model,, e.g. isolation etc. However, this modeling design requires the slice network to expose a lot of details of the network, such as the actual topology including nodes interconnection and different network layers interconnection.

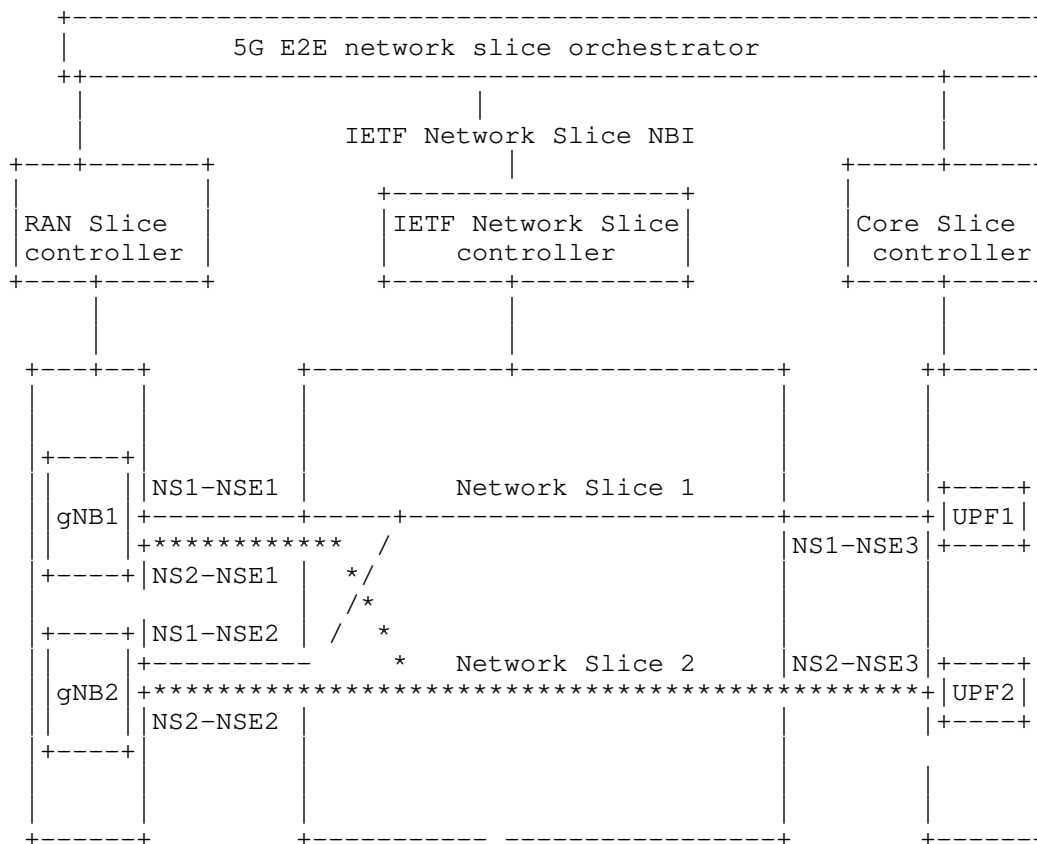
## Appendix C. Appendix B IETF Network Slice Match Criteria

5G is a use case of the IETF Network Slice and 5G End-to-end Network Slice Mapping from the view of IETF Network [I-D.geng-teas-network-slice-mapping]

defines two types of Network Slice interconnection and differentiation methods: by physical interface or by TNSII (Transport Network Slice Interworking Identifier). TNSII is a field in the



packet header when different 5G wireless network slices are transported through a single physical interfaces of the IETF scoped Network. In the 5G scenario, "network-slice-match-criteria" refers to TNSII.



As shown in the figure, gNodeB 1 and gNodeB 2 use IP gNB1 and IP gNB2 to communicate with the IETF network, respectively. In addition, the traffic of NS1 and NS2 on gNodeB 1 and gNodeB 2 is transmitted through the same access links to the IETF slice network. The IETF slice network need to to distinguish different IETF Network Slice traffic of same gNB. Therefore, in addition to using "node-id" and "ep-ip" to identify a Network Slice Endpoint, other information is needed along with these parameters to uniquely distinguish a NSE. For example, VLAN IDs in the user traffic can be used to distinguish the NSEs of gNBs and UPFs.

## Authors' Addresses

Bo Wu  
Huawei Technologies  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: lana.wubo@huawei.com

Dhruv Dhody  
Huawei Technologies  
Divyashree Techno Park  
Bangalore, Karnataka 560066  
India

Email: dhruv.ietf@gmail.com

Reza Rokui  
Nokia

Email: reza.rokui@nokia.com

Tarek Saad  
Juniper Networks

Email: tsaad@juniper.net

Liuyan Han  
China Mobile

Email: hanliuyan@chinamobile.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 30 March 2022

B. Wu  
D. Dhody  
Huawei Technologies  
R. Rokui  
Nokia  
T. Saad  
Juniper Networks  
L. Han  
China Mobile  
L.M. Contreras  
Telefonica  
26 September 2021

IETF Network Slice Service YANG Model  
draft-wd-teas-ietf-network-slice-nbi-yang-05

Abstract

This document provides a YANG data model for the IETF Network Slice service. The model can be used by a IETF Network Slice Customer to manage IETF Network Slice from an IETF Network Slice Controller (NSC).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 30 March 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 2  |
| 2. Conventions used in this document . . . . .                     | 3  |
| 2.1. Tree Diagrams . . . . .                                       | 4  |
| 3. IETF Network Slice Service Model Usage . . . . .                | 4  |
| 4. Background on IETF Network Slice Service Modeling . . . . .     | 5  |
| 4.1. LxSM VPN Service Models . . . . .                             | 5  |
| 4.2. ACTN VN Model Augmentation analysis . . . . .                 | 5  |
| 5. IETF Network Slice Service Model Overview . . . . .             | 8  |
| 6. IETF Network Slice Templates . . . . .                          | 12 |
| 7. IETF Network Slice Modeling Description . . . . .               | 12 |
| 7.1. IETF Network Slice Connectivity Type . . . . .                | 13 |
| 7.2. IETF Network Slice SLO and SLE Policy . . . . .               | 14 |
| 7.3. IETF Network Slice Endpoint (NSE) . . . . .                   | 16 |
| 8. IETF Network Slice Monitoring . . . . .                         | 19 |
| 9. IETF Network Slice Service Module . . . . .                     | 20 |
| 10. Security Considerations . . . . .                              | 40 |
| 11. IANA Considerations . . . . .                                  | 41 |
| 12. Acknowledgments . . . . .                                      | 41 |
| 13. References . . . . .   | 41 |
| 13.1. Normative References . . . . .                               | 41 |
| 13.2. Informative References . . . . .                             | 43 |
| Appendix A. IETF Network Slice NBI Model Usage Example . . . . .   | 44 |
| Appendix B. Appendix B IETF Network Slice Match Criteria . . . . . | 47 |
| Authors' Addresses . . . . .                                       | 48 |

## 1. Introduction

This document provides a YANG [RFC7950] data model for the IETF Network Slice service.

The YANG model discussed in this document is defined based on the description of the IETF Network Slice in [I-D.ietf-teas-ietf-network-slices], which is used to operate IETF Network Slices during the IETF Network Slice instantiation. This YANG model supports various operations on IETF Network Slices such as creation, modification, deletion, and monitoring.

The IETF Network Slice Controller (NSC) is a logical entity that allows customers to manage IETF network slices. Details related to the realization of IETF network slices that fulfil the request are internal to the entity that operates the network. Such details are deployment- and implementation-specific.

The NSC receives request from its customer-facing interface (e.g., from a management system). This interface carries data objects the IETF network slice user provides, describing the needed IETF network slices in terms of topology, target service level objectives (SLO), and also monitoring and reporting requirements. These requirements are then translated into technology-specific actions that are implemented in the underlying network using a network-facing interface. The details of IETF network slices realization are out of scope for this document.

The YANG model discussed in this document describes the requirements of an IETF Network Slice from the point of view of the customer. It is thus classified as customer service model in [RFC8309].

The IETF Network Slice operational state is included in the same tree as the configuration consistent with Network Management Datastore Architecture [RFC8342].

## 2. Conventions used in this document

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC6241] and are used in this specification:

- \* client
- \* configuration data
- \* state data

This document makes use of the terms defined in [RFC7950].

This document also makes use of the terms introduced in the Framework for IETF Network Slices [I-D.ietf-teas-ietf-network-slices]:

This document defines the following term:

- \* IETF Network Slice Connection (NS-Connection): In the context of an IETF Network Slice, an IETF NS-Connection is an abstract entity which represents a particular connection between a pair of NSEs. An IETF Network Slice can has one or multiple NS-Connections.

### 2.1. Tree Diagrams

The tree diagram used in this document follow the notation defined in [RFC8340].

## 3. IETF Network Slice Service Model Usage

The intention of the IETF Network Slice service model is to allow the customer to manage IETF Network Slices. In particular, the model allows customers to operate in an abstract and technology-agnostic manner, with details of the IETF Network Slices realization hidden.

According to the [I-D.ietf-teas-ietf-network-slices] description, IETF Network Slices are applicable to use cases such as (but not limited to) network wholesale services, network infrastructure sharing among operators, NFV connectivity, Data Center Interconnect, and 5G E2E network slice.

As shown in Figure 1, in all these use-cases, the model is used by the higher management system to communicate with NSC for life cycle manage of IETF Network Slices including both enablement and monitoring. The interface is used to support dynamic IETF Network Slice creation and its lifecycle management to facilitate end-to-end network slice services.

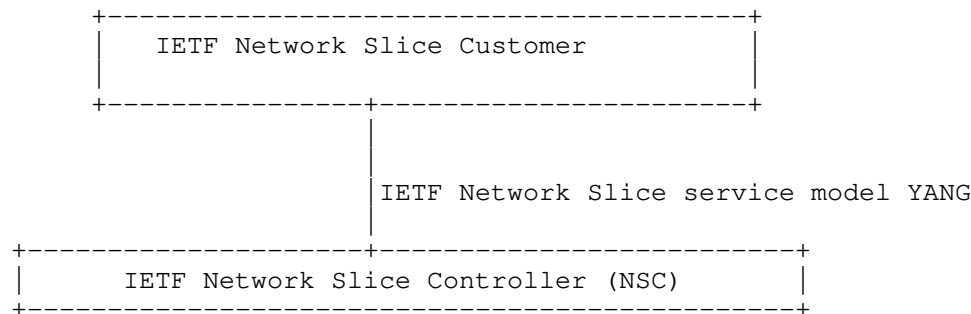


Figure 1: IETF Network Slice Service Reference Architecture

#### 4. Background on IETF Network Slice Service Modeling

[I-D.ietf-teas-ietf-network-slices] defines the IETF Network Slice service model as a technology agnostic interface. That is, customer expresses requirements for a particular slice by specifying what is required rather than how that is to be achieved.

This section explains why a new YANG service data model is proposed for support of IETF network slice services. The following data models are considered:

- \* L3SM, L2SM and L1CSM models
- \* ACTN VN model

##### 4.1. LxSM VPN Service Models

Currently, the three VPN service models defined at IETF are L3SM [RFC8299], L2SM [RFC8466], L1CSM [I-D.ietf-ccamp-llcsm-yang]. These models are related to specific VPN technologies. When using these models as a slicing service interface, customers need to be aware of the network's VPN technology so that right interfaces can be used.

The IETF network slice service requires a technology agnostic interface (similar to intent), to avoiding using multiple VPN models or other technology specific models.

##### 4.2. ACTN VN Model Augmentation analysis

Abstraction and Control of TE Networks (ACTN - [RFC8453]) defines a virtual network (VN) service [I-D.ietf-teas-actn-vn-yang]. Figure 2 shows that the relationship of IETF network slice and ACTN framework.

ACTN VN is independent of VPN technologies, and relays on traffic engineering YANG model [RFC8795] to define VN service in terms of a topology with a single abstract node and its connectivity matrix.

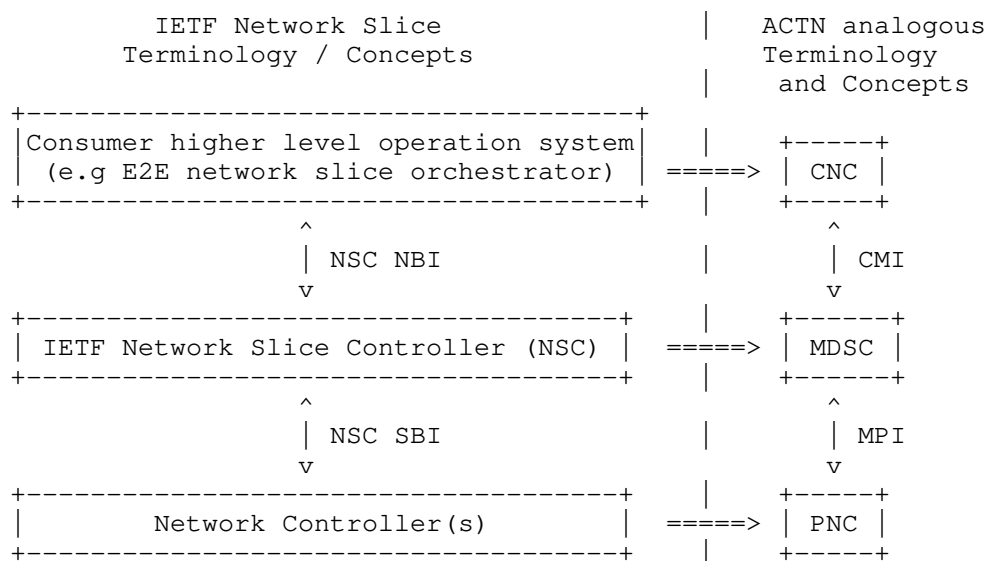


Figure 2: ACTN mapping

The ACTN VN model introduced in[I-D.ietf-teas-actn-vn-yang] is an abstract customer view of the TE network. Its YANG structure includes four components:

- \* VN: A Virtual Network (VN) is a network provided by a service provider to a customer for use and two types of VN has defined. The Type 1 VN can be seen as a set of edge-to-edge abstract links between VNAP.
- \* AP: An AP is a logical identifier used to identify the access link which is shared between the customer and the IETF scoped Network.
- \* VN-AP: A VN-AP is a logical binding between an AP and a given VN.
- \* VN-member: A VN-member is an abstract edge-to-edge link between any two APs or VN-APs.

Figure 3 illustrates the difference between AP/VNAPs in a VN and NSEs of IETF network slice. Though AP is a logical identifier, it maps to a access link between the customer nodes and provider nodes, which is also TP (Termination Port) of the provider node). When the access link changes, the VN connection matrix changes accordingly. For example, when the backup link of AP5/TP5 is added, the corresponding VN members, AP5-AP3 and AP5-AP4, also need to be added. These changes are underlying topology details. The slice service focuses on the connection matrix between C1, C2, and C3.



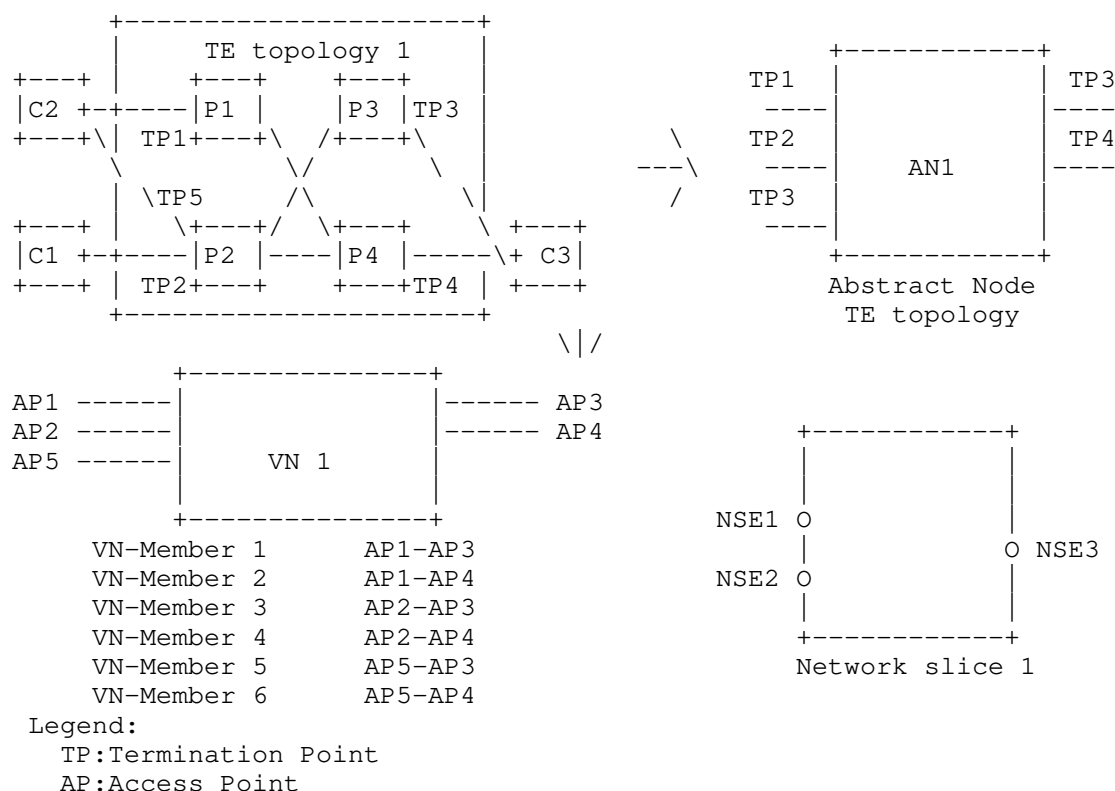
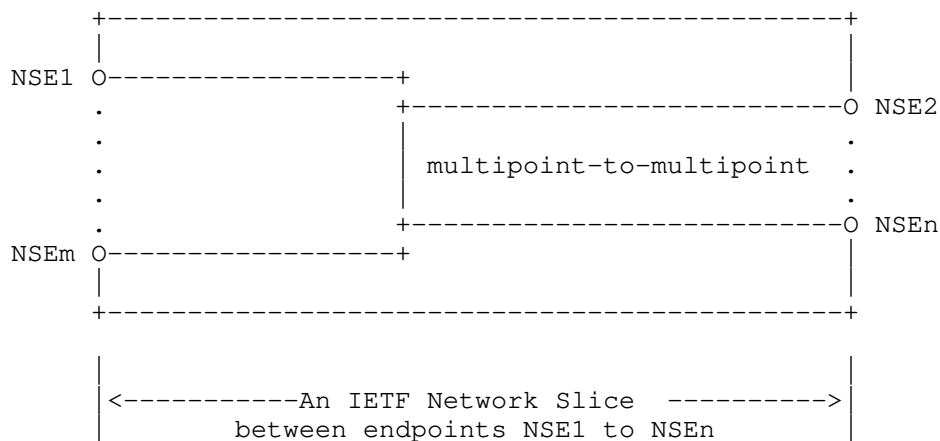


Figure 3: Difference between AP and NSE

In summary, the ACTN VN model cannot be used to model the IETF network slice service model because the VN model is tightly bound to the IETF TE Topology model and the constraints are buried deep inside the TE topology connectivity matrix and thus does not provide a clear mechanism to specify SLO/SLE of IETF network slice. The IETF network slice endpoint also does not ascribe to the concept of AP/VNAP. The realization of the IETF Network Slice does not necessarily require the slice network to support the TE technology. As the IETF network slice could be realized with non-TE techniques (FlexAlgo, MT). Reusing or augmenting VN model is problematic.

## 5. IETF Network Slice Service Model Overview

As defined in [I-D.ietf-teas-ietf-network-slices], an IETF Network Slice is a logical network topology connecting a number of endpoints using a set of shared or dedicated network resources that are used to satisfy specific service requirements. The logical topology types are: point-to-point, point-to-multipoint, multipoint-to-point, or multipoint-to-multipoint. The endpoints are conceptual points that could map to a device, application or a network function. And the specific service requirements, typically expressed as bandwidth, latency, latency variation, and other desired or required characteristics, such as security, MTU, traffic-type (e.g., IPv4, IPv6, Ethernet or unstructured) or a higher-level behavior to process traffic according to user-application (which may be realized using network function). An example of an IETF network slice is shown in Figure 4 .



Legend:

NSE: IETF Network Slice Endpoint

O: Represents IETF Network Slice Endpoints

Figure 4: An IETF Network Slice Example

As shown in the example, an IETF network slice may have multiple NSEs. The NSEs are the ingress/egress points where traffic enters/exits the IETF network slice. As the edge of the IETF network slice, the NSEs also delimit a topological network portion within which the committed SLOs apply.

When an NSC receives a message via its customer-facing interface for creation/modification of an IETF network slice, it uses the provided NSEs to retrieve the corresponding border link or "Provider Node"

(e.g., PE). The NSC further maps them to the appropriate service/tunnel/path endpoints in the underlying network. It then uses services/tunnels/paths to realize the IETF network slice.

The 'ietf-network-slice' module uses two main data nodes: list 'ietf-network-slice' and container 'ns-templates' (see Figure 5).

The 'ietf-network-slice' list includes the set of IETF Network slices managed within a provider network. 'ietf-network-slice' is the data structure that abstracts an IETF Network Slice. Under the "ietf-network-slice", list "ns-endpoint" is used to abstract the NSEs, e.g. NSEs in the example above. And list "ns-connection" is used to abstract connections between NSEs.

The 'ns-templates' container is used by the NSC to maintain a set of common network slice templates that apply to one or several IETF Network Slices.

The figure below describes the overall structure of the YANG module:

```

module: ietf-network-slice
  +--rw network-slices
    +--rw ns-slo-sle-templates
      +--rw ns-slo-sle-template* [id]
        +--rw id string
        +--rw template-description? string
      +--rw network-slice* [ns-id]
        +--rw ns-id string
        +--rw ns-description? string
        +--rw customer-name* string
        +--rw ns-connectivity-type? identityref
        +--rw (ns-slo-sle-policy)?
          +--:(standard)
            +--rw slo-sle-template? leafref
          +--:(custom)
            +--rw slo-sle-policy
              +--rw policy-description? string
              +--rw ns-metric-bounds
                +--rw ns-metric-bound* [metric-type]
                  +--rw metric-type identityref
                  +--rw metric-unit string
                  +--rw value-description? string
                  +--rw bound? uint64
              +--rw security* identityref
              +--rw isolation? identityref
              +--rw max-occupancy-level? uint8
              +--rw mtu uint16
              +--rw steering-constraints

```

```

    +---rw path-constraints
    +---rw service-function
+---rw status
  +---rw admin-enabled?    boolean
  +---ro oper-status?      operational-type
+---rw ns-endpoints
  +---rw ns-endpoint* [ep-id]
    +---rw ep-id            string
    +---rw ep-description?  string
    +---rw ep-role?         identityref
    +---rw location
      +---rw altitude?      int64
      +---rw latitude?      decimal64
      +---rw longitude?     decimal64
    +---rw node-id?         string
    +---rw ep-ip?           inet:host
    +---rw ns-match-criteria
      +---rw ns-match-criterion* [match-type]
        +---rw match-type  identityref
        +---rw values* [index]
          +---rw index      uint8
          +---rw value?     string
    +---rw ep-peering
      +---rw protocol* [protocol-type]
        +---rw protocol-type identityref
        +---rw attribute* [index]
          +---rw index      uint8
          +---rw attribute-description? string
          +---rw value?     string
    +---rw ep-network-access-points
      +---rw ep-network-access-point* [network-access-id]
        +---rw network-access-id      string
        +---rw network-access-description? string
        +---rw network-access-node-id? string
        +---rw network-access-tp-id?  string
        +---rw network-access-tp-ip?  inet:host
        +---rw mtu                     uint16
      +---rw ep-rate-limit
        +---rw incoming-rate-limit?
          | te-types:te-bandwidth
        +---rw outgoing-rate-limit?
          | te-types:te-bandwidth
    +---rw ep-rate-limit
      +---rw incoming-rate-limit? te-types:te-bandwidth
      +---rw outgoing-rate-limit? te-types:te-bandwidth
+---rw status
  +---rw admin-enabled?    boolean
  +---ro oper-status?      operational-type

```

```

    +--ro ep-monitoring
      +--ro incoming-utilized-bandwidth?
        |   te-types:te-bandwidth
      +--ro incoming-bw-utilization          decimal64
      +--ro outgoing-utilized-bandwidth?
        |   te-types:te-bandwidth
      +--ro outgoing-bw-utilization          decimal64
+--rw ns-connections
  +--rw ns-connection* [ns-connection-id]
    +--rw ns-connection-id                  uint32
    +--rw ns-connection-description?        string
    +--rw src
      |   +--rw src-ep-id?    leafref
    +--rw dest
      |   +--rw dest-ep-id?   leafref
    +--rw (ns-slo-sle-policy)?
      +--:(standard)
        |   +--rw slo-sle-template?    leafref
      +--:(custom)
        +--rw slo-sle-policy
          +--rw policy-description?      string
          +--rw ns-metric-bounds
            +--rw ns-metric-bound* [metric-type]
              +--rw metric-type          identityref
              +--rw metric-unit          string
              +--rw value-description?    string
              +--rw bound?               uint64
          +--rw security*                identityref
          +--rw isolation?               identityref
          +--rw max-occupancy-level?     uint8
          +--rw mtu                     uint16
          +--rw steering-constraints
            +--rw path-constraints
            +--rw service-function
    +--rw monitoring-type?                 ns-monitoring-type
  +--ro ns-connection-monitoring
    +--ro latency?                        yang:gauge64
    +--ro jitter?                         yang:gauge32
    +--ro loss-ratio?                     decimal64

```

Figure 5

## 6. IETF Network Slice Templates

The 'ns-templates' container (Figure 5) is used by service provider of the NSC to define and maintain a set of common IETF Network Slice templates that apply to one or several IETF Network Slices. The exact definition of the templates is deployment specific to each network provider.

The model includes only the identifiers of SLO and SLE templates. When creation of IETF Network slice, the SLO and SLE policies can be easily identified.

The following shows an example where two network slice templates can be retrieved by the upper layer management system:

```
{
  "ietf-network-slices": {
    "ns-templates": {
      "slo-sle-template": [
        {
          "id": "GOLD-template",
          "template-description": "Two-way bandwidth: 1 Gbps,
            one-way latency 100ms "
          "sle-isolation": "ns-isolation-shared",
        },
        {
          "id": "PLATINUM-template",
          "template-description": "Two-way bandwidth: 1 Gbps,
            one-way latency 50ms "
          "sle-isolation": "ns-isolation-dedicated",
        },
      ],
    }
  }
}
```

## 7. IETF Network Slice Modeling Description

The 'ietf-network-slice' is the data structure that abstracts an IETF Network Slice of the IETF network. Each 'ietf-network-slice' is uniquely identified by an identifier: 'ns-id'.

An IETF Network Slice has the following main parameters:

- \* "ns-id": Is an identifier that is used to uniquely identify the IETF Network Slice within NSC.

- \* "ns-description": Gives some description of an IETF Network Slice service.
- \* "ns-connectivity-type": Indicates the network connectivity type for the IETF Network Slice: Hub-and-Spoke, any-to-any, or custom type.
- \* "status": Is used to show the operative and administrative status of the IETF Network Slice, and can be used as indicator to detect network slice anomalies.
- \* "customer-name": Is used to show the correlation between actual slice customers and IETF network slices. It can be used by the NSC for monitoring and assurance of the IETF network slices where NSC can notify the higher system by issuing the notifications. For example, multiple actual customers use a same network slice.
- \* "ns-slo-sle-policy": Defines SLO and SLE policies for the "ietf-network-slice". More description are provided in Section 7.2

The "ns-endpoint" is an abstrac entity that represents a set of matching rules applied to an IETF network edge device or a customer network edge device involved in the IETF Network Slice and each 'ns-endpoint' belongs to a single 'ietf-network-slice'. More description are provided in Section 7.3

#### 7.1. IETF Network Slice Connectivity Type

Based on the customer's traffic pattern requirements, an IETF Network Slice connection type could be point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), or multipoint-to-multipoint (MP2MP). The "ns-connectivity-type" under the node "ietf-network-slice" is used for this.

According to the network services defined in [I-D.ietf-opsawg-vpn-common], some well-known connectivity types are proposed for IETF network slices. The type could be any-to-any, Hub-and-Spoke (where Hubs can exchange traffic), and the custom. By default, the any-to-any is used. New connectivity type could be added via augmentation or by list of 'ns-connection' specified.

In addition, "ep-role" under the node "ns-endpoint" also needs to be defined, which specifies the role of the NSE in a particular Network Slice connectivity type. In the any-to-any, all NSEs MUST have the same role, which will be "any-to-any-role". In the Hub-and-Spoke, NSEs MUST have a Hub role or a Spoke role.

## 7.2. IETF Network Slice SLO and SLE Policy

As defined in [I-D.ietf-teas-ietf-network-slices], the SLO and SLE policy of an IETF Network Slice defines the minimum IETF Network Slice SLO attributes, and additional attributes can be added as needed.

"ns-slo-sle-policy" is used to represent specific SLO and SLE policies. During the creation of an IETF Network Slice, the policy can be specified either by a standard SLO and SLE template or a customized SLO and SLE policy.

The policy could both apply one per Network Slice or per connection 'ns-connection'.

The model allows multiple SLO and SLE attributes to be combined to meet different SLO and SLE requirements. For example, some NSs are used for video services and require high bandwidth, some NSs are used for key business services and request low latency and reliability, and some NSs need to provide connections for a large number of NSEs. That is, not all SLO or SLE attributes must be specified to meet the particular requirements of a slice.

"ns-metric-bounds" contains all these variations, which includes a list of "ns-metric-bound" and each "ns-metric-bound" could specify a particular "metric-type". "metric-type" is defined with YANG identity and the YANG module supports the following options:

"ns-slo-one-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two NSE. And the bandwidth is unidirectional.

"ns-slo-two-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two NSE. And the bandwidth is bidirectional.

"network-slice-slo-one-way-latency": Indicates the maximum one-way latency between two NSE.

"network-slice-slo-two-way-latency": Indicates the maximum round-trip latency between two NSE.

"ns-slo-one-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the one-way latency between sequential packets in a flow.

"ns-slo-two-way-delay-variation": Indicates the jitter constraint



of the slice maximum permissible delay variation, and is measured by the difference in the two-way latency between sequential packets in a flow.

"ns-slo-one-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

"ns-slo-two-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

"ns-slo-availability": Is defined as the ratio of up-time to total\_time(up-time+down-time), where up-time is the time the IETF Network Slice is available in accordance with the SLOs associated with it.

Some other Network Slice SLOs or SLEs could be extended when needed.

Note: The definition of "slo-sle-policy" and "steering-constraints" will be updated when WG converge on the terms.

Note: RFC7297 shaping/policing for out of profile traffic.

The following shows an example where a network slice policy can be configured:

```
{
  "ietf-network-slices": {
    "ietf-network-slice": {
      "slo-policy": {
        "policy-description": "video-service-policy",
        "ns-metric-bounds": {
          "ns-metric-bound": [
            {
              "metric-type": "ns-slo-one-way-bandwidth",
              "metric-unit": "mbps",
              "bound": "1000"
            },
            {
              "metric-type": "ns-slo-availability",
              "bound": "99.9%"
            }
          ],
        }
      }
    }
  }
}
```

### 7.3. IETF Network Slice Endpoint (NSE)

An IETF Network Slice Endpoint has several characteristics:

- \* "ep-id": Uniquely identifies the NSE within Network Slice Controller (NSC). The identifier is a string that allows any encoding for the local administration of the IETF Network Slice.
- \* "location": Indicates NSE location information that facilitates NSC easy identification of a NSE.
- \* "ep-role": Represents a connectivity type role of a NSE belonging to an IETF network slice, as described in Section 7.1. The "ep-role" leaf defines the role of the endpoint in a particular NS connectivity type. In the any-to-any, all NSEs MUST have the same role, which will be "any-to-any-role".
- \* "node-id": The NSE node information facilitates NSC with easy identification of a NSE.
- \* "ep-ip": The NSE IP information facilitates NSC with easy identification of a NSE.
- \* "ns-match-criteria": A matching policies to apply on a given NSE.

- \* "ep-network-access-points": The list of the interfaces attached to an edge device of the IETF Network Slice by which the customer traffic is received.
- \* "ep-rate-limit": Set the rate-limiting policies to apply on a given NSE, including ingress and egress traffic to ensure access security. When applied in the incoming direction, the rate-limit is applicable to the traffic from the NSE to the IETF scope Network that passes through the external interface. When Bandwidth is applied to the outgoing direction, it is applied to the traffic from the IETF Network to the NSE of that particular NS.
- \* "ep-protocol": Specify the protocol for a NSE for exchanging control-plane information, e.g. L1 signaling protocol or L3 routing protocols, etc.
- \* "status": Enable the control of the operative and administrative status of the NSE, can be used as indicator to detect NSE anomalies.

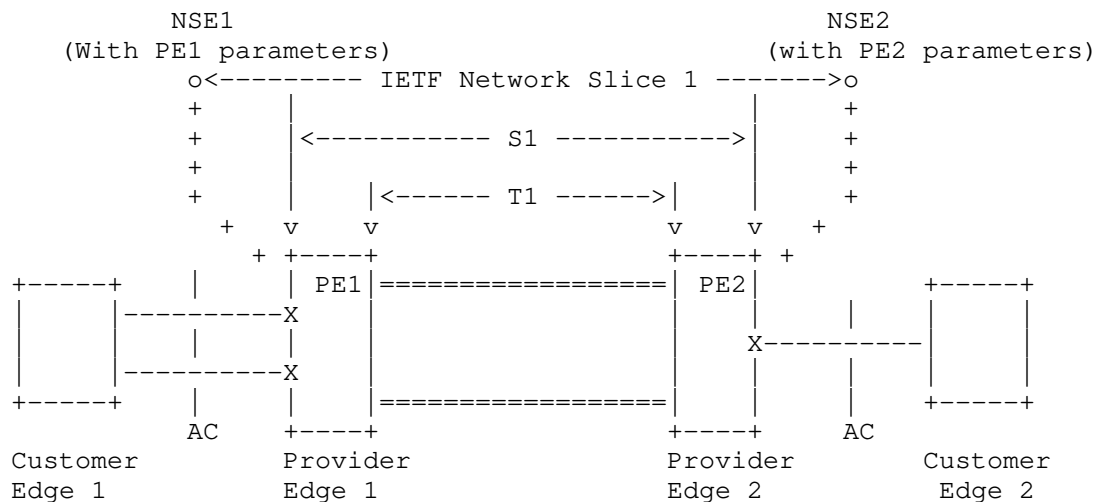
An NSE belong to a single IETF Network Slice. An IETF Network Slice involves two or more NSEs. An IETF Network Slice can be modified by adding new "ns-endpoint" or removing existing "ns-endpoint".

A NSE is used to define the matching rule on the customer traffic that can be injected to an IETF Network Slice. "network-slice-match-criteria" is defined to support different options. Classification can be based on many criteria, such as:

- \* Physical interface: Indicates all the traffic received from the interface belongs to the IETF Network Slice.
- \* Logical interface: For example, a given VLAN ID is used to identify an IETF Network Slice.
- \* Encapsulation in the traffic header: For example, a source IP address is used to identify an IETF Network Slice.

To illustrate the use of NSE parameters, the below are two examples. How the NSC realize the mapping is out of scope for this document.

- \* NSE with PE parameters example: As shown in Figure 6 , customer of the IETF network slice would like to connect two NSEs to satisfy specific service, e.g., Network wholesale services. In this case, the IETF network slice endpoints are mapped to physical interfaces of PE nodes. The IETF network slice controller (NSC) uses 'node-id' (PE device ID), 'ep-network-access-points' (Two PE interfaces ) to map the interfaces and corresponding services/tunnels/paths.

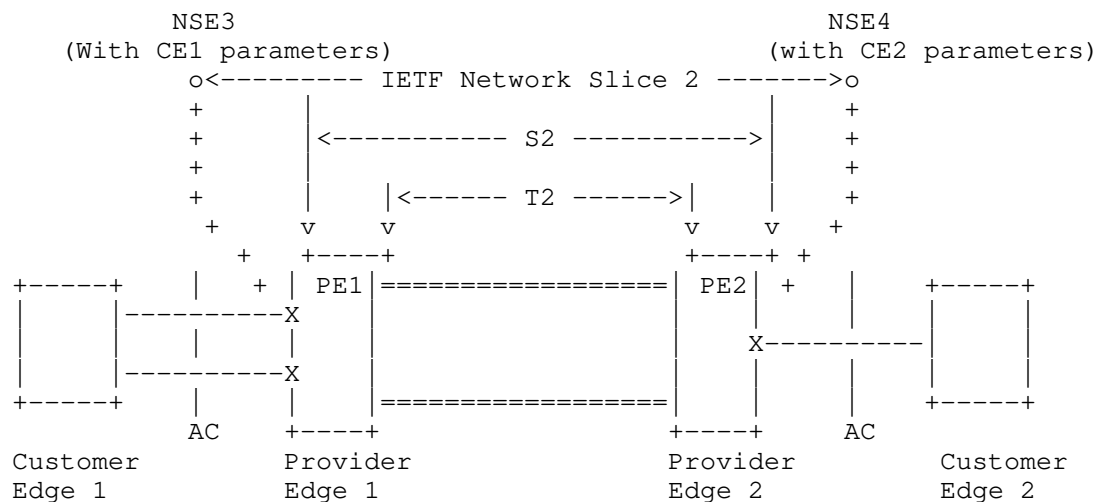


Legend:

- O: Representation of the IETF network slice endpoints (NSE)
- +: Mapping of NES to PE or CE nodes on IETF network
- X: Physical interfaces used for realization of IETF network slice
- S1: L0/L1/L2/L3 services used for realization of IETF network slice
- T1: Tunnels used for realization of IETF network slice

Figure 6

- \* NSE with CE parameters example: As shown in Figure 7 , customer of the IETF network slice would like to connect two NSEs to provide connectivity between transport portion of 5G RAN to 5G Core network functions. In this scenario, the IETF network slice controller (NSC) uses 'node-id' (CE device ID) , 'ep-ip' (CE tunnel endpoint IP), 'network-slice-match-criteria' (VLAN interface), 'ep-network-access-points' (Two nexthop interfaces ) to retrieve the corresponding border link or PE, and further map to services/tunnels/paths.



#### Legend:

O: Representation of the IETF network slice endpoints (NSE)  
 +: Mapping of NSE to PE or CE-PE interfaces on IETF network  
 X: Physical interfaces used for realization of IETF network slice  
 S2: L0/L1/L2/L3 services used for realization of IETF network slice  
 T2: Tunnels used for realization of IETF network slice

Figure 7

Note: The model needs to be optimized for better extension of other protocols or AC technologies.

## 8. IETF Network Slice Monitoring

An IETF Network Slice is a connectivity with specific SLO characteristics, including bandwidth, latency, etc. The connectivity is a combination of logical unidirectional connections, represented by 'ns-connection'.

This model also describes performance status of an IETF Network Slice. The statistics are described in the following granularity:

- \* Per NS connection: specified in 'ns-connection-monitoring' under the "ns-connection"
- \* Per NS Endpoint: specified in 'ep-monitoring' under the "ns-endpoint"

This model does not define monitoring enabling methods. The mechanism defined in [RFC8640] and [RFC8641] can be used for either periodic or on-demand subscription.

By specifying subtree filters or xpath filters to 'ns-connection' or 'ns-endpoint', so that only interested contents will be sent. These mechanisms can be used for monitoring the IETF Network Slice performance status so that the customer management system could initiate modification based on the IETF Network Slice running status.

Note: More critical events affecting service delivery need to be added.

## 9. IETF Network Slice Service Module

The "ietf-network-slice" module uses types defined in [RFC6991], [RFC8776].

```
<CODE BEGINS> file "ietf-network-slice@2021-07-20.yang"
module ietf-network-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-slice";
  prefix ins;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Types.";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Types.";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering.";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group";
  contact
    "WG Web: <https://tools.ietf.org/wg/teas/>
     WG List: <mailto:teas@ietf.org>
     Editor: Bo Wu <lana.wubo@huawei.com>
           : Dhruv Dhody <dhruv.ietf@gmail.com>
```

```
        : Reza Rokui <reza.rokui@nokia.com>
        : Tarek Saad <tsaad@juniper.net>;
description
  "This module contains a YANG module for the IETF Network Slice.

  Copyright (c) 2021 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject to
  the license terms contained in, the Simplified BSD License set
  forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see the
  RFC itself for full legal notices."

revision 2021-07-20 {
  description
    "initial version.";
  reference
    "RFC XXXX: A Yang Data Model for IETF Network Slice Operation";
}

/* Features */
/* Identities */

identity ns-isolation-type {
  description
    "Base identity for IETF Network slice isolation level.";
}

identity ns-isolation-shared {
  base ns-isolation-type;
  description
    "Shared resources (e.g. queues) are associated with the Network
    Slice traffic. Hence, the IETF network slice traffic can be
    impacted by effects of other services traffic sharing
    the same resources.";
}

identity ns-isolation-dedicated {
  base ns-isolation-type;
  description
    "Dedicated resources (e.g. queues) are associated with the
    Network Slice traffic. Hence, the IETF network slice traffic
    is isolated from other servceis traffic sharing the same
```

```
        resources.";
    }

    identity ns-security-type {
        description
            "Base identity for for IETF Network security level.";
    }

    identity ns-security-authenticate {
        base ns-security-type;
        description
            "IETF Network Slice requires authentication.";
    }

    identity ns-security-integrity {
        base ns-security-type;
        description
            "IETF Network Slice requires data integrity.";
    }

    identity ns-security-encryption {
        base ns-security-type;
        description
            "IETF Network Slice requires data encryption.";
    }

    identity ns-connectivity-type {
        description
            "Base identity for IETF Network Slice topology.";
    }

    identity any-to-any {
        base ns-connectivity-type;
        description
            "Identity for any-to-any IETF Network Slice topology.";
    }

    identity hub-spoke {
        base ns-connectivity-type;
        description
            "Identity for Hub-and-Spoke IETF Network Slice topology.";
    }

    identity custom {
        base ns-connectivity-type;
        description
            "Identity of a custom NS topology where Hubs can act as
            Spoke for certain parts of the network or Spokes as Hubs.";
```



```
}

identity endpoint-role {
  description
    "Base identity of a NSE role in an IETF Network Slice topology.";
}

identity any-to-any-role {
  base endpoint-role;
  description
    "Identity of any-to-any NS.";
}

identity spoke-role {
  base endpoint-role;
  description
    "A NSE is acting as a Spoke.";
}

identity hub-role {
  base endpoint-role;
  description
    "A NSE is acting as a Hub.";
}

identity ns-slo-metric-type {
  description
    "Base identity for IETF Network Slice SLO metric type.";
}

identity ns-slo-one-way-bandwidth {
  base ns-slo-metric-type;
  description
    "SLO bandwidth metric. Minimum guaranteed bandwidth between
    two endpoints at any time and is measured unidirectionally";
}

identity ns-slo-two-way-bandwidth {
  base ns-slo-metric-type;
  description
    "SLO bandwidth metric. Minimum guaranteed bandwidth between
    two endpoints at any time";
}

identity ns-slo-one-way-latency {
  base ns-slo-metric-type;
  description
    "SLO one-way latency is upper bound of network latency when
```

```
        transmitting between two endpoints. The metric is defined in
        RFC7679";
    }

    identity ns-slo-two-way-latency {
        base ns-slo-metric-type;
        description
            "SLO two-way latency is upper bound of network latency when
            transmitting between two endpoints. The metric is defined in
            RFC2681";
    }

    identity ns-slo-one-way-delay-variation {
        base ns-slo-metric-type;
        description
            "SLO one-way delay variation is defined by RFC3393, is the
            difference in the one-way delay between sequential packets
            between two endpoints.";
    }

    identity ns-slo-two-way-delay-variation {
        base ns-slo-metric-type;
        description
            "SLO two-way delay variation is defined by RFC5481, is the
            difference in the round-trip delay between sequential packets
            between two endpoints.";
    }

    identity ns-slo-one-way-packet-loss {
        base ns-slo-metric-type;
        description
            "SLO loss metric. The ratio of packets dropped to packets
            transmitted between two endpoints in one-way
            over a period of time as specified in RFC7680";
    }

    identity ns-slo-two-way-packet-loss {
        base ns-slo-metric-type;
        description
            "SLO loss metric. The ratio of packets dropped to packets
            transmitted between two endpoints in two-way
            over a period of time as specified in RFC7680";
    }

    identity ns-slo-availability {
        base ns-slo-metric-type;
        description
            "SLO availability level.";
```

```
    }

    identity ns-match-type {
      description
        "Base identity for IETF Network Slice traffic match type.";
    }

    identity ns-phy-interface-match {
      base ns-match-type;
      description
        "Use the physical interface as match criteria for the IETF
        Network Slice traffic.";
    }

    identity ns-vlan-match {
      base ns-match-type;
      description
        "Use the VLAN ID as match criteria for the IETF Network Slice
        traffic.";
    }

    identity ns-label-match {
      base ns-match-type;
      description
        "Use the MPLS label as match criteria for the IETF Network
        Slice traffic.";
    }

    identity peering-protocol-type {
      description
        "Base identity for NSE peering protocol type.";
    }

    identity peering-protocol-bgp {
      base peering-protocol-type;
      description
        "Use BGP as protocol for NSE peering with customer device.";
    }

    identity peering-static-routing {
      base peering-protocol-type;
      description
        "Use static routing for NSE peering with customer device.";
    }

    /*
     * Identity for availability-type
     */
```

```
identity availability-type {
  description
    "Base identity from which specific availability types are
    derived.";
}

identity level-1 {
  base availability-type;
  description
    "level 1: 99.9999%";
}

identity level-2 {
  base availability-type;
  description
    "level 2: 99.999%";
}

identity level-3 {
  base availability-type;
  description
    "level 3: 99.99%";
}

identity level-4 {
  base availability-type;
  description
    "level 4: 99.9%";
}

identity level-5 {
  base availability-type;
  description
    "level 5: 99%";
}

/* typedef */

typedef operational-type {
  type enumeration {
    enum up {
      value 0;
      description
        "Operational status UP.";
    }
    enum down {
      value 1;
      description
```

```
        "Operational status DOWN.";
    }
    enum unknown {
        value 2;
        description
            "Operational status UNKNOWN.";
    }
}
description
    "This is a read-only attribute used to determine the
    status of a particular element.";
}

typedef ns-monitoring-type {
    type enumeration {
        enum one-way {
            description
                "Represents one-way measurments monitoring type.";
        }
        enum two-way {
            description
                "represents two-way measurements monitoring type.";
        }
    }
}
description
    "An enumerated type for monitoring on a IETF Network Slice
    connection.";
}

/* Groupings */

grouping status-params {
    description
        "A grouping used to join operational and administrative status.";
    container status {
        description
            "A container for the administrative and operational state.";
        leaf admin-enabled {
            type boolean;
            description
                "The administrative status.";
        }
        leaf oper-status {
            type operational-type;
            config false;
            description
                "The operational status.";
        }
    }
}
```

```
    }
  }

  grouping ns-match-criteria {
    description
      "A grouping for the IETF Network Slice match definition.";
    container ns-match-criteria {
      description
        "Describes the IETF Network Slice match criteria.";
      list ns-match-criterion {
        key "match-type";
        description
          "List of the IETF Network Slice traffic match criteria.";
        leaf match-type {
          type identityref {
            base ns-match-type;
          }
          description
            "Identifies an entry in the list of the IETF Network Slice
            match criteria.";
        }
        list values {
          key "index";
          description
            "List of match criteria values.";
          leaf index {
            type uint8;
            description
              "Index of an entry in the list.";
          }
          leaf value {
            type string;
            description
              "Describes the IETF Network Slice match criteria, e.g.
              IP address, VLAN, etc.";
          }
        }
      }
    }
  }

  grouping ns-connection-group-metric-bounds {
    description
      "Grouping of Network Slice metric bounds that
      are shared amongst multiple connections of a Network
      Slice.";
    leaf ns-slo-shared-bandwidth {
      type te-types:te-bandwidth;
    }
  }
}
```

```
        description
            "A limit on the bandwidth that is shared amongst
            multiple connections of an IETF Network Slice.";
    }
}

grouping ns-sles {
    description
        "Indirectly Measurable Objectives of a IETF Network
        Slice.";
    leaf-list security {
        type identityref {
            base ns-security-type;
        }
        description
            "The IETF Network Slice security SLE(s)";
    }
    leaf isolation {
        type identityref {
            base ns-isolation-type;
        }
        default "ns-isolation-shared";
        description
            "The IETF Network Slice isolation SLE requirement.";
    }
    leaf max-occupancy-level {
        type uint8 {
            range "1..100";
        }
        description
            "The maximal occupancy level specifies the number of flows to
            be admitted.";
    }
    leaf mtu {
        type uint16;
        units "bytes";
        mandatory true;
        description
            "The MTU specifies the maximum length in octets of data
            packets that can be transmitted by the NS. The value needs
            to be less than or equal to the minimum MTU value of
            all 'ep-network-access-points' in the NSEs of the NS. ";
    }
    container steering-constraints {
        description
            "Container for the policy of steering constraints
            applicable to IETF Network Slice.";
        container path-constraints {
```

```
        description
            "Container for the policy of path constraints
             applicable to IETF Network Slice.";
    }
    container service-function {
        description
            "Container for the policy of service function
             applicable to IETF Network Slice.";
    }
}

grouping ns-metric-bounds {
    description
        "IETF Network Slice metric bounds grouping.";
    container ns-metric-bounds {
        description
            "IETF Network Slice metric bounds container.";
        list ns-metric-bound {
            key "metric-type";
            description
                "List of IETF Network Slice metric bounds.";
            leaf metric-type {
                type identityref {
                    base ns-slo-metric-type;
                }
                description
                    "Identifies an entry in the list of metric type
                     bounds for the IETF Network Slice.";
            }
            leaf metric-unit {
                type string;
                mandatory true;
                description
                    "The metric unit of the parameter. For example,
                     s, ms, ns, and so on.";
            }
            leaf value-description {
                type string;
                description
                    "The description of previous value. ";
            }
            leaf bound {
                type uint64;
                default "0";
                description
                    "The Bound on the Network Slice connection metric. A
                     zero indicate an unbounded upper limit for the
```



```
        specific metric-type.";
    }
}
}

grouping ep-peering {
  description
    "A grouping for the IETF Network Slice Endpoint peering.";
  container ep-peering {
    description
      "Describes NSE peering attributes.";
    list protocol {
      key "protocol-type";
      description
        "List of the NSE peering protocol.";
      leaf protocol-type {
        type identityref {
          base peering-protocol-type;
        }
        description
          "Identifies an entry in the list of NSE peering
            protocol type.";
      }
      list attribute {
        key "index";
        description
          "List of protocol attribute.";
        leaf index {
          type uint8;
          description
            "Index of an entry in the list.";
        }
        leaf attribute-description {
          type string;
          description
            "The description of the attribute. ";
        }
        leaf value {
          type string;
          description
            "Describes the value of protocol attribute, e.g.
              nexthop address, peer address, etc.";
        }
      }
    }
  }
}
```

```
grouping ep-network-access-points {
  description
    "Grouping for the endpoint network access definition.";
  container ep-network-access-points {
    description
      "List of network access points.";
    list ep-network-access-point {
      key "network-access-id";
      description
        "The IETF Network Slice network access points
        related parameters.";
      leaf network-access-id {
        type string;
        description
          "Uniquely identifier a network access point.";
      }
      leaf network-access-description {
        type string;
        description
          "The network access point description.";
      }
      leaf network-access-node-id {
        type string;
        description
          "The network access point node ID in the case of
          multi-homing.";
      }
      leaf network-access-tp-id {
        type string;
        description
          "The termination port ID of the EP network access
          point.";
      }
      leaf network-access-tp-ip {
        type inet:host;
        description
          "The IP address of the EP network access point.";
      }
      leaf mtu {
        type uint16;
        units "bytes";
        mandatory true;
        description
          "Maximum size in octets of a data packet that
          can traverse a NSE network access point. ";
      }
    }
    /* Per ep-network-access-point rate limits */
    uses ns-rate-limit;
  }
}
```

```
    }  
  }  
}  
  
grouping endpoint-monitoring-parameters {  
  description  
    "Grouping for the endpoint monitoring parameters.";  
  container ep-monitoring {  
    config false;  
    description  
      "Container for endpoint monitoring parameters.";  
    leaf incoming-utilized-bandwidth {  
      type te-types:te-bandwidth;  
      description  
        "Incoming bandwidth utilization at an endpoint.";  
    }  
    leaf incoming-bw-utilization {  
      type decimal64 {  
        fraction-digits 5;  
        range "0..100";  
      }  
      units "percent";  
      mandatory true;  
      description  
        "To be used to define the bandwidth utilization  
        as a percentage of the available bandwidth.";  
    }  
    leaf outgoing-utilized-bandwidth {  
      type te-types:te-bandwidth;  
      description  
        "Outgoing bandwidth utilization at an endpoint.";  
    }  
    leaf outgoing-bw-utilization {  
      type decimal64 {  
        fraction-digits 5;  
        range "0..100";  
      }  
      units "percent";  
      mandatory true;  
      description  
        "To be used to define the bandwidth utilization  
        as a percentage of the available bandwidth.";  
    }  
  }  
}  
  
grouping common-monitoring-parameters {  
  description
```

```
    "Grouping for link-monitoring-parameters.";
  leaf latency {
    type yang:gauge64;
    units "usec";
    description
      "The latency statistics per Network Slice connection.
      RFC2681 and RFC7679 discuss round trip times and one-way
      metrics, respectively";
  }
  leaf jitter {
    type yang:gauge32;
    description
      "The jitter statistics per Network Slice member
      as defined by RFC3393.";
  }
  leaf loss-ratio {
    type decimal64 {
      fraction-digits 6;
      range "0 .. 50.331642";
    }
    description
      "Packet loss as a percentage of the total traffic
      sent over a configurable interval. The finest precision is
      0.000003%. where the maximum 50.331642%.";
    reference
      "RFC 7810, section-4.4";
  }
}

grouping geolocation-container {
  description
    "A grouping containing a GPS location.";
  container location {
    description
      "A container containing a GPS location.";
    leaf altitude {
      type int64;
      units "millimeter";
      description
        "Distance above the sea level.";
    }
    leaf latitude {
      type decimal64 {
        fraction-digits 8;
        range "-90..90";
      }
      description
        "Relative position north or south on the Earth's surface.";
    }
  }
}
```

```
    }
    leaf longitude {
      type decimal64 {
        fraction-digits 8;
        range "-180..180";
      }
      description
        "Angular distance east or west on the Earth's surface.";
    }
  }
  // gps-location
}

// geolocation-container

grouping ns-rate-limit {
  description
    "The Network Slice rate limit grouping.";
  container ep-rate-limit {
    description
      "Container for the asymmetric traffic control";
    leaf incoming-rate-limit {
      type te-types:te-bandwidth;
      description
        "The rate-limit imposed on incoming traffic.";
    }
    leaf outgoing-rate-limit {
      type te-types:te-bandwidth;
      description
        "The rate-limit imposed on outgoing traffic.";
    }
  }
}

grouping endpoint {
  description
    "IETF Network Slice endpoint related information";
  leaf ep-id {
    type string;
    description
      "unique identifier for the referred IETF Network
        Slice endpoint";
  }
  leaf ep-description {
    type string;
    description
      "endpoint name";
  }
}
```

```
leaf ep-role {
  type identityref {
    base endpoint-role;
  }
  default "any-to-any-role";
  description
    "Role of the endpoint in the IETF Network Slice.";
}
uses geolocation-container;
leaf node-id {
  type string;
  description
    "Uniquely identifies an edge node within the IETF slice
    network.";
}
leaf ep-ip {
  type inet:host;
  description
    "The address of the endpoint IP address.";
}
uses ns-match-criteria;
uses ep-peering;
uses ep-network-access-points;
uses ns-rate-limit;
/* Per NSE rate limits */
uses status-params;
uses endpoint-monitoring-parameters;
}

//ns-endpoint

grouping ns-connection {
  description
    "The Network Slice connection is described in this container.";
  leaf ns-connection-id {
    type uint32;
    description
      "The Network Slice connection identifier";
  }
  leaf ns-connection-description {
    type string;
    description
      "The Network Slice connection description";
  }
}
container src {
  description
    "the source of Network Slice link";
  leaf src-ep-id {
```

```
        type leafref {
            path "/network-slices/network-slice"
              + "/ns-endpoints/ns-endpoint/ep-id";
        }
        description
            "reference to source Network Slice endpoint";
    }
}
container dest {
    description
        "the destination of Network Slice link ";
    leaf dest-ep-id {
        type leafref {
            path "/network-slices/network-slice"
              + "/ns-endpoints/ns-endpoint/ep-id";
        }
        description
            "reference to dest Network Slice endpoint";
    }
}
uses ns-slo-sle-policy;
/* Per connection ns-slo-sle-policy overrides
 * the per network slice ns-slo-sle-policy.
 */
leaf monitoring-type {
    type ns-monitoring-type;
    description
        "One way or two way monitoring type.";
}
container ns-connection-monitoring {
    config false;
    description
        "SLO status Per network-slice endpoint to endpoint ";
    uses common-monitoring-parameters;
}
}

//ns-connection

grouping slice-template {
    description
        "Grouping for slice-templates.";
    container ns-slo-sle-templates {
        description
            "Contains a set of network slice templates to
             reference in the IETF network slice.";
        list ns-slo-sle-template {
            key "id";
        }
    }
}
```

```
    leaf id {
      type string;
      description
        "Identification of the Service Level Objective (SLO)
        and Service Level Expectation (SLE) template to be used.
        Local administration meaning.";
    }
    leaf template-description {
      type string;
      description
        "Description of the SLO & SLE policy template.";
    }
    description
      "List for SLO and SLE template identifiers.";
  }
}

/* Configuration data nodes */

grouping ns-slo-sle-policy {
  description
    "Network Slice policy grouping.";
  choice ns-slo-sle-policy {
    description
      "Choice for SLO and SLE policy template.
      Can be standard template or customized template.";
    case standard {
      description
        "Standard SLO template.";
      leaf slo-sle-template {
        type leafref {
          path "/network-slices"
            + "/ns-slo-sle-templates/ns-slo-sle-template/id";
        }
        description
          "Standard SLO and SLE template to be used.";
      }
    }
    case custom {
      description
        "Customized SLO template.";
      container slo-sle-policy {
        description
          "Contains the SLO policy.";
        leaf policy-description {
          type string;
          description
```



```
        "Description of the SLO policy.";
    }
    uses ns-metric-bounds;
    uses ns-sles;
}
}
}

container network-slices {
  description
    "IETF network-slice configurations";
  uses slice-template;
  list network-slice {
    key "ns-id";
    description
      "a network-slice is identified by a ns-id";
    leaf ns-id {
      type string;
      description
        "A unique network-slice identifier across an IETF NSC ";
    }
    leaf ns-description {
      type string;
      description
        "Give more description of the network slice";
    }
    leaf-list customer-name {
      type string;
      description
        "List of the customer that actually uses the slice.
        In the case that multiple customers sharing
        same slice service, e.g., 5G, customer name may
        help with operational management";
    }
    leaf ns-connectivity-type {
      type identityref {
        base ns-connectivity-type;
      }
      default "any-to-any";
      description
        "Network Slice topology.";
    }
    uses ns-slo-sle-policy;
    uses status-params;
    container ns-endpoints {
      description
        "Endpoints";
    }
  }
}
```

```
    list ns-endpoint {
      key "ep-id";
      uses endpoint;
      description
        "List of endpoints in this slice";
    }
  }
  container ns-connections {
    description
      "Connections container";
    list ns-connection {
      key "ns-connection-id";
      description
        "List of Network Slice connections.";
      uses ns-connection;
    }
  }
}
//ietf-network-slice list
}
}
<CODE ENDS>
```

## 10. Security Considerations

The YANG module defined in this document is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

o /ietf-network-slice/network-slices/network-slice

The entries in the list above include the whole network configurations corresponding with the slice which the higher management system requests, and indirectly create or modify the PE or P device configurations. Unexpected changes to these entries could lead to service disruption and/or network misbehavior.

## 11. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-network-slice  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

This document requests to register a YANG module in the YANG Module Names registry [RFC7950].

Name: ietf-network-slice  
Namespace: urn:ietf:params:xml:ns:yang:ietf-network-slice  
Prefix: ins  
Reference: RFC XXXX

## 12. Acknowledgments

The authors wish to thank Mohamed Boucadair, Kenichi Ogaki, Sergio Belotti, Qin Wu, Susan Hares, Eric Grey, and many others for their helpful comments and suggestions.

## 13. References

### 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", RFC 8640, DOI 10.17487/RFC8640, September 2019, <<https://www.rfc-editor.org/info/rfc8640>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.

- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.

### 13.2. Informative References

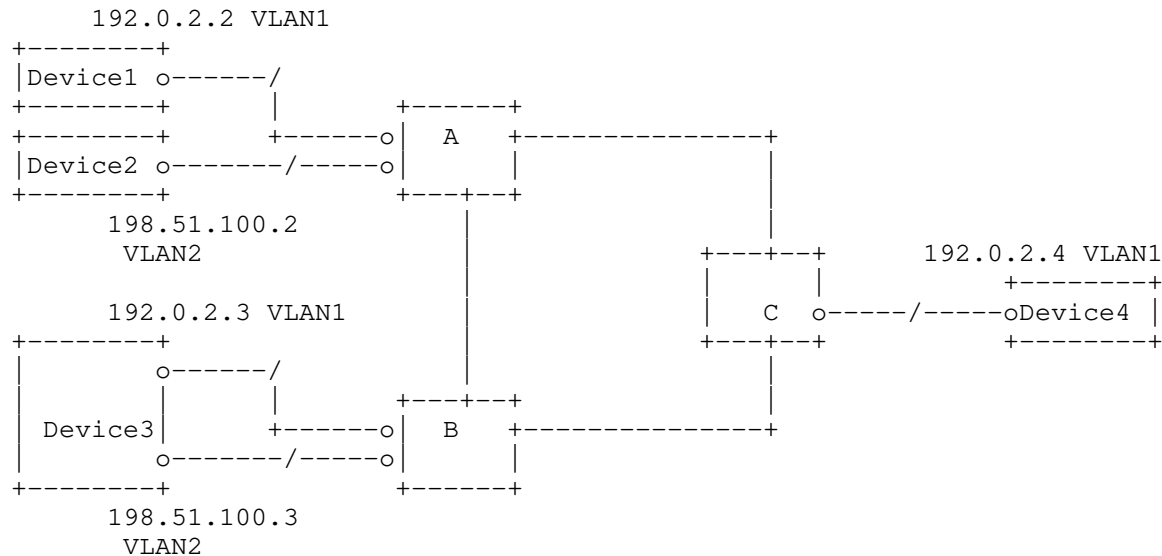
- [I-D.geng-teas-network-slice-mapping]  
Geng, X., Dong, J., Pang, R., Han, L., Niwa, T., Jin, J., Liu, C., and N. Nageshar, "5G End-to-end Network Slice Mapping from the view of Transport Network", Work in Progress, Internet-Draft, draft-geng-teas-network-slice-mapping-03, 22 February 2021, <<https://www.ietf.org/archive/id/draft-geng-teas-network-slice-mapping-03.txt>>.
- [I-D.ietf-opsawg-vpn-common]  
Barguil, S., Dios, O. G. D., Boucadair, M., and Q. Wu, "A Layer 2/3 VPN Common YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-vpn-common-11, 23 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-opsawg-vpn-common-11.txt>>.
- [I-D.ietf-teas-actn-vn-yang]  
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Y. Yoon, "A YANG Data Model for VN Operation", Work in Progress, Internet-Draft, draft-ietf-teas-actn-vn-yang-12, 25 August 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-actn-vn-yang-12.txt>>.
- [I-D.ietf-teas-ietf-network-slices]  
Farrel, A., Gray, E., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-04, 23 August 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-04.txt>>.
- [RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.

[RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.

#### Appendix A. IETF Network Slice NBI Model Usage Example

The following example describes a simplified service configuration of two IETF Network slice instances:

- \* IETF Network Slice 1 on Device1, Device3, and Device4, with any-to-any connectivity type
- \* IETF Network Slice 2 on Device2, Device3, with any-to-any connectivity type



POST: /restconf/data/ietf-network-slice:ietf-network-slices

Host: example.com

Content-Type: application/yang-data+json

```
{
  "network-slices":{
    "network-slice":[
      {
        "ns-id":"1",
        "ns-description":"slice1",
        "ns-connectivity-type":"any-to-any",
```

```
"ns-endpoints":{
  "ns-endpoint":[
    {
      "ep-id":"11",
      "ep-description":"slice1 ep1 connected to device 1",
      "ep-role":"any-to-any-role",
      "ns-match-criteria":[
        {
          "match-type":"ns-vlan-match",
          "value":[
            {
              "index":"1",
              "value":"1"
            }
          ]
        }
      ]
    },
    {
      "ep-id":"12",
      "ep-description":"slice1 ep2 connected to device 3",
      "ep-role":"any-to-any-role",
      "ns-match-criteria":[
        {
          "match-type":"ns-vlan-match",
          "value":[
            {
              "index":"1",
              "value":"20"
            }
          ]
        }
      ]
    },
    {
      "ep-id":"13",
      "ep-description":"slice1 ep3 connected to device 4",
      "ep-role":"any-to-any-role",
      "ns-match-criteria":[
        {
          "match-type":"ns-vlan-match",
          "value":[
            {
              "index":"1",
              "value":"1"
            }
          ]
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
},
{
  "ns-id":"ns2",
  "ns-description":"slice2",
  "ns-connectivity-type":"any-to-any",
  "ns-endpoints":{
    "ns-endpoint":[
      {
        "ep-id":"21",
        "ep-description":"slice2 ep1 connected to device 2",
        "ep-role":"any-to-any-role",
        "ns-match-criteria":[
          {
            "match-type":"ns-vlan-match",
            "value":[
              {
                "index":"1",
                "value":"2"
              }
            ]
          }
        ]
      },
      {
        "ep-id":"22",
        "ep-description":"slice2 ep2 connected to device 3",
        "ep-role":"any-to-any-role",
        "ns-match-criteria":[
          {
            "match-type":"ns-vlan-match",
            "value":[
              {
                "index":"1",
                "value":"2"
              }
            ]
          }
        ]
      }
    ]
  }
}
]
}

```

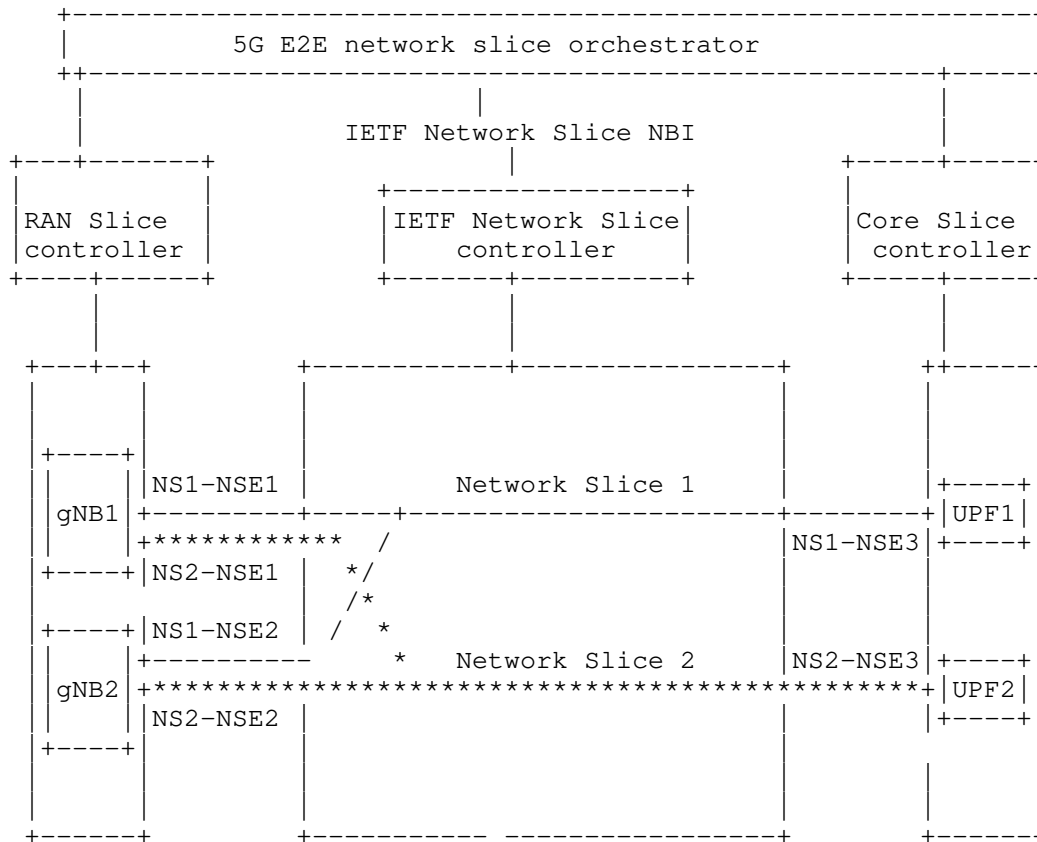


}

## Appendix B. Appendix B IETF Network Slice Match Criteria

5G is a use case of the IETF Network Slice and 5G End-to-end Network Slice Mapping from the view of IETF Network[I-D.geng-teas-network-slice-mapping]

defines two types of Network Slice interconnection and differentiation methods: by physical interface or by TNSII (Transport Network Slice Interworking Identifier). TNSII is a field in the packet header when different 5G wireless network slices are transported through a single physical interfaces of the IETF scoped Network. In the 5G scenario, "network-slice-match-criteria" refers to TNSII.



As shown in the figure, gNodeB 1 and gNodeB 2 use IP gNB1 and IP gNB2 to communicate with the IETF network, respectively. In addition, the traffic of NS1 and NS2 on gNodeB 1 and gNodeB 2 is transmitted through the same access links to the IETF slice network. The IETF slice network need to to distinguish different IETF Network Slice traffic of same gNB. Therefore, in addition to using "node-id" and "ep-ip" to identify a Network Slice Endpoint, other information is needed along with these parameters to uniquely distinguish a NSE. For example, VLAN IDs in the user traffic can be used to distinguish the NSEs of gNBs and UPFs.

#### Authors' Addresses

Bo Wu  
Huawei Technologies  
101 Software Avenue, Yuhua District  
Nanjing  
Jiangsu, 210012  
China

Email: lana.wubo@huawei.com

Dhruv Dhody  
Huawei Technologies  
Divyashree Techno Park  
Bangalore 560066  
Karnataka  
India

Email: dhruv.ietf@gmail.com

Reza Rokui  
Nokia

Email: reza.rokui@nokia.com

Tarek Saad  
Juniper Networks

Email: tsaad@juniper.net

Liuyan Han  
China Mobile

Email: hanliuyan@chinamobile.com

Luis Miguel Contreras  
Telefonica  
Distrito T  
28050 Madrid  
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: December 8, 2021

B. Wu  
D. Dhody  
Huawei Technologies  
June 6, 2021

A VTN Network YANG Module  
draft-wd-teas-vtn-network-yang-00

## Abstract

This document defines a virtual transport network (VTN) network YANG module for retrieving and manipulating VTN topology and resource allocation. The model can be used to implement the provisioning of IETF network slice services.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 8, 2021.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                               | 2  |
| 2. Conventions used in this document . . . . .          | 3  |
| 2.1. Tree Diagrams . . . . .                            | 3  |
| 3. VTN Network Yang Module Consideration . . . . .      | 3  |
| 3.1. VTN Operation . . . . .                            | 6  |
| 3.2. VTN Network Modeling Design . . . . .              | 7  |
| 4. Description of the VTN Network YANG Module . . . . . | 7  |
| 5. VTN Yang Module Tree . . . . .                       | 8  |
| 6. VTN Yang Module . . . . .                            | 10 |
| 7. Security Considerations . . . . .                    | 18 |
| 8. IANA Considerations . . . . .                        | 18 |
| 9. Contributor . . . . .                                | 19 |
| 10. References . . . . .                                | 19 |
| 10.1. Normative References . . . . .                    | 19 |
| 10.2. Informative References . . . . .                  | 21 |
| Appendix A. Example VTN Network Model . . . . .         | 22 |
| Authors' Addresses . . . . .                            | 22 |

## 1. Introduction

[I-D.ietf-teas-ietf-network-slices] defines IETF network slice services that provide connectivity coupled with network resources commitment between a number of endpoints over a shared network infrastructure, and also defines the IETF Network Slice controller (NSC) to realize the network slice services by mapping it to a suitable underlying technology.

[I-D.ietf-teas-enhanced-vpn] describes that enhanced VPN (VPN+) services can be used to realize IETF network slice services. To improve service scalability, The virtual transport network (VTN), which has a customized network topology and a group of dedicated or shared nodes and links of the physical network, is introduced for multiple VPN+ services with similar connection and SLA requirements. For the control and management of these VTN resources, [I-D.dong-teas-enhanced-vpn-vtn-scalability] gives a detailed analysis and description.

This document defines VTN network model that the NSC can use to create and manage VTN instances to realize the network slicing services. According to the YANG model classification of [RFC8309], VTN network model is a network configuration model.

## 2. Conventions used in this document

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC6241] and are used in this specification:

- o configuration data
- o state data

The following terms are defined in [RFC7950] and are used in this specification:

- o augment
- o data model
- o data node

The terminology for describing YANG data models is found in [RFC7950].

### 2.1. Tree Diagrams

The tree diagram used in this document follows the notation defined in [RFC8340].

## 3. VTN Network Yang Module Consideration

To realize the IETF Network Slice based on the reference framework defined in [I-D.ietf-teas-ietf-network-slices] , the Figure 1 shows an approach with VPN network model and VTN network YANG module.

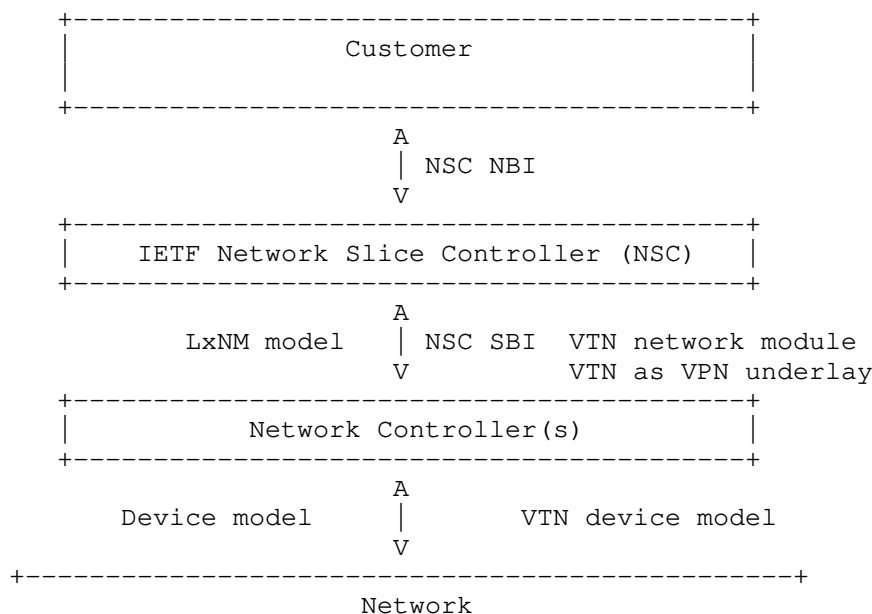


Figure 1: Reference Module Use Case

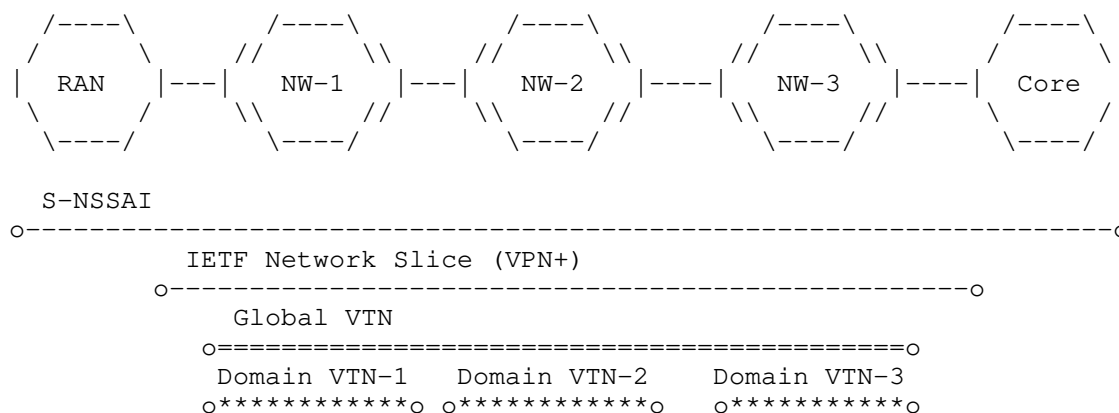
The VTN network model can be used in the following ways:

- o Static VTN configuration: A VTN instance can be created before processing IETF Network Slice service request by a network controller.
- o Dynamic VTN configuration: A VTN instance can be initiated along with configuring IETF Network Slice service request by a network controller.

In the process of realizing an IETF network slice service, when creating a Layer 3 VPN or Layer 2 VPN instance, The NSC can use a static VTN instance or dynamically create one as the VPN underlay transport. Compared with existing VPN underlying full mesh tunneling mechanisms, the VTN could provide resource isolation, topology constraints, and simplified configuration. Additionally, specific service flows of a VPN can be further optimized using SR policies defined in [I-D.dong-idr-sr-policy-vtn].

And also in multi-domain network slicing cases, instead of mapping the overlay VPN to the intra-domain VTNs at the edge of each domain, an inter-domain VTN could be used directly for inter-domain interconnection, which is described in

[I-D.li-teas-e2e-ietf-network-slicing] . The network controller serving the transit domain can only manage the VTNs. A 5G end-to-end network slicing scenario is shown in the following figure.



#### 5G end-to-end network slicing scenario

In addition to providing VTN network configuration, VTN network model also provides monitoring details of the underlying resource created to meet the requirements of IETF network slice service.

An example of VTN instances and a physical network is illustrated in Figure 2.



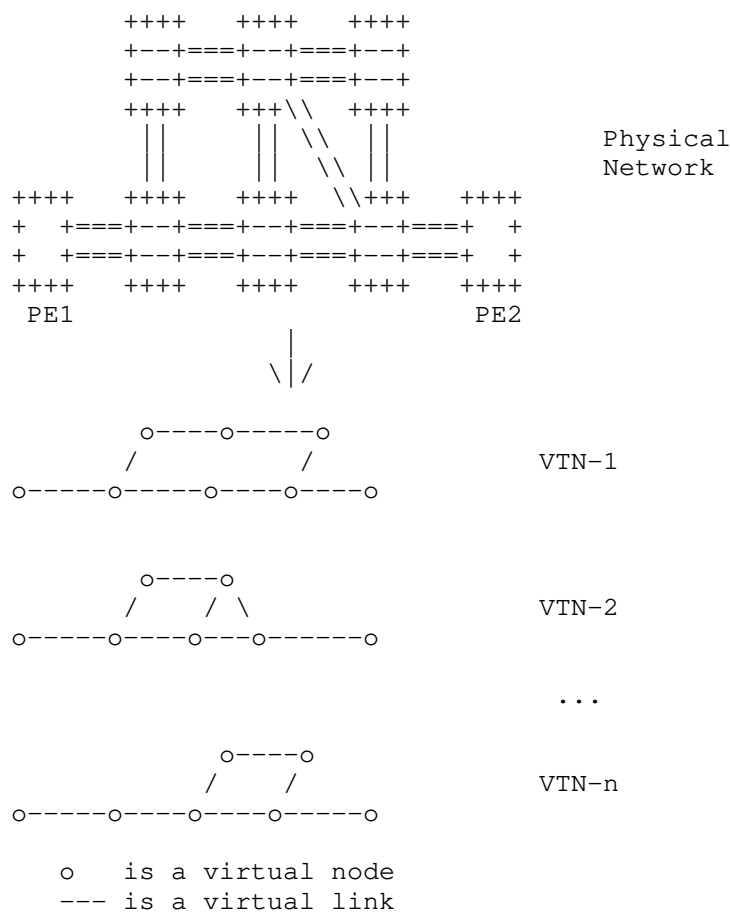


Figure 2: A VTN example

In the example, each VTN instance has a customized network topology comprised of a set of links and nodes in the physical network. In control plane, each VTN is associated with a multi-topology or a Flex-Algo. And it also has its own forwarding plane resources and identifiers which provide VTN-specific packet processing.

### 3.1. VTN Operation

There are multiple modes of VTN operations to be supported as follows.

- o New VTN Binding: In realization, a NSC could request a set of underlay resources that are unaffected by other slice services. A

new VTN could be created and bound to a VPN per the network slice service and not used for any other VPNs.

- o VTN Sharing: A NSC could decide to use allocated underlay resources to meet the requirements of an IETF network slice. Therefore, an existing VTN instance can be reused and multiple VPNs in the VTN instance can share same VTN resources. In some cases, the properties of the existing VTN (e.g., link bandwidth) need modification.
- o VTN Deletion: If the NSC determines that no VPN service is using a VTN, the NSC can delete the VTN instance.
- o VTN Monitoring: The NSC could also use the VTN network model to track and monitor VTN resource status and usage.

### 3.2. VTN Network Modeling Design

A VTN network is modeled as network topology defined in [RFC8345] with augmentations. A new network type "vtn" is defined in this document. When a network topology data instance contains the vtn network type, it represents an instance of a VTN.

Each VTN consists of a set of nodes and a set of links. Each node and link have different attributes that represent the allocated resources or the operational status of the VTN network. VTN supports several resource partition methods, which are defined by 'interface-partition-capability' under a link, which can further be supported by FlexE and independent queue techniques.

The container "vtn" under 'network' of [RFC8345] defines global parameters for a VTN, which defines the specific control plane technique of the VTN and a unique "vtn-data-plane identifier" for data plane. And also, a color attribute for steering traffic, such as VPN traffic, into a VTN is also defined.

## 4. Description of the VTN Network YANG Module

The description of the VTN data nodes are as follows:

- o "vtn-id": Is an identifier that is used to uniquely identify the VTN instance within the network scope.
- o VTN allocation resources: The nodes and links represent the network resource allocated for a VTN instance. 'bandwidth-reservation' specifies the bandwidth allocated to a VTN network, or is overridden by the configuration of the VTN link.

'interface-partition-capability' specifies the resource partition capability of the physical interfaces associated with a VTN link.

- o VTN control plane: Based on the existing work in IETF, control plane mechanism of VTN could be implemented by Multi-Topology Routing (MTR) which defined in [RFC4915], [RFC5120], and [I-D.ietf-lsr-isis-sr-vtn-mt] or Flex-algo which is defined in [I-D.ietf-lsr-flex-algo]. With these control plane technologies, VTN nodes of each VTN instance will create their own VTN-specific forwarding tables.
- o VTN data plane: Defines the data plane mechanism and the VTN identifier of the network domain managed by the network controller. The data plane mechanism could be based on MPLS or IPv6 forwarding. "vtn-domain-identifier" is used to identify network resource of data plane that has been allocated for the VTN. In the case of IPv6 based forwarding, VTN data plane identifier is defined in [I-D.dong-6man-enhanced-vpn-vtn-id]. If a network slice service traverses multiple network domains, a global VTN identifier across the domains may be defined. For example, [I-D.li-6man-e2e-ietf-network-slicing] defines a IPv6 extension header to carry the global VTN identifier.
- o VTN steering policy: "vtn-color-id" is the color attribute of VTN for traffic steering.

## 5. VTN Yang Module Tree

```

module: ietf-vtn-ntw
augment /nw:networks/nw:network/nw:network-types:
  +---rw vtn!
augment /nw:networks/nw:network:
  +---rw vtn
    +---rw vtn-id?                               uint32
    +---rw vtn-name?                             string
    +---rw bandwidth-reservation
      | +---rw (bandwidth-type)?
      |   +---:(bandwidth-value)
      |     | +---rw bandwidth-value?           uint64
      |     +---:(bandwidth-percentage)
      |       +---rw bandwidth-percent?         rt-types:percentage
    +---rw control-plane
      | +---rw (vtn-cp-type)?
      |   +---:(flex-algo)
      |     | +---rw flex-algo
      |     |   +---rw flex-algo-id?           uint32
      |     +---:(multi-topology)
      |       +---rw multi-topology-id?         uint32
    +---rw data-plane
      | +---rw vtn-global-identifier?           uint32
      | +---rw domain-data-plane
      |   +---rw data-plane-type?               identityref
      |   +---rw vtn-domain-identifier?         uint32
    +---rw steering-policy
      +---rw vtn-color-id?                       uint32
augment /nw:networks/nw:network/nw:node:
  +---rw vtn
augment /nw:networks/nw:network/nt:link:
  +---rw vtn
    | +---rw interface-partition-capability?   identityref
    | +---rw bandwidth-reservation
    |   +---rw (bandwidth-type)?
    |     +---:(bandwidth-value)
    |       | +---rw bandwidth-value?           uint64
    |       +---:(bandwidth-percentage)
    |         +---rw bandwidth-percent?         rt-types:percentage
  +---ro statistics
    +---ro admin-status?                         te-types:te-admin-status
    +---ro oper-status?                         te-types:te-oper-status
    +---ro one-way-available-bandwidth?         rt-types:bandwidth-ieee-float32
    +---ro one-way-utilized-bandwidth?         rt-types:bandwidth-ieee-float32
    +---ro one-way-min-delay?                   uint32
    +---ro one-way-max-delay?                   uint32
    +---ro one-way-delay-variation?             uint32
    +---ro one-way-packet-loss?                 decimal64

```

## 6. VTN Yang Module

```
<CODE BEGINS> file "ietf-vtn-ntw@2021-06-04.yang"

module ietf-vtn-ntw {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vtn-ntw";
  prefix vtn-ntw;

  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-network-topology {
    prefix nt;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Traffic Engineering Common YANG Types";
  }
  import ietf-te-packet-types {
    prefix te-packet-types;
    reference
      "RFC 8776: Traffic Engineering Common YANG Types";
  }

  organization
    "IETF TEAS Working Group";
  contact
    "
      WG Web: <http://tools.ietf.org/wg/teas/>
      WG List:<mailto:teas@ietf.org>

      Editor: Bo Wu <lane.wubo@huawei.com>
             : Dhruv Dhody <dhruv.ietf@gmail.com>";
  description
    "This YANG module defines a network data module for
    VTN(Virtual Transport Network)";
```

```
revision 2021-06-04 {
  description
    "This is the initial version of VTN network yang module";
  reference
    "RFC XXX: YANG Data module for VTN network";
}

identity interface-partition-capability {
  description
    "Base identity for interface partition capability.";
}

identity flexe-partition {
  base interface-partition-capability;
  description
    "Identity for FlexE partition capability.";
}

identity queue-partition {
  base interface-partition-capability;
  description
    "Identity for queue partition capability.";
}

identity vtn-data-plane-type {
  description
    "Base identity for VTN data plane type.";
}

identity vtn-data-plane-vtn-ipv6 {
  base vtn-data-plane-type;
  description
    "Identity for VTN based packet forwarding of IPv6.";
}

identity vtn-data-plane-vtn-mpls {
  base vtn-data-plane-type;
  description
    "Identity for VTN based packet forwarding of MPLS.";
}

identity vtn-data-plane-sr-mpls {
  base vtn-data-plane-type;
  description
    "Identity for SR MPLS forwarding mechanism.";
}

identity vtn-data-plane-srv6 {
```

```
    base vtn-data-plane-type;
    description
        "Identity for SRv6 forwarding mechanism.";
}

/*
 * Groupings
 */

grouping traffic-steering-policy {
    description
        "Configuration of the traffic mapping policy.";
    container steering-policy {
        description
            "Policy set that matches to a VTN.";
        leaf vtn-color-id {
            type uint32;
            description
                "VTN color ID for VTN traffic steering";
        }
    }
}

grouping vtn-bandwidth-reservation {
    description
        "Grouping for VTN bandwidth reservation.";
    container bandwidth-reservation {
        description
            "Container for VTN bandwidth reservation.";
        choice bandwidth-type {
            description
                "Choice of bandwidth reservation type.";
            case bandwidth-value {
                leaf bandwidth-value {
                    type uint64;
                    units "bps";
                    description
                        "Bandwidth allocation for the VTN as absolute value.";
                }
            }
            case bandwidth-percentage {
                leaf bandwidth-percent {
                    type rt-types:percentage;
                    description
                        "Bandwidth allocation for the VTN as a percentage of a link.";
                }
            }
        }
    }
}
```

```
    }
  }

  grouping vtn-control-plane-attributes {
    description
      "VTN topology control plane attributes.";
    container control-plane {
      description
        "vtn control plane mechanism.";
      choice vtn-cp-type {
        description
          "Choice of vtn control plane.";
        case flex-algo {
          container flex-algo {
            description
              "A VTN could use flex-algo as a control plane
              mechanism.";
            leaf flex-algo-id {
              type uint32;
              description
                "flex-algo-id for VTN";
            }
          }
        }
        case multi-topology {
          description
            "A VTN could use MT (Multi-Topology) as a control
            plane mechanism.";
          leaf multi-topology-id {
            type uint32;
            description
              "MT-id for VTN";
          }
        }
      }
    }
  }

  grouping vtn-data-plane-attributes {
    description
      "Grouping for VTN topology data plane attributes.";
    container data-plane {
      description
        "VTN data plane mechanism.";
      leaf vtn-global-identifier {
        type uint32;
        description
          "The global VTN identifier for multi-domain is specified.";
      }
    }
  }
}
```



```
    }
    container domain-data-plane {
      description
        "VTN data plane mechanism per network domain.";
      leaf data-plane-type {
        type identityref {
          base vtn-data-plane-type;
        }
        description
          "Specifies the data plane forwarding mechanism of the VTN.
           The mechanism consists of VTN based Packet Forwarding or
           existing Segment Routing with MPLS data plane or IPv6 data
           plane.";
      }
      leaf vtn-domain-identifier {
        type uint32;
        description
          "The domain VTN identifier is specified for
           VTN based Packet Forwarding of a network domain.
           The forwarding plane could be with
           the MPLS Data Plane or IPv6";
        reference
          "draft-li-mpls-enhanced-vpn-vtn-id?
           Carrying Virtual Transport Network identifier
           in MPLS Packet
           draft-dong-6man-enhanced-vpn-vtn-id
           Carrying Virtual Transport Network Identifier
           in IPv6 Extension Header";
      }
    }
  }
}

grouping vtn-topology-attributes {
  description
    "VTN topology scope attributes.";
  container vtn {
    description
      "Containing VTN topology attributes.";
    leaf vtn-id {
      type uint32;
      description
        "VTN identifier";
    }
    leaf vtn-name {
      type string;
      description
        "VTN Name";
    }
  }
}
```

```
    }
    uses vtn-bandwidth-reservation;
    uses vtn-control-plane-attributes;
    uses vtn-data-plane-attributes;
    uses traffic-steering-policy;
  }
  // vtn
}

// vtn-node-attributes

grouping vtn-node-attributes {
  description
    "VTN node scope attributes.";
  container vtn {
    description
      "Containing VTN attributes.";
  }
}

// vtn-node-attributes

grouping vtn-link-attributes {
  description
    "VTN link scope attributes";
  container vtn {
    description
      "Containing VTN attributes.";
    leaf interface-partition-capability {
      type identityref {
        base interface-partition-capability;
      }
      description
        "Describes different resource partition type of a link.";
    }
    uses vtn-bandwidth-reservation;
  }
}

// vtn-statistics

grouping statistics-per-vtn {
  description
    "Statistics attributes per VTN.";
}

// vtn-node-statistics
```

```
grouping statistics-per-node {
  description
    "Statistics attributes per VTN node.";
}

// one-way-performance-metrics

grouping one-way-performance-bandwidth {
  description
    "Grouping for one-way performance bandwidth .";
  leaf one-way-available-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    units "bytes per second";
    default "0x0p0";
    description
      "Available bandwidth that is defined to be VTN link
       bandwidth minus bandwidth utilization.  For a
       bundled link, available bandwidth is defined to be the
       sum of the component link available bandwidths.";
  }
  leaf one-way-utilized-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    units "bytes per second";
    default "0x0p0";
    description
      "Bandwidth utilization that represents the actual
       utilization of the link (i.e. as measured in the router).
       For a bundled link, bandwidth utilization is defined to
       be the sum of the component link bandwidth
       utilizations.";
  }
}

// vtn-link-statistics

grouping vtn-statistics-per-link {
  description
    "Statistics attributes per VTN link.";
  container statistics {
    config false;
    description
      "Statistics for VTN link.";
    leaf admin-status {
      type te-types:te-admin-status;
      description
        "The administrative state of the link.";
    }
    leaf oper-status {
```

```
    type te-types:te-oper-status;
    description
      "The current operational state of the link.";
  }
  uses one-way-performance-bandwidth;
  uses te-packet-types:one-way-performance-metrics-packet;
}

augment "/nw:networks/nw:network/nw:network-types" {
  description
    "Defines the VTN topology type.";
  container vtn {
    presence "Indicates VTN topology";
    description
      "Its presence identifies the VTN type.";
  }
}

augment "/nw:networks/nw:network" {
  when 'nw:network-types/vtn-ntw:vtn' {
    description
      "Augment only for VTN topology.";
  }
  description
    "Augment VTN configuration and state.";
  uses vtn-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node" {
  when '../nw:network-types/vtn-ntw:vtn' {
    description
      "Augment only for VTN topology.";
  }
  description
    "Augment node configuration and state.";
  uses vtn-node-attributes;
}

augment "/nw:networks/nw:network/nt:link" {
  when '../nw:network-types/vtn-ntw:vtn' {
    description
      "Augment only for VTN topology.";
  }
  description
    "Augment link configuration and state.";
  uses vtn-link-attributes;
  uses vtn-statistics-per-link;
}
```

```
}  
}
```

<CODE ENDS>

## 7. Security Considerations

The YANG module defined in this document is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

**vtn-link:** A malicious client could attempt to remove a link from a topology, add a new link. In each case, the structure of the topology would be sabotaged, and this scenario could, for example, result in an VTN topology that is less than optimal.

The entries in the nodes above include the whole network configurations corresponding with the VTN, and indirectly create or modify the PE or P device configurations. Unexpected changes to these entries could lead to service disruption and/or network misbehavior.

## 8. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-vtn-ntw  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

This document requests to register a YANG module in the YANG Module Names registry [RFC7950].

Name: ietf-vtn-ntw  
Namespace: urn:ietf:params:xml:ns:yang:ietf-vtn-ntw  
Prefix: vtn-ntw  
Reference: RFC XXXX

## 9. Contributor

Zhenbin Li  
Huawei

Email: lizhenbin@huawei.com

Jie Dong  
Huawei

Email: jie.dong@huawei.com

## 10. References

### 10.1. Normative References

- [I-D.dong-6man-enhanced-vpn-vtn-id]  
Dong, J., Li, Z., Xie, C., and C. Ma, "Carrying Virtual Transport Network Identifier in IPv6 Extension Header", draft-dong-6man-enhanced-vpn-vtn-id-03 (work in progress), February 2021.
- [I-D.dong-idr-sr-policy-vtn]  
Dong, J., Hu, Z., and R. Pang, "BGP SR Policy Extensions for Virtual Transport Network", draft-dong-idr-sr-policy-vtn-00 (work in progress), October 2020.
- [I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-15 (work in progress), April 2021.

- [I-D.ietf-lsr-isis-sr-vtn-mt]  
Xie, C., Ma, C., Dong, J., and Z. Li, "Using IS-IS Multi-Topology (MT) for Segment Routing based Virtual Transport Network", draft-ietf-lsr-isis-sr-vtn-mt-00 (work in progress), March 2021.
- [I-D.ietf-spring-segment-routing-policy]  
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-11 (work in progress), April 2021.
- [I-D.li-6man-e2e-ietf-network-slicing]  
Li, Z. and J. Dong, "Encapsulation of End-to-End IETF Network Slice Information in IPv6", draft-li-6man-e2e-ietf-network-slicing-00 (work in progress), April 2021.
- [I-D.zhu-lsr-isis-sr-vtn-flexalgo]  
Zhu, Y., Dong, J., and Z. Hu, "Using Flex-Algo for Segment Routing based VTN", draft-zhu-lsr-isis-sr-vtn-flexalgo-02 (work in progress), February 2021.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <<https://www.rfc-editor.org/info/rfc4915>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

## 10.2. Informative References

- [I-D.dong-teas-enhanced-vpn-vtn-scalability]  
Dong, J., Li, Z., Qin, F., Yang, G., and J. N. Guichard,  
"Scalability Considerations for Enhanced VPN (VPN+)",  
draft-dong-teas-enhanced-vpn-vtn-scalability-02 (work in  
progress), February 2021.



[I-D.ietf-teas-enhanced-vpn]

Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", draft-ietf-teas-enhanced-vpn-07 (work in progress), February 2021.

[I-D.ietf-teas-ietf-network-slices]

Farrel, A., Gray, E., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", draft-ietf-teas-ietf-network-slices-00 (work in progress), April 2021.

[I-D.li-teas-e2e-ietf-network-slicing]

Li, Z. and J. Dong, "Framework for End-to-End IETF Network Slicing", draft-li-teas-e2e-ietf-network-slicing-00 (work in progress), April 2021.

#### Appendix A. Example VTN Network Model

Device could map

#### Authors' Addresses

Bo Wu  
Huawei Technologies  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: lana.wubo@huawei.com

Dhruv Dhody  
Huawei Technologies  
Divyashree Techno Park  
Bangalore, Karnataka 560066  
India

Email: dhruv.ietf@gmail.com