TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2021

T. Saad
V. Beeram
Juniper Networks

October 30, 2020

Realizing Network Slices in IP/MPLS Networks
draft-bestbar-teas-ns-packet-00

Abstract

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. Network slices need to operate in parallel with varying degrees of isolation 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 treatments 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. The proposed solution is agnostic to the path control technology used in the network slicing domain and allows service differentiation of traffic within a given network slice.

Table of Contents

1. Introduction

Network slicing allows a Service Provider, or a network operator to create independent and isolated 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. For example, certain network resource capabilities and functionality can be grouped together providing a self-contained unit (network slice) of varying size and complexity.

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

In a Differentiated Service (Diffserv) domain [RFC2475], packets requiring the same forwarding treatment 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.

Multiple network slices can be realized on top of a shared physical infrastructure network. A single network slice may also support multiple forwarding treatments or Diffserv classes that can be carried over the same logical network slice. This document proposes a solution that allows proper placement of paths and respective treatment of traffic traversing network slice resource(s) in IP/MPLS networks. The network slice traffic may be marked at slice boundary nodes with a Slice Selector (SS) to allow routers to apply a specific forwarding treatment that guarantees the slice Service Level Agreements (SLAs). Network slice traffic may further carry a Diffserv CS to allow differentiation of forwarding treatments for packets forwarded over the same network slice network resources.

For example, when using MPLS as a dataplane, it is possible to identify packets belonging to the same slice by carrying a global MPLS Slice Selector Label (SSL) in the MPLS label stack that identifies the slice in each packet. Additional Diffserv

classification may be indicated in the MPLS Traffic Class (TC) bits of the SSL to allow further differentiation of traffic treatments of traffic traversing the same slice network resources.

1.1. Terminology

The reader is expected to be familiar with the terminology specified in [I-D.nsdt-teas-ietf-network-slice-definition] and [I-D.draft-nsdt-teas-ns-framework-04].

The following terminology is used in the document:

Slicing capable node:
a node that supports one of the network slicing approaches described in this document.

Slicing incapable node:
a node that does not support one of the network slicing approaches described in this document.

Slice traffic:
traffic that is forwarded over network resource(s) associated with a specific network slice.

Slice path:
a path that is setup over network resource(s) associated with a specific network slice.

Slice-aware TE:
a mechanism for TE path selection that takes into account the available network resource(s) associated with a specific network slice.

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

CS: Class Selector

SS: Slice Selector

Slice-PHD: Slice Per-Hop Definition as described in Section 5.2

Slice-PHB: Slice Per-Hop Behavior as described in Section 5.2.3

SSL: Slice Selector Label as described in section Section 5.2.1

SSLI: Slice Selector Label Indicator

SLA: Service Level Agreement

SLO: Service Level Objective

Diffserv: Differentiated Services

DS-TE: Differentiated Services Traffic Engineering

MPLS: Multiprotocol Label Switching

LSP: Label Switched Path

LSR: Label Switching Router

LER: Label Edge Router

RSVP: Resource Reservation Protocol

TE: Traffic Engineering

SR: Segment Routing

1.3. Scope

The definition of Network Slice for use within the IETF and the characteristics of IETF Network Slice are specified in [I-D.draft-nsdt-teas-transport-slice-definition-04]. A framework for reusing IETF VPN and traffic-engineering technologies to realize IETF Network Slices is discussed in [I-D.draft-nsdt-teas-ns-framework-04].

This document provides a solution that addresses the network slice requirements in packet networks from a device and network resource level perspective based on DiffServ principles.

2. Network Resource Slicing Membership

A network slice can span multiple 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 operator domains. A network slice may include all or a sub-set of the physical nodes and links of an IP/MPLS network, and it may be comprised of dedicated and/or shared network resources (e.g. in terms of processing power, storage, and bandwidth) and may have varying degrees of isolation from the other network slices.

2.1. Dedicated Network Resources

Physical network resources may be fully dedicated to a specific network slice. For example, this allows traffic belonging to a slice to traverse the dedicated resources without network resource contention from traffic of another network slice. Dedicated network resource slicing allows for simple partitioning of the physical network resources into multiple isolated network slices without the need to distinguish packets traversing the dedicated network resources since only one slice traffic can use them.

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 partitioned among logical network slice(s). Shared network resources can be partitioned in the dataplane (for example by applying hardware policers and shapers), 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 of [I-D.draft-dt-teas-rfc3272bis-11]. The latter is the choice commonly used by IGP(s) 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 (LSDB)'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 link(s), and reduces the chance of congestion on traversed link(s).

To enable TE path placement, the link state is advertised with current reservation(s), 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 aware, the link state can carry per network slice state (e.g. per network slice link reservable bandwidth). This allows path computation to take into account the specific network resources available for a network slice when determining the path for a specific flow. In this case, we refer to the process of path placement and path provisioning has Slice-aware Traffic Engineering (Slice-aware TE).

4. Approaches to Network Resource Slicing

The partitioning of the shared network resources amongst multiple slices can be achieved in:

a) control plane only, or

b) data plane only, or

c) both control and data planes.

4.1. Data Plane Network Resource Slicing

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

When dataplane network resource slicing is required, packets need to be forwarded on the specific slice network resources and be applied a specific forwarding treatment that is dictated by the Slice Per-Hop Definition (Slice-PHD) (refer to Section 5.2 below) consumed by each device. A slice Selector (SS) MAY be carried in each slice packet to identify the slice that it belongs to.

The ingress node of a slice domain, in addition to marking packets with a Diffserv CS, MAY also add a SS to each slice packet. The transit node(s) within a slice domain MAY use the SS to associate packets with a slice and to determine the Slice Per Hop Behavior (Slice-PHB) that is applied to the packet (refer to Section 5.2.3 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 network slice.

When dataplane only network resource slicing is desirable, routers may rely on a network state independent view of the topology to determine the best path(s) to reach destination(s). In this case, the best path selection dictates the forwarding path of packets to the destination. The SS that is carried in each packet determines the specific Slice-PHB treatment for each slice along the selected path.

For example, Segment-Routing flexible algorithm may be deployed in a network to steer packet(s) on the lowest cumulative delay. A Slice-PHD may be used to enable the link(s) along the least latency path for dataplane slicing. Network slice packet(s) forwarded along the lowest delay path can carry the SS when forwarded along the least latency path. Transit nodes along the lowest delay path can inspect the SS and Diffserv CS to determine the Slice-PHB and the Diffserv class PHB to apply to packets before they are forwarded downstream.

4.2. Control Plane Network Resource Slicing

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 resource(s). 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 belonging to a slice. Each fraction of the physical link bandwidth can be represented as a logical link in a virtual topology that is used when determining path(s) in a specific slice. The per slice virtual network can be used by routing protocol(s), or by the ingress/PCE when computing slice aware path(s).

To perform network state dependent path computation in each slice, the resource reservation on each link needs to be slice aware (Slice-aware TE). Depending on the network Slice-PHD, a physical link may be part of one or more slice(s). Each such link may be sliced 'n' ways so that each slice will have certain network resources associated with it. The per slice network resource availability on link(s) are updated (and may eventually be advertised in the network) when new path(s) are placed in the network. The per slice resource reservation, in this case, can be maintained on each device(s) or be centralized on a resource reservation manager that holds link reservation state(s) on links in the network.

A number of network slice(s) can share the available network resource(s) allocated to each network slice amongst a group. In this case, a node can update the reservable bandwidth for each slice to

take into consideration the available bandwidth from other slice(s) in the same group.

For illustration purposes, the diagram below represents bandwidth isolation or sharing amongst a group of network slice(s). In Figure 1a, the network slices: Slice1, Slice2, Slice3 and Slice4 are not sharing any bandwidths between each other. In Figure 1b, the network slices: Slice1 and Slice2 can share the available bandwidth portion allocated to each amongst them. Similarly, Slice3 and Slice4 can share amongst themselves any available bandwidth allocated to them, but they cannot share available bandwidth allocated to Slice1 or Slice2. In both cases, the Max Reservable Bandwidth may exceed the actual physical link resource capacity to allow for over subscription.

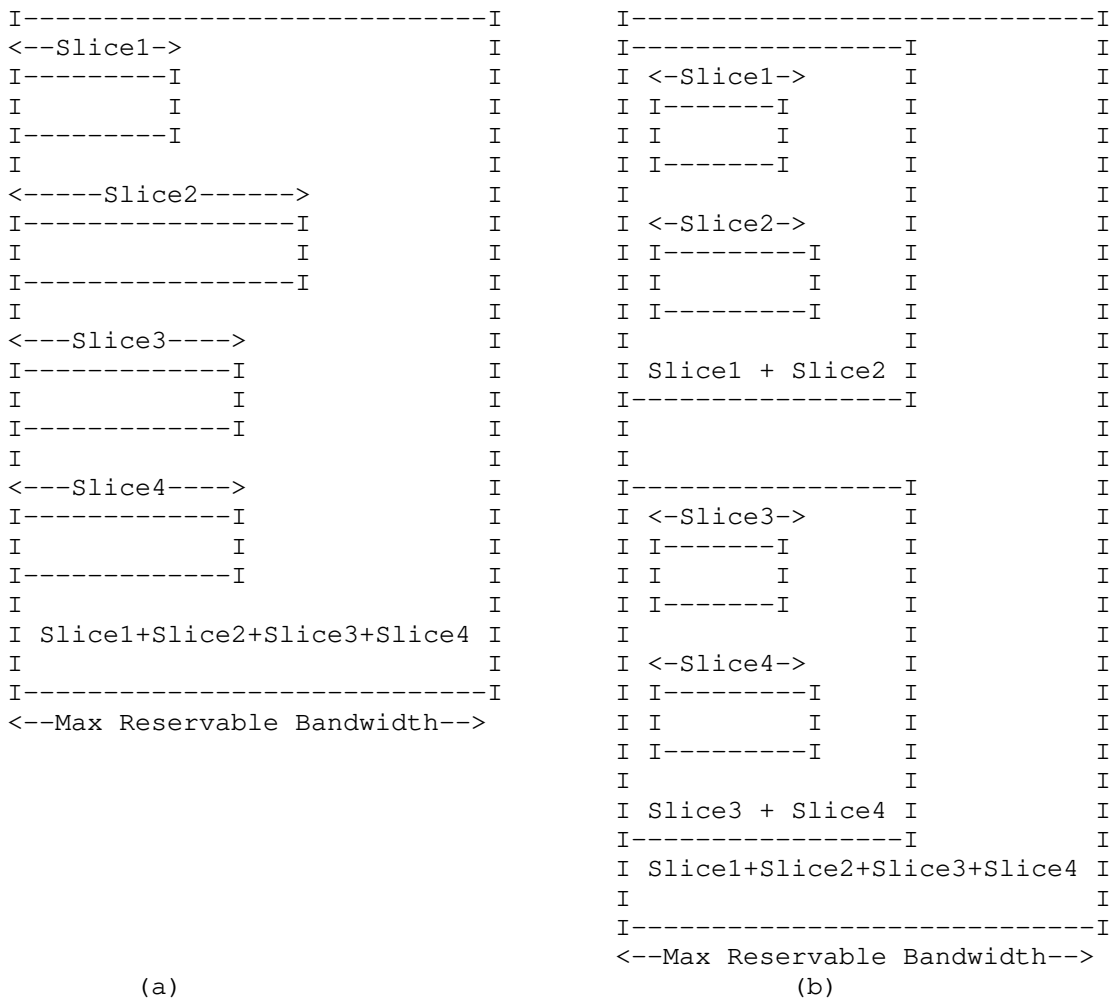


Figure 1: (a) bandwidth allocation(s) when no sharing between slice(s), (b) bandwidth allocation(s) when sharing between slice(s) of the same group.

4.3. Data and Control Plane Network Resource Slicing

In order to support strict guarantees and hard isolation between network slice(s), the network resource(s) can be partitioned in both the control plane and data plane.

The control plane partitioning allows the creation of customized topologies per slice that router(s) or a Path Computation Engine

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

The data plane partitioning protects slice traffic from network resource contention that occurs due to bursts in traffic from different slice(s) traversing the same shared network resource.

5. Network Slice 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 isolated from other network slices in terms of data, control or management planes.

The instantiation of a network slice may necessitate a network Slice Manager or service orchestrator that accepts a Service Layer Slice Intent as input and is translates it to a network wide device specific Slice-PHD as shown in Figure 2.

The Diffserv procedures may be employed within the same network slice to realize multiple classes of traffic belonging to the same slice.

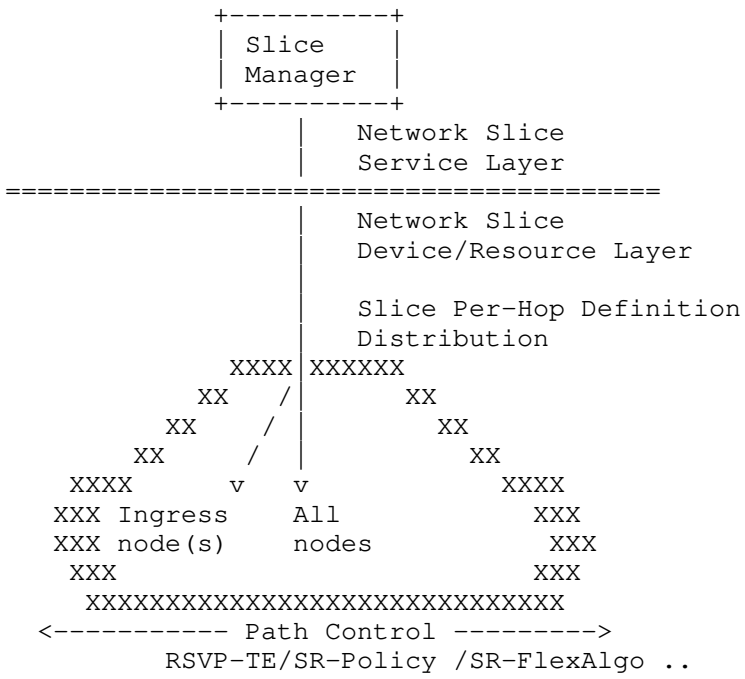


Figure 2: Network Slice instantiation model.

5.1. Slice Intent

A network slicing solution may be realized using a network slice service Layer, and a device/resource Layer. The service layer can be managed by a service orchestrator that exposes a north bound interface to slice consumers that can be used to convey the intent. Depending on the use cases and type of services for which the end-to-end slice is instantiated, multiple levels of control may be exposed to the tenants by a slice provider.

For example, network slicing provider may allow for a connectivity and data processing that is tailored to specific customer requirements. At the service layer, the consumer of a network slice expresses their intent for a particular network slice by specifying requirements rather than how a slice is realized. 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.draft-nsdt-teas-transport-slice-definition-04] for more details). These capabilities are always provided based on a Service Level Agreement (SLA) between the network slice consumer and the provider.

The network slice orchestrator is responsible for translating the network slice consumer intent into a Slice-PHD that can be instantiated by network elements at Device/Resource layer so that the network slice consumer requirements in terms of network characteristics are met.

5.2. Slice Per-Hop Definition

The high-level slice intent is consumed to produce a set of features and attributes that can be provisioned on network elements. The device level Slice-PHD includes attributes related to:

- Dataplane specific properties: This includes the SS, any firewall rules or flow-spec filters, and QoS profiles associated with the slice and any classes within it.

- Control plane specific properties: This includes guaranteed bandwidth, any network resource sharing amongst slice(s), and slice reservation preference to prioritize any reservations of a specific slice over others.

- Membership policies: This defines policies that dictate node/link/function network resource topology association for a specific slice.

There is a desire for flexibility in implementing network slices to support the services across networks consisting of products from multiple vendors, and may be grouped into disparate domains and using various path control technologies and tunnel types. It is expected that having a standardized data model for a Slice-PHD will facilitate the instantiation of a network slice on a network slicing capable node.

It is also possible to deliver a Slice-PHD to network devices using several mechanisms, including using 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).

5.2.1. Slice Data Plane Selector

A router MUST be able to identify a packet as belonging to a network slice before it can apply the proper forwarding treatment or PHB associated with the slice. One or more fields within the packet MAY be selected as a SS to do this.

Per Slice Forwarding Address:

One approach to distinguish packets targeted to a destination but belong to different slices is to assign multiple forwarding addresses (or multiple MPLS label bindings in the case of MPLS network) to the same destination - one for each slice the destination can be reached over. For example, when realizing a network slice over an IP dataplane, the same destination can be assigned multiple IP addresses (or multiple SRv6 locators in the case of SRv6 network) to enable steering of traffic to the same destination over multiple network slices.

Similarly, when an MPLS dataplane is used, [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'. In such case, the same destination can be assigned multiple MPLS label bindings corresponding to an LSP that traverses network resources of a specific slice towards the destination.

The specific slice forwarding address (or MPLS forwarding label) can be carried in the packet belonging to a network slice to allow (IP or MPLS) routers along the path to identify the packets and apply the respective per Slice-PHB and forwarding treatment. This

approach requires maintaining per slice 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 is reachable in as many as 'M' network slice(s), the node will have to assign and advertise reachability for 'N' unique forwarding addresses, or MPLS forwarding labels corresponding to the 'N' slices. Similarly, each node will have to assign a unique forwarding address (or MPLS forwarding label) for each of its 'K' adjacencies to enable strict steering over each. Consequently, the control plane at any node in the network will need to store as many as (N+K)*M states. In addition, a node will have to store and program (N+K)*M forwarding addresses or labels entries in its Forwarding Information Base (FIB) to realize this. Therefore, as 'N', 'K', and 'M' parameters increase, this approach will have scalability challenges both in the control and data planes.

Per Slice Selector:

A Slice Selector (SS) field can be carried in each packet to identify the packet belonging to a specific slice, independent of the forwarding address or MPLS forwarding label that is bound to the destination. Routers within the network slice domain can use the forwarding address (or MPLS forwarding label) to determine the forwarding path, and use the SS field in the packet to determine the specific Slice-PHB that gets applied on the packet. This approach allows better scale since it relies on a single forwarding address or MPLS label binding to be used independent of the number of network slices required along the path. In this case, the additional SS field will need to be carried, and maintained in each packet while it traverses the slice domain.

The SS can be carried in one of multiple fields within the packet, depending on the dataplane type used. For example in MPLS networks, the SS can be represented as a global MPLS label that is carried in the packet's MPLS label stack. All packets that belong to the same slice MAY carry the same SS label in the MPLS label stack. It is possible, as well, to have multiple SS label(s) point to the same Slice-PHB.

The MPLS SS Label (SSL) may appear in several positions in the MPLS label stack. For example, the MPLS SSL can be maintained at the top of the label stack while the packet is forwarded along the MPLS path. In this case, the forwarding at each hop is determined by the forwarding label that resides below the SSL. Figure 3

shows an example where the SSL appears at the top of MPLS label stack in a packet. PE1 is a network Slice edge node that receives the packet that needs to be steered over a slice specific MPLS Path. PE1 computes the SR Path composed of the Label Segment-List={9012, 9023}. It imposes a SSL=1001 corresponding to Slice-ID=1001 followed by the SR Path Segment-List. At P1, the top label sets the context of the packet to Slice-ID=1001. The forwarding of the packet is determined by inspecting the forwarding label (below the SSL) within the context of SSL.

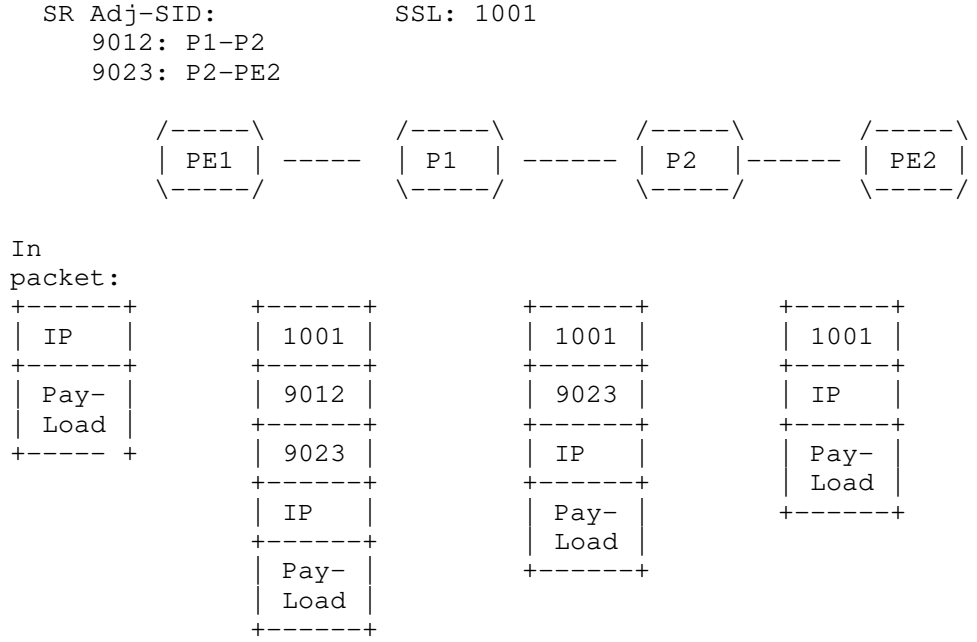


Figure 3: SSL at top of label stack.

The SSL can also reside at the bottom of the label stack. For example, the VPN service label may also be used as an SSL which allows steering of traffic towards one or more egress PEs over the same network slice. In such cases, one or more service labels MAY be mapped to the same slice. The same VPN label may also be allocated on all Egress PEs so it can serve as a single SSL for a specific network slice. Alternatively, a range of SSL (VPN labels) may be mapped to a single network slice to allow carrying multiple VPN(s) over the same network slice as shown in Figure 4.

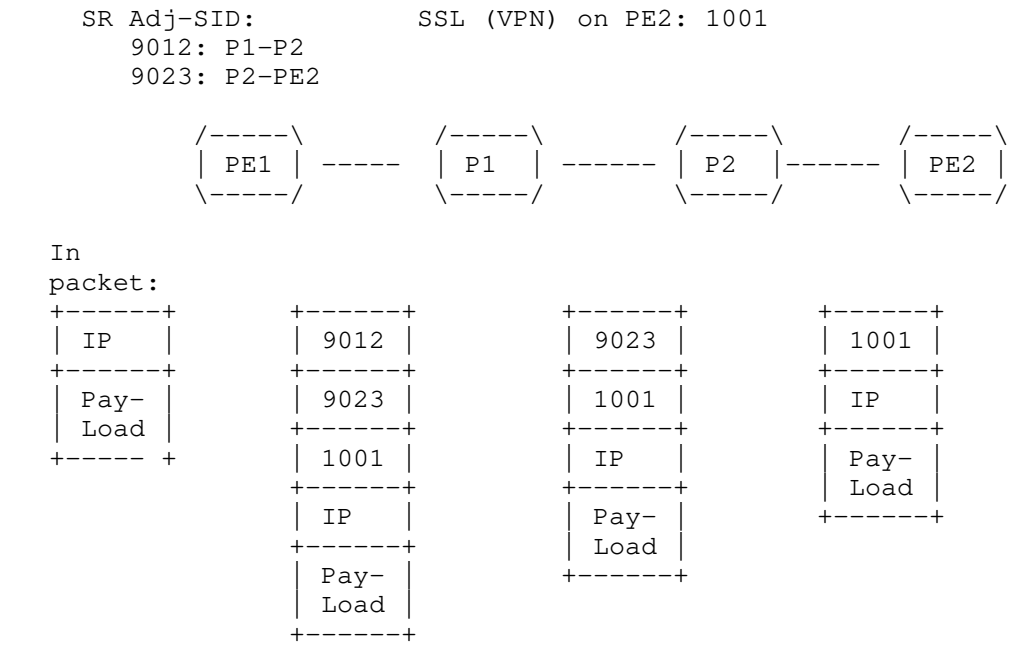


Figure 4: SSL or VPN label at bottom of label stack.

In some cases, the position of the SSL may not be at a fixed place in the MPLS label header. In this case, transit routers cannot expect the SSL at a fixed place in the MPLS label stack. This can be addressed by introducing a new Special Purpose Label from the label reserved space called a Slice Selector Label Indicator (SSLI). The slice network ingress boundary node, in this case, will need to impose at least two additional MPLS labels (SSLI + SSL) to identify the slice that the packets belong to as shown in Figure 5.

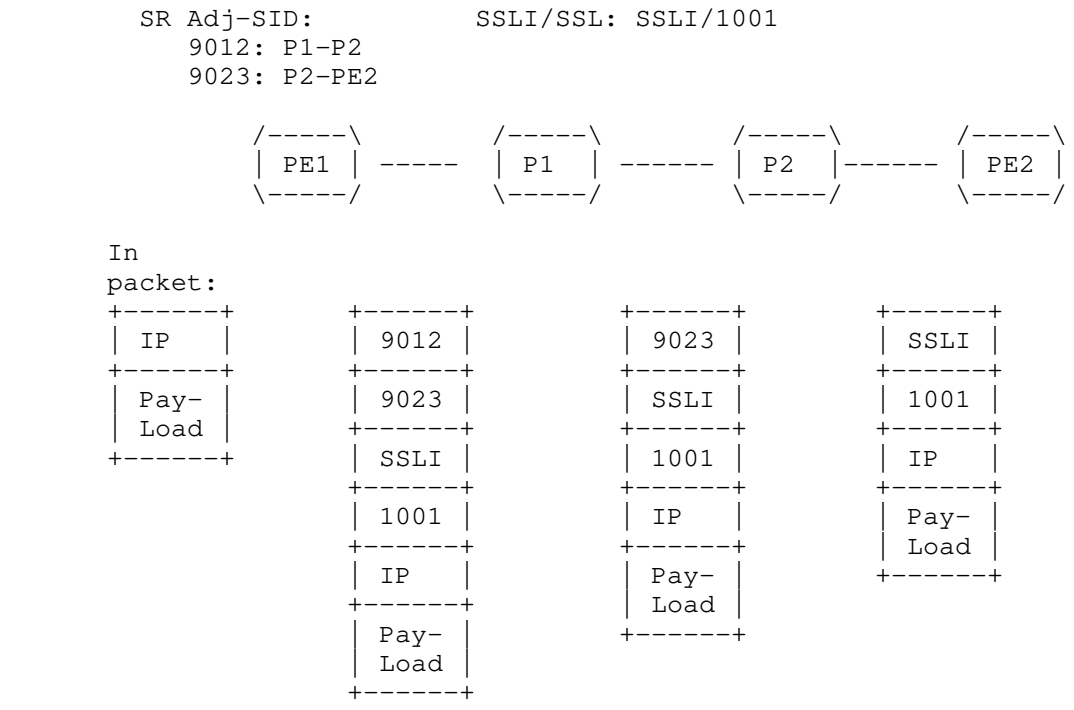


Figure 5: SSLI and bottom SSL at bottom of label stack.

When the slice is realized over an IP dataplane, the SSL 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.draft-filsfils-spring-srv6-stateless-slice-id-01].

5.2.2. Slice Network Resource Reservation

Bandwidth and network resource allocation strategies for network slicing 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 slices may also be desirable. For example, a slice may not always be using all of its reservable bandwidth; this allows other slices 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 network slices. When this occurs, preemption of some slice specific paths may be desirable to alleviate congestion. A preference based allocation scheme enables prioritization of slice paths that can be preempted.

Since network characteristics and its state can change over time, the per slice topology and its state also needs 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 network slice.

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

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

With Differentiated Services (Diffserv) it is possible to carry multiple service(s) 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 Aggregated (BAs) may be supported for a given Forwarding Equivalence Class (FEC) [RFC2475]. To support multiple services over the same network slice, a slice packet MAY also carry a Diffserv CS to identify the specific Diffserv forwarding treatment to be applied on the different service traffic belonging to the same slice.

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

5.2.4. Slice Topology Membership

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

The network slice topology can overlap or share a subset of links with another network slice topology. A number of policies or topology filters can be defined to limit the specific topology elements that belong to a network slice.

The Slice-PHD membership can carry the topology filtering policies. For example, such policies can leverage Resource Affinities as defined in [RFC2702] to include or exclude certain link(s) in a specific network slice topology. The Slice-PHD may also include a reference to a predefined topology (e.g. derived from from a Flexible Algorithm Definition (FAD) as defined in [I-D.ietf-lsr-flex-algo], or Multi-Topology ID as defined [RFC4915].

Alternatively, the topology filtering policies can specify specific link properties (such as delay, bandwidth capacity, security) to filter/include such link(s) in a network slice topology.

5.3. Network Slice Boundary

A network slice originates at the boundary of a network slice provider at edge node(s). Traffic that is steered over the network slice may traverse network slicing capable interior nodes, as well as, network slicing incapable interior nodes.

The network slice may compose of one or more administrative domain(s); for example, an organization's intranet or an ISP. The administration of the network is responsible for ensuring that adequate network resources are provisioned and/or reserved to support the SLAs offered by the network end-to-end.

5.3.1. Network Slice Edge Nodes

Network slicing edge nodes sit at the boundary of a network slice provider network and receive traffic that requires steering over network resources specific to a network slice. The slice edge nodes are responsible for identifying network slice 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 network slice it can be forwarded.

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

When dataplane slicing is required, the slice boundary nodes are responsible for adding a suitable SS onto packets that belong to specific network slices. In addition, edge nodes MAY mark the corresponding Diffserv CS to differentiate between different types of traffic carried over the same network slice.

5.3.2. Network Slice Interior Nodes

A network slice interior node receives slice traffic and MAY be able to identify the packets belonging to a specific network slice by inspecting the SS field carried inside each packet, or by inspecting other fields within the packet that may identify specific flows belonging to a specific network slice. For example when dataplane slicing is required, interior nodes can use the SS carried within the packet to apply the corresponding Slice-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 network slice, and allow differentiation of forwarding treatments for packets forwarded over the same network slice network resources.

5.3.3. Network Slice Incapable Nodes

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

When dataplane network slicing is desirable, 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 devices incapable of network slicing.

For example, when the SS is an MPLS label at the bottom of the MPLS label stack, packets can traverse over devices that are incapable of network slicing 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 network slicing incapable devices

towards the next device that supports network slicing as shown in Figure 6.

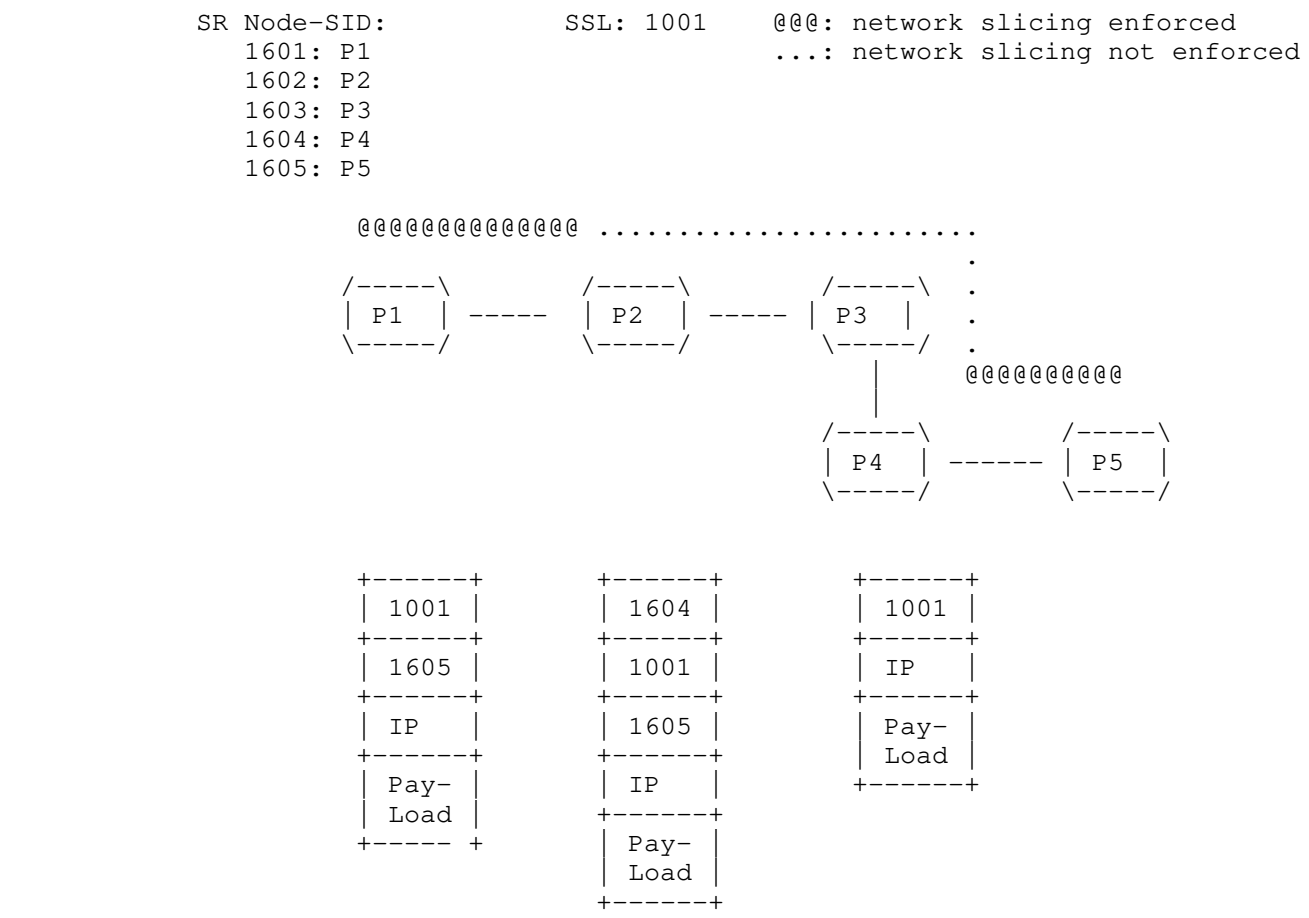


Figure 6: Extending network slice over slicing incapable device(s).

5.3.4. Combining Network Resource Slicing Approaches

It is possible to employ a combination of the approaches that were discussed in Section 4 to realize an end-to-end network slice. For example, data and control plane network resource slicing can be employed in parts of a network, while control plane only slicing can be employed in the other parts of the network. The Slice-aware path selection in such case can take into account the per slice available network resources. Packets carry a SS within them so the corresponding Slice-PHB can be enforced on the parts of the network that realize dataplane network resource slicing. The SS can be

maintained while traffic traverses nodes that do not enforce any dataplane slicing, and so slice PHB enforcement can resume once traffic traverses slicing capable nodes.

5.4. Slice Traffic Steering

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

For example, one or more (layer-2 or layer-3) VPN services can be directly mapped to paths established in a specific network slice. In this case, traffic that arrives on the Provider Edge (PE) router over external interface(s) can be directly mapped to a specific network slice path. External interface(s) can be further partitioned (e.g. using VLANs) to allow mapping one or more VLANs to specific network slice paths.

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

A third option that can also be used is to utilize a dataplane 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. This option allows for applying a rich set of rule(s) to identify specific packets to be mapped to a network slice. However, it requires dataplane network resources to be able to perform the additional checks in hardware.

6. Control Plane Extensions

Routing protocol(s) may need to be extended to carry additional per slice 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 document(s), and are outside the scope of this document.

The provisioning of the network slicing device Slice-PHD may need to be automated. Multiple options are possible to facilitate automation of provisioning a network slice definition on device(s) that are capable of network slicing.

For example, a YANG data model for the network Slice-PHD 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 network slicing on network device(s). The network Slice-PHD YANG data model may be defined in a separate document, and is outside the scope of this document.

7. Applicability to Path Control Technologies

The network slicing approach(s) described in this document are agnostic to the technology used to setup path(s) that carry networking slice traffic. Once feasible path(s) within a network slice are selected, it is possible to use RSVP-TE protocol [RFC3209] to setup or signal the LSP(s) that would be used to carry slice traffic. Specific extension(s) to RSVP-TE protocol to enable signaling of slice aware RSVP LSP(s) are outside the scope and will be tackled in a separate document(s).

Alternatively, Segment Routing (SR) [RFC8402] may be used and the feasible path(s) can realized by steering over specific segment(s) or segment-list(s) using an SR policy. Further detail(s) on how the approach(es) presented in this document can be realized over an SR network will be tackled in a separate document.

8. IANA Considerations

This document has no IANA actions.

9. Security Considerations

The main goal of network slicing is to allow for some level of isolation for 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 single network slice resource(s).

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

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

10. Acknowledgement

The authors would like to thank Krzysztof Szarkowicz, Swamy SRK, and Prabhu Raj Villadathu Karunakaran 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.draft-filsfils-spring-srv6-stateless-slice-id-01]
Filsfils, C., Clad, F., Camarillo, P., and K. Raza, "Stateless and Scalable Network Slice Identification for SRv6", draft-filsfils-spring-srv6-stateless-slice-id-01 (work in progress), July 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-13 (work in progress), October 2020.

[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>.

[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.draft-dt-teas-rfc3272bis-11]
Farrel, A., "Overview and Principles of Internet Traffic Engineering", draft-dt-teas-rfc3272bis-11 (work in progress), May 2020.

[I-D.draft-nsdt-teas-ns-framework-04]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

[I-D.draft-nsdt-teas-transport-slice-definition-04]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "IETF Definition of Transport Slice", draft-nsdt-teas-transport-slice-definition-04 (work in progress), September 2020.

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-00 (work in progress), October 2020.

[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., 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>.

[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

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2021

T. Saad
V. Beeram
Juniper Networks

November 2, 2020

YANG Data Model for Network Slice Per-Hop Definition
draft-bestbar-teas-yang-ns-phd-00

Abstract

This document defines a YANG data model for the management of Network Slice Per-Hop Definitions (Slice-PHDs) on network slicing capable nodes in IP/MPLS networks.

Requirements Language

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.

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 May 6, 2021.

Copyright Notice

Table of Contents

1. Introduction

Network slicing in IP/MPLS networks can be realized by partitioning the shared network resources in just the control plane or in just the data plane or in both the control and data planes [I-D.bestbar-teas-ns-packet]. The latter two approaches require the forwarding engine on each network slicing capable node to identify the traffic belonging to a specific slice and to apply the corresponding Slice Per-Hop Behavior (Slice-PHB) that determines the forwarding treatment of the packets belonging to the network slice. The identification of the slice that the packet belongs to and the corresponding forwarding treatment that needs to be applied to the packet is dictated by the Network Slice Per-Hop Definition (Slice-PHD) that is provisioned on each network slicing capable node.

This document defines a YANG data model for the provisioning and management of Slice-PHDs on network slicing capable nodes in IP/MPLS networks.

1.1. Terminology

The terminology for describing YANG data models is found in [RFC7950].

The reader is expected to be familiar with the terminology specified in [I-D.nsdt-teas-ietf-network-slice-definition], [I-D.nsdt-teas-ns-framework] and [I-D.bestbar-teas-ns-packet]. The term "Network Slice" used in this document must be interpreted as "IETF Network Slice" [I-D.nsdt-teas-ietf-network-slice-definition].

1.2. Tree Structure

A simplified graphical representation of the data model is presented in Appendix A of this document. The tree format defined in [RFC8340] is used for the YANG data model tree representation.

2. Network Slice Per-Hop Definition Model

2.1. Model Usage

The instantiation of a network slice may require a network slice controller that accepts a service layer slice customer intent as input and translates it to a network-wide consistent per-hop slice definition that is distributed to network slicing capable nodes. The specification of the service layer slice customer intent is outside the scope of this document. The data model defined in this document covers the per-hop slice definition that is consumed by the network slicing capable nodes.

2.2. Model Structure

The high-level model structure defined by this document is as shown below:

```
module: ietf-network-slice-phd
+--rw network-slicing!
  +--rw network-slice-phbs
  |  +--rw network-slice-phb* [id]
  |     .............
  +--rw network-slices
     +--rw network-slice* [name]
        |  .............
        +--rw slice-resource-reservation
        |  .............
        +--rw slice-selectors
        |  +--rw slice-selector* [id]
        |     .............
        +--rw slice-phb?                    ns-phb-ref
        |  .............
        +--rw slice-membership
           .............
```

In addition to the set of Slice-PHDs (network-slices), the model also includes a placeholder for the set of Slice-PHBs (network-slice-phbs) that are referenced by the Slice-PHDs.

2.3. Network Slice Per-Hop-Behaviors

The Slice-PHBs (network-slice-phbs) container carries a list of Slice-PHB (network-slice-phb) entries. Each of these entries can be referenced by one or more Slice-PHD. A Slice-PHB entry can either carry a reference to a generic PHB profile available on the node or carry a custom PHB profile. The custom PHB profile includes sufficient attributes to construct a slice specific Qos profile and any classes within it.

```
+--rw network-slice-phbs
|  +--rw network-slice-phb* [id]
|     +--rw id                                  uint16
|     +--rw (profile-type)?
|        +--:(profile)
|        |  +--rw profile?                      string
|        +--:(custom-profile)
|           .............
```

2.4. Network Slices

The Slice-PHDs are held in a container called 'network-slices'. Each network-slice entry is identified by a name and holds the set of per-hop attributes needed to instantiate the network slice. The four key elements of each network-slice entry are discussed in the following sub-sections.

2.4.1. Slice Resource Reservation

The 'slice-resource-reservation' container carries data nodes that are used to support Slice-aware Bandwidth Engineering. The data nodes in this container facilitate preference-based preemption of Slice-aware TE paths, sharing of resources amongst a group of slices and backup slice path bandwidth protection.

```
+--rw slice-resource-reservation
|  +--rw preference?                          uint16
|  +--rw (max-bw-type)?
|  |  +--:(bw-value)
|  |  |  +--rw maximum-bandwidth?            uint64
|  |  +--:(bw-percentage)
|  |     +--rw maximum-bandwidth-percent?
|  |              rt-types:percentage
|  +--rw shared-resource-groups*              uint32
|  +--rw protection
|     +--rw backup-slice-id?                    uint32
|     +--rw (backup-bw-type)?
|        +--:(backup-bw-value)
|        |  +--rw backup-bandwidth?             uint64
|        +--:(backup-bw-percentage)
|           +--rw backup-bandwidth-percent?
|                    rt-types:percentage
```

2.4.2. Slice Selectors

The 'slice-selectors' container carries a set of data plane field selectors which are used to identify the packets belonging to the given network slice. Each slice selector is uniquely identified by a 16-bit ID. The slice selector with the lowest ID is the default slice selector used by all the topological elements that are members of the given network slice. The other entries may be used when there is a need to override the default slice selector on some select topological elements.

```
            +--rw slice-selectors
            |  +--rw slice-selector* [id]
            |     +--rw id         uint16
            |     +--rw mpls
            |     |  +--rw (ss-mpls-type)?
            |     |     +--:(label-value)
            |     |     |  +--rw label?
            |     |     |  |       rt-types:mpls-label
            |     |     |  +--rw label-position?          identityref
            |     |     |  +--rw label-position-offset?   uint8
            |     |     +--:(label-ranges)
            |     |        +--rw label-range* [index]
            |     |           +--rw index                     string
            |     |           +--rw start-label?
            |     |           |       rt-types:mpls-label
            |     |           +--rw end-label?
            |     |           |       rt-types:mpls-label
            |     |           +--rw label-position?
            |     |           |       identityref
            |     |           +--rw label-position-offset?   uint8
            |     +--rw ipv4
            |     |  +--rw destination-prefix*   inet:ipv4-prefix
            |     +--rw ipv6
            |     |  +--rw (ss-ipv6-type)?
            |     |     +--:(ipv6-destination)
            |     |     |  +--rw destination-prefix*
            |     |     |          inet:ipv6-prefix
            |     |     +--:(ipv6-flow-label)
            |     |        +--rw slid-flow-labels
            |     |           +--rw slid-flow-label* [slid]
            |     |              +--rw slid       inet:ipv6-flow-label
            |     |              +--rw bitmask?   uint32
            |     +--rw acl-ref*   ns-acl-ref
```

2.4.3. Slice Per-Hop-Behavior

The Slice-PHB leaf carries a reference to the appropriate PHB that needs to be applied for the given network slice. Unless specified otherwise, this is the default Slice-PHB to be used by all the topological elements that are members of the given network slice.

```
            +--rw slice-phb?                    ns-phb-ref
```

2.4.4. Slice Membership

The 'slice-membership' container consists of a set of filtering policies that are used to determine which topological elements on the given node belong to the specific network slice. A filtering policy

could either reference a predefined topology or specify the rules to construct a customized topology using a set of include and exclude filters. The topological elements that satisfy the network slice membership criteria can optionally override the default Slice-PHB and/or the default slice selector.

```
            +--rw slice-membership
               +--rw filter-policies
                  +--rw filter-policy* [id]
                     +--rw id
                     |       uint16
                     +--rw (filter-type)?
                     |  +--:(topology-ref)
                     |  |  +--rw (topo-ref-type)?
                     |  |     +--:(algo-id)
                     |  |     |  +--rw algo-id?                  uint8
                     |  |     +--:(te-topo-id)
                     |  |        +--rw te-topology-identifier
                     |  |           +--rw provider-id?   te-global-id
                     |  |           +--rw client-id?     te-global-id
                     |  |           +--rw topology-id?
                     |  |                   te-topology-id
                     |  +--:(custom-topology)
                     |     +--rw include
                     |     |  +--rw link-affinity*   string
                     |     |  +--rw link-name*       string
                     |     |  +--rw node-prefix*     inet:ip-prefix
                     |     |  +--rw as*              inet:as-number
                     |     +--rw exclude
                     |        +--rw link-affinity*   string
                     |        +--rw link-name*       string
                     |        +--rw node-prefix*     inet:ip-prefix
                     |        +--rw as*              inet:as-number
                     +--rw slice-selector?
                     |       ns-ss-ref
                     +--rw slice-phb?
                             ns-phb-ref
```

2.5. YANG Module

```
<CODE BEGINS> file "ietf-network-slice-phd@2020-11-02"
module ietf-network-slice-phd {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-slice-phd";
  prefix "ns-phd";

  import ietf-inet-types {
    prefix "inet";
```

```
    reference
      "RFC 6991: Common YANG Data Types";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }

  import ietf-access-control-list {
    prefix "acl";
    reference
      "RFC 8519: YANG Data Model for Network Access Control Lists
       (ACLs)";
  }

  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:   <http://tools.ietf.org/wg/teas/>
     WG List:  <mailto:teas@ietf.org>

     Editor:   Vishnu Pavan Beeram
               <mailto:vbeeram@juniper.net>

     Editor:   Tarek Saad
               <mailto:tsaad@juniper.net>";

  description
    "This YANG module defines a data model for managing Network
     Slice Per-Hop Definitions (Slice-PHDs) on a network slicing
     capable node.

     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
        (https://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 "2020-11-02" {
        description "Initial revision.";
        reference
          "RFC XXXX: YANG Data Model for Network Slice Per-Hop
           Definitions (Slice-PHDs).";
      }


      /*
       * I D E N T I T I E S
       */


      /*
       * Identity - MPLS Slice Selector Label Position Type
       */

      identity ss-mpls-label-position-type {
        description
          "Base identity for the position of the MPLS label that is used
           for slice selection.";
      }

      identity ss-mpls-label-position-top {
        base ss-mpls-label-position-type;
        description
          "MPLS label that is used for slice selection is at the top of
           the label stack.";
      }

      identity ss-mpls-label-position-bottom {
        base ss-mpls-label-position-type;
        description
          "MPLS label that is used for slice selection is either at the
           bottom or at a specific offset from the bottom of the label
           stack.";
      }

      identity ss-mpls-label-position-indicator {
        base ss-mpls-label-position-type;
        description
```

```
        "MPLS label that is used for slice selection is immediately
         preceded by a special purpose slice indicator label in the
         label stack.";
    }

    /*
     * Identity - Slice-PHB Class Direction
     */

    identity s-phb-class-direction {
      description
        "Base identity for the direction of traffic to which the Slice
         PHB class profile is applied.";
    }

    identity s-phb-class-direction-in {
      base s-phb-class-direction;
      description
        "Slice PHB class profile is applied to incoming traffic.";
    }

    identity s-phb-class-direction-out {
      base s-phb-class-direction;
      description
        "Slice PHB class profile is applied to outgoing traffic.";
    }

    identity s-phb-class-direction-in-out {
      base s-phb-class-direction;
      description
        "Slice PHB class profile is applied to both incoming and
         outgoing directions of traffic.";
    }

    /*
     * Identity - Slice-PHB Class Priority
     */

    identity s-phb-class-priority {
      description
        "Base identity for the priority of the child class scheduler.";
    }

    identity s-phb-class-priority-low {
      base s-phb-class-drop-probability;
      description
        "Priority of the child class scheduler is low.";
    }
```

```
identity s-phb-class-priority-strict-high {
  base s-phb-class-drop-probability;
  description
    "Priority of the child class scheduler is strict-high.";
}

/*
 * Identity - Slice-PHB Class Drop Probability
 */

identity s-phb-class-drop-probability {
  description
    "Base identity for the drop probability applied to packets
     exceeding the CIR of the class queue.";
}

identity s-phb-class-drop-probability-low {
  base s-phb-class-drop-probability;
  description
    "Low drop probability applied to packets exceeding the CIR of
     the class queue.";
}

identity s-phb-class-drop-probability-medium {
  base s-phb-class-drop-probability;
  description
    "Medium drop probability applied to packets exceeding the CIR
     of the class queue.";
}

identity s-phb-class-drop-probability-high {
  base s-phb-class-drop-probability;
  description
    "High drop probability applied to packets exceeding the CIR of
     the class queue.";
}

/*
 * T Y P E D E F S
 */

typedef ns-acl-ref {
  type leafref {
    path "/acl:acls/acl:acl/acl:name";
  }
  description
    "This type is used to reference an ACL.";
}
```

```
  typedef ns-ss-ref {
    type leafref {
      path "/network-slicing/network-slices/network-slice/"
        + "slice-selectors/slice-selector/id";
    }
    description
      "This type is used to reference a Slice Selector (SS).";
  }

  typedef ns-phb-ref {
    type leafref {
      path "/network-slicing/network-slice-phbs/network-slice-phb/"
        + "id";
    }
    description
      "This type is used to reference a Slice Per-Hop Behavior
       (Slice-PHB).";
  }

  /*
   * G R O U P I N G S
   */

  /*
   * Grouping - Slice Selector MPLS: Label location specific fields
   */
  grouping ns-ss-mpls-label-location {
    description
      "Grouping for MPLS (SS) label location specific fields.";
    leaf label-position {
      type identityref {
        base ss-mpls-label-position-type;
      }
      description
        "MPLS label position - top, bottom with offset, Slice label
         indicator.";
    }
    leaf label-position-offset {
      when "derived-from-or-self(../label-position,"
        + "'ns-phd:ss-mpls-label-position-bottom')" {
        description
          "MPLS label position offset is relevant only when the
           label-position is set to 'bottom'.";
      }
      type uint8;
      description
        "MPLS label position offset.";
    }
```

```
    }

    /*
     * Grouping - Slice Selector (SS)
     */
    grouping ns-slice-selector {
      description
        "Grouping for Slice Selectors.";
      container slice-selectors {
        description
          "Container for Slice Selectors.";
        list slice-selector {
          key "id";
          description
            "List of Slice Selectors - this includes the default
             selector and others used for overriding the default.";
          leaf id {
            type uint16;
            description
              "A 16-bit ID to uniquely identify the Slice Selector.
               The Slice Selector with the lowest ID is the default
               selector.";
          }
          container mpls {
            description
              "Container for MPLS Slice Selector.";
            choice ss-mpls-type {
              description
                "Choices for MPLS Slice Selector.";
              case label-value {
                leaf label {
                  type rt-types:mpls-label;
                  description
                    "MPLS Slice Selector Label is explicitly
                     specified.";
                }
                uses ns-ss-mpls-label-location;
              }
              case label-ranges {
                list label-range {
                  key "index";
                  unique "start-label end-label";
                  description
                    "MPLS Slice Selector Label is picked from a
                     specified set of label ranges.";
                  leaf index {
                    type string;
                    description
```

```
                    "A string that uniquely identifies a label
                     range.";
                }
                leaf start-label {
                  type rt-types:mpls-label;
                  must '. <= ../end-label' {
                    error-message
                      "The start-label must be less than or equal "
                    + "to end-label";
                  }
                  description
                    "Label-range start.";
                }
                leaf end-label {
                  type rt-types:mpls-label;
                  must '. >= ../start-label' {
                    error-message
                      "The end-label must be greater than or equal "
                    + "to start-label";
                  }
                  description
                    "Label-range end.";
                }
                uses ns-ss-mpls-label-location;
              }
            }
          }
        }
        container ipv4 {
          description
            "Container for IPv4 Slice Selector.";
          leaf-list destination-prefix {
            type inet:ipv4-prefix;
            description
              "IPv4 Slice Selector is picked from a specified set of
               IPv4 destination prefixes.";
          }
        }
        container ipv6 {
          description
            "Container for IPv6 Slice Selector.";
          choice ss-ipv6-type {
            description
              "Choices for IPv6 Slice Selector.";
            case ipv6-destination {
              leaf-list destination-prefix {
                type inet:ipv6-prefix;
                description
```

```
                    "IPv6 Slice Selector is picked from a specified
                     set of IPv6 destination prefixes.";
                }
              }
              case ipv6-flow-label {
                container slid-flow-labels {
                  description
                    "Container for a set of Slice IDs that are
                     encoded within the flow label.";
                  list slid-flow-label {
                    key "slid";
                    description
                      "IPv6 Slice Selector is picked from a set of
                       Slice IDs that are encoded within the flow
                       label.";
                    leaf slid {
                      type inet:ipv6-flow-label;
                      description
                        "Slice ID encoded inside the IPv6 flow label.";
                    }
                    leaf bitmask {
                      type uint32;
                      description
                        "Bitmask to extract the encoded Slice ID from
                         the IPv6 flow label.";
                    }
                  }
                }
              }
            }
            leaf-list acl-ref {
              type ns-acl-ref;
              description
                "Slice Selection is done based on the specified list of
                 ACLs.";
            }
          }
        }
      }

      /*
       * Grouping - Slice Resource Reservation
       */
      grouping ns-slice-resource-reservation {
        description
          "Grouping for Slice Resource Reservation.";
        container slice-resource-reservation {
```

```
        description
          "Container for Slice Resource Reservation.";
        leaf preference {
          type uint16;
          description
            "Slice control plane preference. A higher preference
             indicates a more favorable slice resource
             reservation than a lower preference.";
        }
        choice max-bw-type {
          description
            "Choice of maximum bandwidth specification.";
          case bw-value {
            leaf maximum-bandwidth {
              type uint64;
              description
                "The maximum bandwidth allocated to a network slice on
                 the network resources - specified as absolute value.";
            }
          }
          case bw-percentage {
            leaf maximum-bandwidth-percent {
              type rt-types:percentage;
              description
                "The maximum bandwidth allocated to a network slice on
                 the network resources - specified as percentage of
                 link capacity.";
            }
          }
        }
        leaf-list shared-resource-groups {
          type uint32;
          description
            "List of shared resource groups that a network slice
             shares its allocated resources with.";
        }
        container protection {
          description
            "Container for network slice protection reservation.";
          leaf backup-slice-id {
            type uint32;
            description
              "The Slice ID that identifies the network slice used
               for backup paths that protect primary paths in a
               specific network slice.";
          }
          choice backup-bw-type {
            description
```

```
            "Choice of backup bandwidth specification.";
          case backup-bw-value {
            leaf backup-bandwidth {
              type uint64;
              description
                "The maximum bandwidth on a network resource that
                 is allocated for backup traffic - specified as
                 absolute value.";
            }
          }
          case backup-bw-percentage {
            leaf backup-bandwidth-percent {
              type rt-types:percentage;
              description
                "The maximum bandwidth on a network resource that
                 is allocated for backup traffic - specified as
                 percentage of the link capacity.";
            }
          }
        }
      }
    }

    /*
     * Grouping - Slice-PHB
     */
    grouping ns-slice-phb {
      description
        "Grouping for Slice-PHB.";
      leaf slice-phb {
        type ns-phb-ref;
        description
          "Reference to a specific Slice-PHB from the list global
           Slice-PHBs.";
      }
    }

    /*
     * Grouping - Slice default profile override
     */
    grouping ns-slice-default-profile-override {
      description
        "Grouping of fields that are used to override the default
         profile of the network slice.";
      leaf slice-selector {
        type ns-ss-ref;
        description
```

```
          "Reference to a specific Slice Selector.";
      }
      uses ns-slice-phb;
    }

    /*
     * Grouping - Slice membership filter: Topology reference
     */
    grouping ns-slice-membership-topo-ref {
      description
        "Grouping for topology reference slice membership filter.";
      choice topo-ref-type {
        description
          "Choice of topology reference.";
        case algo-id {
          leaf algo-id {
            type uint8;
            description
              "Algorithm ID.";
          }
        }
        case te-topo-id {
          uses te-types:te-topology-identifier;
        }
      }
    }

    /*
     * Grouping - Slice membership filters: Custom topology
     */
    grouping ns-slice-membership-custom-topo {
      description
        "Grouping for custom topology slice membership filters.";
      leaf-list link-affinity {
        type string;
        description
          "Match-filter is a list of link affinities.";
      }
      leaf-list link-name {
        type string;
        description
          "Match-filter is a list of link names.";
      }
      leaf-list node-prefix {
        type inet:ip-prefix;
        description
          "Match-filter is a list of node IDs.";
      }
```

```
      leaf-list as {
        type inet:as-number;
        description
          "Match-filter is a list of AS numbers.";
      }
    }

    /*
     * Grouping - Slice membership filters
     */
    grouping ns-slice-membership-filters {
      description
        "Grouping for Slice Membership filters.";
      choice filter-type {
        description
          "Choice of filter type.";
        case topology-ref {
          uses ns-slice-membership-topo-ref;
        }
        case custom-topology {
          container include {
            description
              "Include policies.";
            uses ns-slice-membership-custom-topo;
          }
          container exclude {
            description
              "Exclude policies.";
            uses ns-slice-membership-custom-topo;
          }
        }
      }
    }

    /*
     * Grouping - Slice Membership
     */
    grouping ns-slice-membership {
      description
        "Grouping for 'Slice Membership'.";
      container slice-membership {
        description
          "Container for Slice Membership.";
        container filter-policies {
          description
            "Container for topology filtering policies.";
          list filter-policy {
            key "id";
```

```
            description
              "List of topology filtering policies.";
            leaf id {
              type uint16;
              description
                "A 16-bit ID that uniquely identifies the topology
                 filtering policy.";
            }
            uses ns-slice-membership-filters;
            uses ns-slice-default-profile-override;
          }
        }
      }
    }

    /*
     * Grouping - Network Slice Per-Hop Behaviors (Slice-PHBs)
     */
    grouping ns-phbs {
      description
        "Grouping for Slice-PHBs.";
      container network-slice-phbs {
        description
          "Container for Slice-PHBs.";
        list network-slice-phb {
          key "id";
          description
            "List of Slice-PHBs.";
          leaf id {
            type uint16;
            description
              "A 16-bit ID that uniquely identifies the Slice-PHB.";
          }
          choice profile-type {
            description
              "Choice of PHB profile type.";
            case profile {
              description
                "Generic PHB profile available on the network
                 element.";
              leaf profile {
                type string;
                description
                  "Generic PHB profile identifier.";
              }
            }
            case custom-profile {
              description
```

```
        "Custom PHB profile.";
      choice guaranteed-rate-type {
        description
          "Guaranteed rate is the committed information rate
           (CIR) of the Slice. The guaranteed rate also
           determines the amount of excess (extra) bandwidth
           that a group of Slices can share. Extra bandwidth
           is allocated among the group in proportion to the
           guaranteed rate of each Slice.";
        case rate {
          leaf guaranteed-rate {
            type uint64;
            description
              "Guaranteed rate specified as absolute value.";
          }
        }
        case percentage {
          leaf guaranteed-rate-percent {
            type rt-types:percentage;
            description
              "Guaranteed rate specified in percentage.";
          }
        }
      }
      choice shaping-rate-type {
        description
          "Shaping rate is the maximum bandwidth of the slice,
           or the peak information rate (PIR) of a Slice.";
        case rate {
          leaf shaping-rate {
            type uint64;
            description
              "Shaping rate specified as absolute value.";
          }
        }
        case percentage {
          leaf shaping-rate-percent {
            type rt-types:percentage;
            description
              "Shaping rate specified in percentage.";
          }
        }
      }
      container classes {
        description
          "Container for classes.";
        list class {
          key class-id;
```

```
                    description
                      "List of classes.";
                    leaf class-id {
                      type string;
                      description
                        "A string to uniquely identify a class.";
                    }
                    leaf direction {
                      type identityref {
                        base s-phb-class-direction;
                      }
                      description
                        "Class direction.";
                    }
                    leaf priority {
                      type identityref {
                        base s-phb-class-priority;
                      }
                      description
                        "Priority of the class scheduler.  Only one Slice
                         class queue can be set as a strict-high priority
                         queue. Strict-high priority allocates the
                         scheduled bandwidth to the queue before any
                         other queue receives bandwidth. Other queues
                         receive the bandwidth that remains after the
                         strict-high queue has been serviced.";
                    }
                    choice guaranteed-rate-type {
                      description
                        "Guaranteed Rate is the Committed information
                         rate (CIR) of Slice class - specified as
                         absolute value or percentage.";
                      case rate {
                        leaf guaranteed-rate {
                          type uint64;
                          description
                            "Guaranteed rate specified as absolute
                             value.";
                        }
                      }
                      case percentage {
                        leaf guaranteed-rate-percent {
                          type rt-types:percentage;
                          description
                            "Guaranteed rate specified in percentage.";
                        }
                      }
                    }
```

```
                leaf drop-probability {
                  type identityref {
                    base s-phb-class-drop-probability;
                  }
                  description
                    "Drop probability applied to packets exceeding
                     the CIR of the class queue.";
                }
                choice maximum-bandwidth-type {
                  description
                    "Maximum bandwidth is the Peak information
                     rate (PIR) of Slice class - specified as
                     absolute value or percentage.";
                  case rate {
                    leaf maximum-bandwidth {
                      type uint64;
                      description
                        "Maximum bandwidth specified as absolute
                         value.";
                    }
                  }
                  case percentage {
                    leaf maximum-bandwidth-percent {
                      type rt-types:percentage;
                      description
                        "Maximum bandwidth specified as percentage.";
                    }
                  }
                }
                choice delay-buffer-size-type {
                  description
                    "Size of the queue buffer as a percentage of the
                     dedicated buffer space - specified as value or
                     percentage.";
                  case value {
                    leaf delay-buffer-size {
                      type uint64;
                      description
                        "Delay buffer size.";
                    }
                  }
                  case percentage {
                    leaf delay-buffer-size-percent {
                      type rt-types:percentage;
                      description
                        "Delay buffer size specified as percentage.";
                    }
                  }
```

```
                    }
                  }
                }
              }
            }
          }
        }
      }

      /*
       * Grouping - Network Slice Per-Hop Definitions (Slice-PHDs)
       */
      grouping ns-entries {
        description
          "Grouping for Slice-PHDs.";
        container network-slices {
          description
            "Container for Slice-PHD entries (network-slices).";
          list network-slice {
            key "name";
            unique "id";
            description
              "List of network slices.";
            leaf name {
              type string;
              description
                "A string that uniquely identifies the network slice.";
            }
            leaf id {
              type uint32;
              description
                "A 32-bit ID that uniquely identifies the network
                 slice.";
            }
            uses ns-slice-resource-reservation;
            uses ns-slice-selector;
            uses ns-slice-phb;
            uses ns-slice-membership;
          }
        }
      }

      /*
       * Top-level container - Network Slicing
       */
      container network-slicing {
        presence "Enable network slicing.";
        description
```

```
        "Top-level container for network slicing specific constructs
         on a network slicing capable node.";
      uses ns-phbs;
      uses ns-entries;
    }
  }
  <CODE ENDS>
```

3. Acknowledgements

The authors would like to thank Krzysztof Szarkowicz for his input from discussions.

4. 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

5. IANA Considerations

This document registers the following 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-phd
Registrant Contact: The TEAS WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-network-slice-phd
namespace: urn:ietf:params:xml:ns:yang:ietf-network-slice-phd
prefix: ns-phd
reference: RFCXXXX

6. 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.

The data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default) 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:

* "/network-slicing/network-slice-phbs": This subtree specifies the configurations for network slice per-hop behaviors. By manipulating these data nodes, a malicious attacker may cause unauthorized and improper behavior to be provided for the slice traffic on the network element.

* "/network-slicing/network-slices": This subtree specifies the configurations for network slices on a given network element. By manipulating these data nodes, a malicious attacker may cause unauthorized and improper behavior to be provided for the slice traffic on the network element.

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:

* "/network-slicing/network-slice-phbs": Unauthorized access to this subtree can disclose the network slice PHBs defined on the network element.

* "/network-slicing/network-slices": Unauthorized access to this subtree can disclose the network slice definitions on the network element.

7. References

7.1. Normative References

[I-D.bestbar-teas-ns-packet]
Saad, T. and V. Beeram, "Realizing Network Slices in IP/MPLS Networks", draft-bestbar-teas-ns-packet-00 (work in progress), October 2020.

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-00 (work in progress), October 2020.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

[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>.

[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>.

7.2. Informative References

[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>.

Appendix A. Complete Model Tree Structure

```
module: ietf-network-slice-phd
  +--rw network-slicing!
     +--rw network-slice-phbs
     |  +--rw network-slice-phb* [id]
     |     +--rw id                                    uint16
     |     +--rw (profile-type)?
     |        +--:(profile)
     |        |  +--rw profile?                        string
     |        +--:(custom-profile)
     |           +--rw (guaranteed-rate-type)?
     |           |  +--:(rate)
     |           |  |  +--rw guaranteed-rate?          uint64
     |           |  +--:(percentage)
     |           |     +--rw guaranteed-rate-percent?
     |           |             rt-types:percentage
     |           +--rw (shaping-rate-type)?
     |           |  +--:(rate)
     |           |  |  +--rw shaping-rate?             uint64
     |           |  +--:(percentage)
     |           |     +--rw shaping-rate-percent?
     |           |             rt-types:percentage
     |           +--rw classes
     |              +--rw class* [class-id]
```

```
|                  +--rw class-id
|                  |       string
|                  +--rw direction?
|                  |       identityref
|                  +--rw priority?
|                  |       identityref
|                  +--rw (guaranteed-rate-type)?
|                  |  +--:(rate)
|                  |  |  +--rw guaranteed-rate?
|                  |  |          uint64
|                  |  +--:(percentage)
|                  |     +--rw guaranteed-rate-percent?
|                  |             rt-types:percentage
|                  +--rw drop-probability?
|                  |       identityref
|                  +--rw (maximum-bandwidth-type)?
|                  |  +--:(rate)
|                  |  |  +--rw maximum-bandwidth?
|                  |  |          uint64
|                  |  +--:(percentage)
|                  |     +--rw maximum-bandwidth-percent?
|                  |             rt-types:percentage
|                  +--rw (delay-buffer-size-type)?
|                     +--:(value)
|                     |  +--rw delay-buffer-size?
|                     |          uint64
|                     +--:(percentage)
|                        +--rw delay-buffer-size-percent?
|                                rt-types:percentage
|
+--rw network-slices
   +--rw network-slice* [name]
      +--rw name                              string
      +--rw id?                               uint32
      +--rw slice-resource-reservation
      |  +--rw preference?                              uint16
      |  +--rw (max-bw-type)?
      |  |  +--:(bw-value)
      |  |  |  +--rw maximum-bandwidth?                 uint64
      |  |  +--:(bw-percentage)
      |  |     +--rw maximum-bandwidth-percent?
      |  |             rt-types:percentage
      |  +--rw shared-resource-groups*                  uint32
      |  +--rw protection
      |     +--rw backup-slice-id?                        uint32
      |     +--rw (backup-bw-type)?
      |        +--:(backup-bw-value)
      |        |  +--rw backup-bandwidth?                 uint64
      |        +--:(backup-bw-percentage)
```

```
|           +--rw backup-bandwidth-percent?
|                   rt-types:percentage
+--rw slice-selectors
|  +--rw slice-selector* [id]
|     +--rw id         uint16
|     +--rw mpls
|     |  +--rw (ss-mpls-type)?
|     |     +--:(label-value)
|     |     |  +--rw label?
|     |     |  |       rt-types:mpls-label
|     |     |  +--rw label-position?          identityref
|     |     |  +--rw label-position-offset?   uint8
|     |     +--:(label-ranges)
|     |        +--rw label-range* [index]
|     |           +--rw index                     string
|     |           +--rw start-label?
|     |           |       rt-types:mpls-label
|     |           +--rw end-label?
|     |           |       rt-types:mpls-label
|     |           +--rw label-position?
|     |           |       identityref
|     |           +--rw label-position-offset?   uint8
|     +--rw ipv4
|     |  +--rw destination-prefix*   inet:ipv4-prefix
|     +--rw ipv6
|     |  +--rw (ss-ipv6-type)?
|     |     +--:(ipv6-destination)
|     |     |  +--rw destination-prefix*
|     |     |          inet:ipv6-prefix
|     |     +--:(ipv6-flow-label)
|     |        +--rw slid-flow-labels
|     |           +--rw slid-flow-label* [slid]
|     |              +--rw slid       inet:ipv6-flow-label
|     |              +--rw bitmask?   uint32
|     +--rw acl-ref*   ns-acl-ref
+--rw slice-phb?                    ns-phb-ref
+--rw slice-membership
   +--rw filter-policies
      +--rw filter-policy* [id]
         +--rw id
         |       uint16
         +--rw (filter-type)?
         |  +--:(topology-ref)
         |  |  +--rw (topo-ref-type)?
         |  |     +--:(algo-id)
         |  |     |  +--rw algo-id?                  uint8
         |  |     +--:(te-topo-id)
         |  |        +--rw te-topology-identifier
```

```
|   |           +--rw provider-id?   te-global-id
|   |           +--rw client-id?     te-global-id
|   |           +--rw topology-id?
|   |                   te-topology-id
|   +--:(custom-topology)
|      +--rw include
|      |  +--rw link-affinity*   string
|      |  +--rw link-name*       string
|      |  +--rw node-prefix*     inet:ip-prefix
|      |  +--rw as*              inet:as-number
|      +--rw exclude
|         +--rw link-affinity*   string
|         +--rw link-name*       string
|         +--rw node-prefix*     inet:ip-prefix
|         +--rw as*              inet:as-number
+--rw slice-selector?
|       ns-ss-ref
+--rw slice-phb?
        ns-phb-ref
```

Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

TEAS
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

LM. Contreras
Telefonica
R. Rokui
Nokia
J. Tantsura
Apstra
B. Wu
Huawei
X. Liu
Volta
D. Dhody
Huawei
S. Belloti
Nokia

November 2, 2020

IETF Network Slice Controller and its associated data models
draft-contreras-teas-slice-controller-models-00

Abstract

This document describes the major functional components of an IETF Network Slice Controller (NSC) as well as references the data models required for supporting the requests of IETF network slices and their realization.

Table of Contents

1. Introduction

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.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework] is primarily used here. Consensus to use "IETF Network Slice" term has been reached.

The generic idea of network slicing intends to provide tailored end-to-end network capabilities to customers in the way that they could be perceived as a dedicated network, despite the fact that it makes use of shared physical infrastructure facilities.

Among the capabilities mentioned, connectivity of different parts of a network slice with particular characteristics play a central role. Thus, the concept of IETF Network Slice, realized by any of the IETF technologies, emerges as complementary but essential part of an end-to-end network slice.

In order to facilitate the request, realization and lifecycle control and management of a transport slice, a new element named IETF Network

Slice Controller (NSC) is being proposed in [I-D.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework].

The NSC from its North Bound Interface (NBI) exposes set of APIs that allow a higher level system to request an end-to-end transport slice. It receives the request of enablement of an IETF Network Slice by a customer (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 request and the management of its lifecycle. Figure 1 presents a high-level view of the TSC.

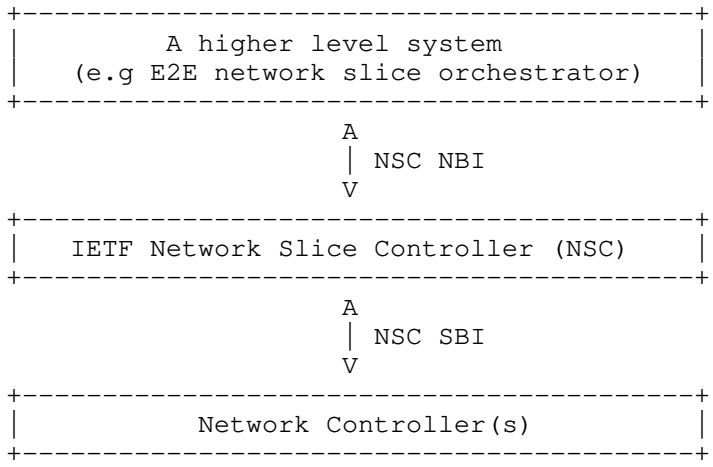


Figure 1: Interface of Transport Slice Controller

This memo describes the characteristics of the NSC as well as a detailed structure of the NSC and its major components. In addition, it describes the characteristics of the data models to identify the IETF Network Slice and its realization. Then the data models referred are mapped to the interfaces among components.

2. IETF Network Slice data models

At the time of provisioning and operating IETF Network Slices different views can be identified as necessary:

- Customer's view, mostly focused on the individual IETF Network Slice request process, reflecting the needs of each particular customer, including SLOs and other characteristics of the slice relevant for it. This view is technology agnostics and describes the characteristics of the IETF Network Slice from a customer's point of view. It can include the slice topology, performance parameters, endpoints of the slice, traffic characteristics of the slice, and the KPIs to monitor the slice.

- Provider's view, mostly focused on the provisioning and operation of the IETF Network Slices in the transport network, considering how a particular IETF Network Slice interplays with other IETF Network Slices maintained by the provider on a shared infrastructure. In other words, operator's view shows how an IETF Network Slice is realized in operator's network along with all the resources used during the its realization.

Both views are complementary, each of them specialized for a given purpose. In consequence, it should be consistency between both in order to ensure alignment.

Currently there are two different models proposed, on for each of the categories above. The model in [I-D.wd-teas-ietf-network-slice-nbi-yang] fits into the customer view, while the model defined in [I-D.liu-teas-transport-network-slice-yang] fits in to the provider view.

It should be noted that for the realization of a transport slice, the NSC interacts with one or more Network Controllers. In that case, the data models to be used are particular for each Network Controller (e.g., technology dependent), as well as the mapping function from its NBI to SBI and the details of this mapping function are both out of the scope of this document.

3. Structure of the IETF Network Slice Controller (NSC)

The NSC should work with both data models. The NSC takes first the customer's view by analyzing the needs of the customer, processing such requests taking into account the overall view of the network and the IETF Network Slices already instantiated, normalizing its instantiation across different technologies, and finally generates the provider view.

Once the new request is processed and declared as feasible, the NSC triggers its realization by interacting with the Network Controllers and communicates back to the higher level controller to start the billing cycle.

In order to accommodate these procedures, the internal structure of the NSC can be divided into:

- 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 2 illustrates the components described and the associated models, as follows

- (a) -> customer's view, e.g. [I-D.wd-teas-ietf-network-slice-nbi-yang].

- (b) -> provider's view, e.g. [I-D.liu-teas-transport-network-slice-yang].

- (c) -> models per network controller, out of scope of this document

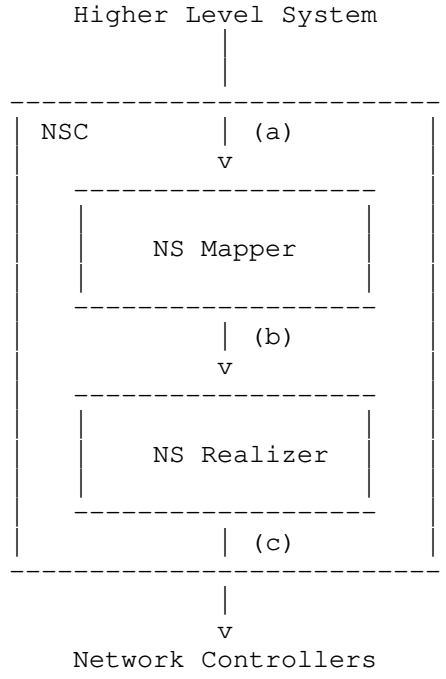


Figure 2: IETF Network Slice Controller structure and asspociated data models

TODO item #1 - Breakdown "NS mapper" and "NS Realizer" to their logical components.

TODO item #2- Add complementarity of the models for satisfying Type 1 and Type 2 Services as per [RFC8453]. Discussion: equivalent to the Virtual Network (VN) described in [RFC8453], there are two views of an IETF network slices as well:

- The IETF network slice can be abstracted as a set of edge-to-edge links (Type 1).
- The IETF network slice can be abstracted as a topology of virtual nodes and virtual links (Type 2) which represent the partitioning of underlay network resources for use by network slice connectivity.

The use cases of these two types of networks are further described by [RFC8453]. [I-D.wd-teas-ietf-network-slice-nbi-yang] models the Type 1 service, while [I-D.liu-teas-transport-network-slice-yang] models the Type 2 service. When a customer intends to request a Type 2 service, [I-D.liu-teas-transport-network-slice-yang] can also be used

at the point (a) in Figure 2. As an example, when ACTN is used to realize an IETF network slice, model mappings are described in more details in [I-D.ietf-teas-actn-yang].

4. Security Considerations

To be done.

5. IANA Considerations

This draft does not include any IANA considerations

6. References

[I-D.ietf-teas-actn-yang]
Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B., Dios, O., Shin, J., and S. Belotti, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-yang-06 (work in progress), August 2020.

[I-D.liu-teas-transport-network-slice-yang]
Liu, X., Tantsura, J., Bryskin, I., Contreras, L., WU, Q., Belotti, S., and R. Rokui, "Transport Network Slice YANG Data Model", draft-liu-teas-transport-network-slice-yang-01 (work in progress), July 2020.

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-00 (work in progress), October 2020.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

[I-D.wd-teas-ietf-network-slice-nbi-yang]
Bo, W., Dhody, D., Han, L., and R. Rokui, "A Yang Data Model for IETF Network Slice NBI", draft-wd-teas-ietf-network-slice-nbi-yang-00 (work in progress), October 2020.

[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>.

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
URI: http://lmcontreras.com/

Reza Rokui
Nokia
Canada

Email: reza.rokui@nokia.com

Jeff Tantsura
Apstra
USA

Email: jefftant.ietf@gmail.com

Bo Wu
Huawei Technologies
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: lana.wubo@huawei.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Sergio Belloti
Nokia

Email: sergio.belotti@nokia.com

TEAS Working Group
Internet-Draft
Intended status: Informational
Expires: May 3, 2021

LM. Contreras
Telefonica
S. Homma
NTT
J. Ordonez-Lucena
Telefonica

October 30, 2020

IETF Network Slice use cases and attributes for Northbound Interface of controller
draft-contreras-teas-slice-nbi-03

Abstract

The transport network is an essential component in the end-to-end delivery of services and, consequently, with the advent of network slicing it is necessary to understand what could be the way in which the transport network is consumed as a slice. This document analyses the needs of potential IETF network slice customers (i.e., use cases) in order to identify the functionality required on the North Bound Interface (NBI) of a IETF network slice controller for satisfying such IETF network slice 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 May 3, 2021.

Table of Contents

1. Introduction

Editor's Note: 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.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework] is primarily used here.

Editor's Note: 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.

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.nsdt-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

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].

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 Fig. 1.

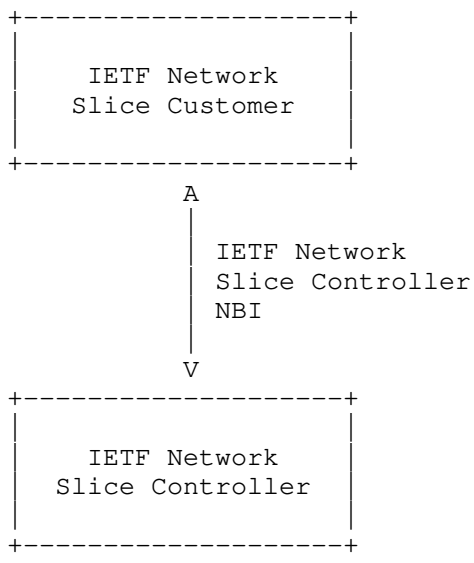


Figure 1: IETF network slice NBI concept

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 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 mIoT, 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. 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:

- Attribute definition, which provides a formal definition of what the attribute represents.
- 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.
- 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.

4.1.2. 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:

- Directly impactive 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.

- Indirectly impactive attributes, which are thise 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.

- 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.

4.1.2.1. Attributes with direct impact on the IETF network slice definition

The following attributes impose requirements in the IETF network slice

- 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 (i.e., tunneling mechanisms)

4.1.2.2. 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.

- Area of service (i.e., the area where terminals can access a particular network slice)
- Delay tolerance (i.e., if the service can be delivered when the system has sufficient resources)
- Downlink (maximum) throughput per UE
- Network functions owned by Network Slice Customer
- Maximum number of (concurrent) PDU sessions
- Performance prediction (i.e., capability to predict the network and service status)
- Root cause investigation
- Session and Service Continuity support
- Simultaneous use of the network slice
- Supported device velocity
- UE density
- Uplink (maximum) throughput per UE
- User management openness (i.e., capability to manage users' network services and corresponding requirements)
- Latency from (last) UPF to Application Server

4.1.2.3. Attributes with no impact on the IETF network slice definition

The following attributes do not impact the IETF network slice.

- Location based message delivery (not related to the geographical spread of the network slice itself but with the localized distribution of information)
- MMTel support, i.e. support of and Multimedia Telephony Service (MMTel)as well as IP Multimedia Subsystem (IMS) support.

- NB-IoT Support, i.e., support of NB-IoT in the RAN in the network slice.
- Maximum number of (simultaneous) UEs
- Positioning support
- Radio spectrum
- Synchronicity (among devices)
- V2X communication mode
- Network Slice Specific Authentication and Authorization (NSSAA)

4.1.3. 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:

- Slice instance allocation: this procedure permits to create a new slice instance (or reuse an existing one).
- Slice instance de-allocation: this procedure decommissions a previously instantiated slice.
- Slice instance modification: this procedure permits the change in the characteristics of an existing slice instance.
- Get slice instance status: this procedure helps to retrieve run-time information on the status of a deployed slice instance.
- 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:

- Incoming and outgoing bandwidth: bandwidth required for the connectivity services (in Mbps).
- Qos metrics: set of metrics (e.g., cost, latency and delay variation) applicable to a specific connectivity service
- Directionality: indication if the traffic is unidirectional or bidirectional.
- MTU: value of the largest PDU to be transmitted in the connectivity service.
- Protection scheme: indication of the kind of protection to be performed (e.g., 1;1, 1+1, etc.)
- 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:

- 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.

- 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.

- Fault Management: this interface serves for the provision of alarms related to the MSCSs.

- Performance Management: this interface assists on the retrieval of performance information (measurement results collection and notifications) related to MSCSs.

4.3. RAN 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:

- Maximum and Guaranteed Bit Rate (MBR and GBR respectively).
- Bounded latency (e.g., for user plane, control plane, etc)
- Packet loss rate.
- IP addressing (consistent among the operators sharing the infrastructure).
- L2/L3 reachability.
- Recovery time (on the event of failures).
- Secure connection (e.g., encryption support).

4.3.2. Provisioning procedures

The expected provisioning procedures are:

- Connection provisioning between site and interconnection point. Those connections could evolve in time in terms of capacity depending on the capacity grow of each particular site.
- Collection of management data, including performance measurements, fault alarms and trace data.

4.4. 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., Lopez, D., Contreras, L., Ordonez-Lucena, J., Martinez-Julia, P., Qiang, L., Rokui, R., Ciavaglia, L., and X. Foy, "Network Slice Provision Models", draft-homma-slice-provision-models-02 (work in progress), November 2019.

[I-D.nsdt-teas-ietf-network-slice-definition] Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-00 (work in progress), October 2020.

[I-D.nsdt-teas-ns-framework] Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

[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.

[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
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

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 5, 2021

Y. Lee, Ed.
Samsung Electronics
D. Dhody, Ed.
S. Karunanithi
Huawei Technologies
R. Vilalta
CTTC
D. King
Lancaster University
D. Ceccarelli
Ericsson

November 1, 2020

YANG models for VN/TE Performance Monitoring Telemetry and Scaling Intent Autonomics

draft-ietf-teas-actn-pm-telemetry-autonomics-04

Abstract

This document provides YANG data models that describe performance monitoring telemetry and scaling intent mechanism for TE-tunnels and Virtual Networks (VN).

The models presented in this draft allow customers to subscribe to and monitor their key performance data of their interest on the level of TE-tunnel or VN. The models also provide customers with the ability to program autonomic scaling intent mechanism on the level of TE-tunnel as well as VN.

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 May 5, 2021.

Table of Contents

1. Introduction

The YANG [RFC7950] model discussed in [I-D.ietf-teas-actn-vn-yang] is used to operate customer-driven Virtual Networks (VNs) during the VN instantiation, VN computation, and its life-cycle service management and operations. YANG model discussed in [I-D.ietf-teas-yang-te] is

used to operate TE-tunnels during the tunnel instantiation, and its life-cycle management and operations.

The models presented in this draft allow the applications hosted by the customers to subscribe to and monitor their key performance data of their interest on the level of VN [I-D.ietf-teas-actn-vn-yang] or TE-tunnel [I-D.ietf-teas-yang-te]. The key characteristic of the models presented in this document is a top-down programmability that allows the applications hosted by the customers to subscribe to and monitor key performance data of their interest and autonomic scaling intent mechanism on the level of VN as well as TE-tunnel.

According to the classification of [RFC8309], the YANG data models presented in this document can be classified as customer service models, which is mapped to CMI (Customer Network Controller (CNC)-Multi-Domain Service Coordinator (MSDC) interface) of ACTN [RFC8453].

[RFC8233] describes key network performance data to be considered for end-to-end path computation in TE networks. Key performance indicator (KPI) is a term that describes critical performance data that may affect VN/TE-tunnel service. The services provided can be optimized to meet the requirements (such as traffic patterns, quality, and reliability) of the applications hosted by the customers.

This document provides YANG data models generically applicable to any VN/TE-Tunnel service clients to provide an ability to program their customized performance monitoring subscription and publication data models and automatic scaling in/out intent data models. These models can be utilized by a client network controller to initiate these capability to a transport network controller communicating with the client controller via a NETCONF [RFC8341] or a RESTCONF [RFC8040] interface.

The term performance monitoring being used in this document is different from the term that has been used in transport networks for many years. Performance monitoring in this document refers to subscription and publication of streaming telemetry data. Subscription is initiated by the client (e.g., CNC) while publication is provided by the network (e.g., MDSC/PNC) based on the client's subscription. As the scope of performance monitoring in this document is telemetry data on the level of client's VN or TE- tunnel, the entity interfacing the client (e.g., MDSC) has to provide VN or TE-tunnel level information. This would require controller capability to derive VN or TE-tunnel level performance data based on lower-level data collected via PM counters in the Network Elements (NE). How the controller entity derives such customized level data (i.e., VN or TE-tunnel level) is out of the scope of this document.

The data model includes configuration and state data according to the new Network Management Datastore Architecture [RFC8342].

1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

Key Performance Data: This refers to a set of data the customer is interested in monitoring for their instantiated VNs or TE-tunnels. Key performance data and key performance indicators are inter-exchangeable in this draft.

Scaling: This refers to the network ability to re-shape its own resources. Scale out refers to improve network performance by increasing the allocated resources, while scale in refers to decrease the allocated resources, typically because the existing resources are unnecessary.

Scaling Intent: To declare scaling conditions, scaling intent is used. Specifically, scaling intent refers to the intent expressed by the client that allows the client to program/configure conditions of their key performance data either for scaling out or scaling in. Various conditions can be set for scaling intent on either VN or TE-tunnel level.

Network Autonomics: This refers to the network automation capability that allows client to initiate scaling intent mechanisms and provides the client with the status of the adjusted network resources based on the client’s scaling intent in an automated fashion.

1.1.1. Requirements Language

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. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

1.3. 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
inet	ietf-inet-types	[RFC6991]
te	ietf-te	[I-D.ietf-teas-yang-te]
te-types	ietf-te-types	[RFC8776]
te-tel	ietf-te-kpi-telemetry	[RFCXXXX]
vn	ietf-vn	[I-D.ietf-teas-actn-vn-yang]
vn-tel	ietf-vn-kpi-telemetry	[RFCXXXX]

Table 1: Prefixes and corresponding YANG modules

Note: The RFC Editor will replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

Further, the following additional documents are refrenced in the model defined in this document -

- [RFC7471] - OSPF Traffic Engineering (TE) Metric Extensions.
- [RFC8570] - IS-IS Traffic Engineering (TE) Metric Extensions.
- [RFC7823] - Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions.

2. Use-Cases

[I-D.xu-actn-perf-dynamic-service-control] describes use-cases relevant to this draft. It introduces the dynamic creation, modification and optimization of services based on the performance monitoring. Figure 1 shows a high-level workflows for dynamic service control based on traffic monitoring.

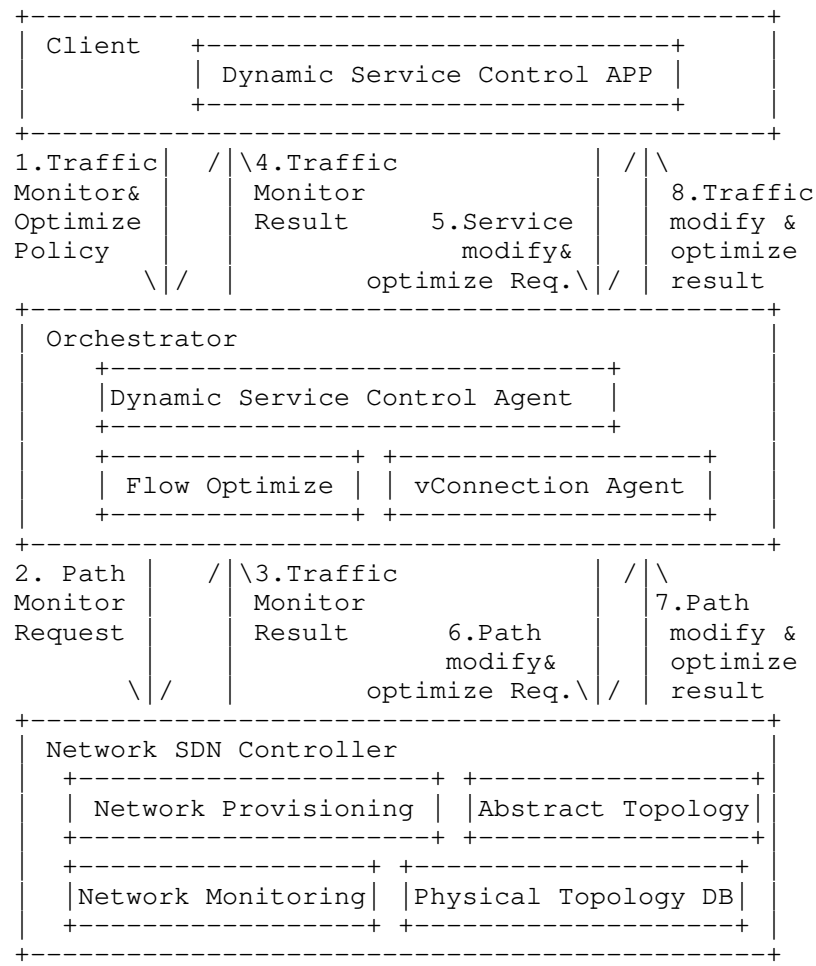


Figure 1: Workflows for dynamic service control based on traffic monitoring

Some of the key points from [I-D.xu-actn-perf-dynamic-service-control] are as follows:

- Network traffic monitoring is important to facilitate automatic discovery of the imbalance of network traffic, and initiate the network optimization, thus helping the network operator or the virtual network service provider to use the network more efficiently and save the Capital Expense (CAPEX) and the Operating Expense (OPEX).

- Customer services have various Service Level Agreement (SLA) requirements, such as service availability, latency, latency jitter, packet loss rate, Bit Error Rate (BER), etc. The transport network can satisfy service availability and BER requirements by providing different protection and restoration mechanisms. However, for other performance parameters, there are no such mechanisms. In order to provide high quality services according to customer SLA, one possible solution is to measure the SLA related performance parameters, and dynamically provision and optimize services based on the performance monitoring results.

- Performance monitoring in a large scale network could generate a huge amount of performance information. Therefore, the appropriate way to deliver the information in the client and network interfaces should be carefully considered.

3. Design of the Data Models

The YANG models developed in this document describe two models:

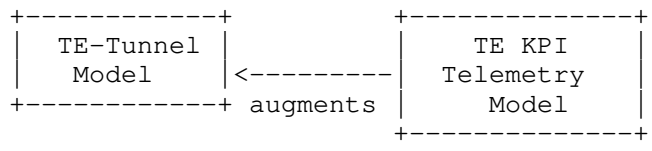
(i) TE KPI Telemetry Model which provides the TE-Tunnel level of performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer. (See Section 3.1 & Section 7.1 for details).

(ii) VN KPI Telemetry Model which provides the VN level of the aggregated performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer (See Section 3.2 & Section 7.2 for details).

3.1. TE KPI Telemetry Model

This module describes performance telemetry for TE-tunnel model. The telemetry data is augmented to tunnel state. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the TE-tunnel level. Various conditions can be set for auto-scaling based on the telemetry data (See Section 5 for details)

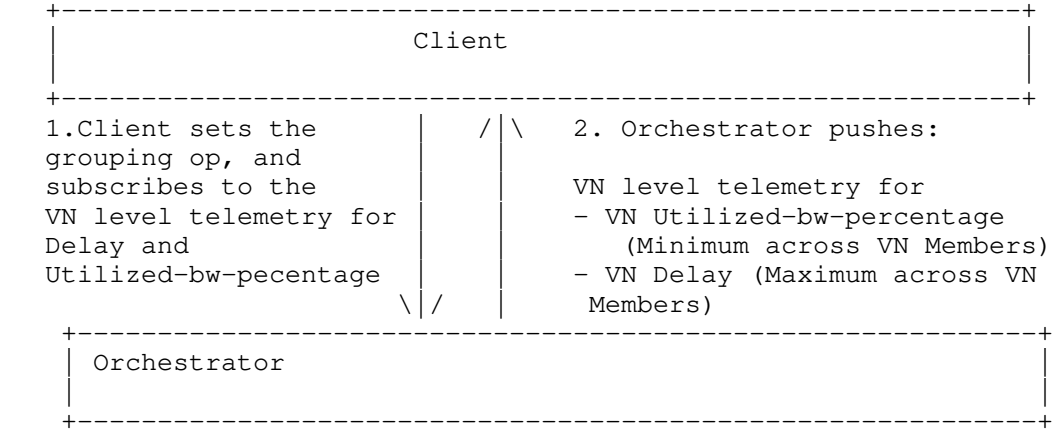
The TE KPI Telemetry Model augments the TE-Tunnel Model to enhance TE performance monitoring capability. This monitoring capability will facilitate proactive re-optimization and reconfiguration of TEs based on the performance monitoring data collected via the TE KPI Telemetry YANG model.



3.2. VN KPI Telemetry Model

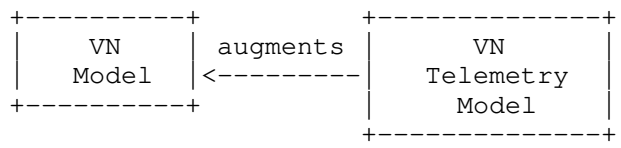
This module describes performance telemetry for VN model. The telemetry data is augmented both at the VN Level as well as individual VN member level. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the VN level. Scale in/out criteria might be used for network autonomics in order the controller to react to a certain set of variations in monitored parameters (See Section 4 for illustrations).

Moreover, this module also provides mechanism to define aggregated telemetry parameters as a grouping of underlying VN level telemetry parameters. Grouping operation (such as maximum, mean) could be set at the time of configuration. For example, if maximum grouping operation is used for delay at the VN level, the VN telemetry data is reported as the maximum {delay_vn_member_1, delay_vn_member_2,.. delay_vn_member_N}. Thus, this telemetry abstraction mechanism allows the grouping of a certain common set of telemetry values under a grouping operation. This can be done at the VN-member level to suggest how the E2E telemetry be inferred from the per domain tunnel created and monitored by PNCs. One proposed example is the following:



The VN Telemetry Model augments the basic VN model to enhance VN monitoring capability. This monitoring capability will facilitate proactive re-optimization and reconfiguration of VNs based on the

performance monitoring data collected via the VN Telemetry YANG model.



4. Autonomic Scaling Intent Mechanism

Scaling intent configuration mechanism allows the client to configure automatic scale-in and scale-out mechanisms on both the TE-tunnel and the VN level. Various conditions can be set for auto- scaling based on the PM telemetry data.

There are a number of parameters involved in the mechanism:

- scale-out-intent or scale-in-intent: whether to scale-out or scale-in.
- performance-type: performance metric type (e.g., one-way-delay, one-way-delay-min, one-way-delay-max, two-way-delay, two-way-delay-min, two-way-delay-max, utilized bandwidth, etc.)
- threshold-value: the threshold value for a certain performance-type that triggers scale-in or scale-out.
- scaling-operation-type: in case where scaling condition can be set with one or more performance types, then scaling-operation-type (AND, OR, MIN, MAX, etc.) is applied to these selected performance types and its threshold values.
- Threshold-time: the duration for which the criteria MUST hold true.
- Cooldown-time: the duration after a scaling action has been triggered, for which there will be no further operation.

The following tree is a part of ietf-te-kpi-telemetry tree whose model is presented in full detail in Sections 6 & 7.

```
module: ietf-te-kpi-telemetry
  augment /te:te/te:tunnels/te:tunnel:
    +--rw te-scaling-intent
    |  +--rw scale-in-intent
    |  |  +--rw threshold-time?      uint32
    |  |  +--rw cooldown-time?       uint32
    |  |  +--rw scaling-condition* [performance-type]
    |  |     +--rw performance-type           identityref
    |  |     +--rw threshold-value?           string
    |  |     +--rw scale-in-operation-type?
    |  |             scaling-criteria-operation
    |  +--rw scale-out-intent
    |     +--rw threshold-time?      uint32
    |     +--rw cooldown-time?       uint32
    |     +--rw scaling-condition* [performance-type]
    |        +--rw performance-type            identityref
    |        +--rw threshold-value?            string
    |        +--rw scale-out-operation-type?
    |                scaling-criteria-operation
```

Let say the client wants to set the scaling out operation based on two performance-types (e.g., two-way-delay and utilized-bandwidth for a te-tunnel), it can be done as follows:

- Set Threshold-time: x (sec) (duration for which the criteria must hold true)

- Set Cooldown-time: y (sec) (the duration after a scaling action has been triggered, for which there will be no further operation)

- Set AND for the scale-out-operation-type

In the scaling condition's list, the following two components can be set:

List 1: Scaling Condition for Two-way-delay

- performance type: Two-way-delay

- threshold-value: z milli-seconds

List 2: Scaling Condition for Utilized bandwidth

- performance type: Utilized bandwidth

- threshold-value: w megabytes

5. Notification

This model does not define specific notifications. To enable notifications, the mechanism defined in [RFC8641] and [RFC8640] can be used. This mechanism currently allows the user to:

- Subscribe to notifications on a per client basis.
- Specify subtree filters or xpath filters so that only interested contents will be sent.
- Specify either periodic or on-demand notifications.

5.1. YANG Push Subscription Examples

[RFC8641] allows subscriber applications to request a continuous, customized stream of updates from a YANG datastore.

Below example shows the way for a client to subscribe to the telemetry information for a particular tunnel (Tunnel1). The telemetry parameter that the client is interested in is one-way-delay.

```
<netconf:rpc netconf:message-id="101"
    xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
    <establish-subscription
       xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
       <filter netconf:type="subtree">
          <te xmlns="urn:ietf:params:xml:ns:yang:ietf-te">
             <tunnels>
                <tunnel>
                  <name>Tunnel1</name>
                  <identifier/>
                  <state>
                    <te-telemetry xmlns="urn:ietf:params:xml:ns:yang:
                                         ietf-te-kpi-telemetry">
                        <one-way-delay/>
                     </te-telemetry>
                  </state>
                </tunnel>
             </tunnels>
          </te>
       </filter>
       <period>500</period>
       <encoding>encode-xml</encoding>
    </establish-subscription>
 </netconf:rpc>
```

This example shows the way for a client to subscribe to the telemetry information for all VNs. The telemetry parameter that the client is interested in is one-way-delay and one-way-utilized- bandwidth.

```
<netconf:rpc netconf:message-id="101"
    xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
    <establish-subscription
       xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
       <filter netconf:type="subtree">
          <vn-state xmlns="urn:ietf:params:xml:ns:yang:ietf-vn">
             <vn>
                <vn-list>
                  <vn-id/>
                  <vn-name/>
                  <vn-telemetry xmlns="urn:ietf:params:xml:ns:yang:
                                       ietf-vn-kpi-telemetry">
                      <one-way-delay/>
                      <one-way-utilized-bandwidth/>
                  </vn-telemetry >
                </vn-list>
              </vn>
          </vn-state>
       </filter>
       <period>500</period>
    </establish-subscription>
 </netconf:rpc>
```

6. YANG Data Tree

```
module: ietf-te-kpi-telemetry
  augment /te:te/te:tunnels/te:tunnel:
    +--rw te-scaling-intent
    |  +--rw scale-in-intent
    |  |  +--rw threshold-time?      uint32
    |  |  +--rw cooldown-time?       uint32
    |  |  +--rw scaling-condition* [performance-type]
    |  |     +--rw performance-type           identityref
    |  |     +--rw threshold-value?           string
    |  |     +--rw scale-in-operation-type?
    |  |             scaling-criteria-operation
    |  +--rw scale-out-intent
    |     +--rw threshold-time?      uint32
    |     +--rw cooldown-time?       uint32
    |     +--rw scaling-condition* [performance-type]
    |        +--rw performance-type            identityref
```

```
      |        +--rw threshold-value?            string
      |        +--rw scale-out-operation-type?
      |                scaling-criteria-operation
      +--ro te-telemetry
         +--ro id?                              telemetry-id
         +--ro performance-metrics-one-way
         |  +--ro one-way-delay?                             uint32
         |  +--ro one-way-delay-normality?
         |  |       te-types:performance-metrics-normality
         |  +--ro one-way-residual-bandwidth?
         |  |       rt-types:bandwidth-ieee-float32
         |  +--ro one-way-residual-bandwidth-normality?
         |  |       te-types:performance-metrics-normality
         |  +--ro one-way-available-bandwidth?
         |  |       rt-types:bandwidth-ieee-float32
         |  +--ro one-way-available-bandwidth-normality?
         |  |       te-types:performance-metrics-normality
         |  +--ro one-way-utilized-bandwidth?
         |  |       rt-types:bandwidth-ieee-float32
         |  +--ro one-way-utilized-bandwidth-normality?
         |          te-types:performance-metrics-normality
         +--ro performance-metrics-two-way
            +--ro two-way-delay?             uint32
            +--ro two-way-delay-normality?
                    te-types:performance-metrics-normality
```

```
module: ietf-vn-kpi-telemetry
  augment /vn:vn/vn:vn-list:
    +--rw vn-scaling-intent
    |  +--rw scale-in-intent
    |  |  +--rw threshold-time?      uint32
    |  |  +--rw cooldown-time?       uint32
    |  |  +--rw scaling-condition* [performance-type]
    |  |     +--rw performance-type           identityref
    |  |     +--rw threshold-value?           string
    |  |     +--rw scale-in-operation-type?
    |  |             scaling-criteria-operation
    |  +--rw scale-out-intent
    |     +--rw threshold-time?      uint32
    |     +--rw cooldown-time?       uint32
    |     +--rw scaling-condition* [performance-type]
    |        +--rw performance-type            identityref
    |        +--rw threshold-value?            string
    |        +--rw scale-out-operation-type?
    |                scaling-criteria-operation
```

```
       +--ro vn-telemetry
          +--ro performance-metrics-one-way
          |  +--ro one-way-delay?                            uint32
          |  +--ro one-way-delay-normality?
          |  |       te-types:performance-metrics-normality
          |  +--ro one-way-residual-bandwidth?
          |  |       rt-types:bandwidth-ieee-float32
          |  +--ro one-way-residual-bandwidth-normality?
          |  |       te-types:performance-metrics-normality
          |  +--ro one-way-available-bandwidth?
          |  |       rt-types:bandwidth-ieee-float32
          |  +--ro one-way-available-bandwidth-normality?
          |  |       te-types:performance-metrics-normality
          |  +--ro one-way-utilized-bandwidth?
          |  |       rt-types:bandwidth-ieee-float32
          |  +--ro one-way-utilized-bandwidth-normality?
          |          te-types:performance-metrics-normality
          +--ro performance-metrics-two-way
          |  +--ro two-way-delay?             uint32
          |  +--ro two-way-delay-normality?
          |          te-types:performance-metrics-normality
          +--ro grouping-operation?            grouping-operation
     augment /vn:vn/vn:vn-list/vn:vn-member-list:
       +--ro vn-member-telemetry
          +--ro performance-metrics-one-way
          |  +--ro one-way-delay?                            uint32
          |  +--ro one-way-delay-normality?
          |  |       te-types:performance-metrics-normality
          |  +--ro one-way-residual-bandwidth?
          |  |       rt-types:bandwidth-ieee-float32
          |  +--ro one-way-residual-bandwidth-normality?
          |  |       te-types:performance-metrics-normality
          |  +--ro one-way-available-bandwidth?
          |  |       rt-types:bandwidth-ieee-float32
          |  +--ro one-way-available-bandwidth-normality?
          |  |       te-types:performance-metrics-normality
          |  +--ro one-way-utilized-bandwidth?
          |  |       rt-types:bandwidth-ieee-float32
          |  +--ro one-way-utilized-bandwidth-normality?
          |          te-types:performance-metrics-normality
          +--ro performance-metrics-two-way
          |  +--ro two-way-delay?             uint32
          |  +--ro two-way-delay-normality?
          |          te-types:performance-metrics-normality
          +--ro te-grouped-params*
          |       -> /te:te/tunnels/tunnel/te-kpi:te-telemetry/id
          +--ro grouping-operation?            grouping-operation
```

7. YANG Data Model

7.1. ietf-te-kpi-telemetry model

The YANG code is as follows:

```
 <CODE BEGINS> file "ietf-te-kpi-telemetry@2020-11-02.yang"
module ietf-te-kpi-telemetry {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry";
  prefix te-tel;

  /* Import inet-types */

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }

  /* Import TE */

  import ietf-te {
    prefix te;
    reference
      "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
       Engineering Tunnels and Interfaces";
  }

  /* Import TE Common 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:  Young Lee <younglee.tx@gmail.com>
              Dhruv Dhody <dhruv.ietf@gmail.com>";
  description
    "This module describes YANG data model for performance
     monitoring telemetry for te tunnels.
```

```
      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
      (https://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX; see the
      RFC itself for full legal notices.

      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 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.";

   /* Note: The RFC Editor will replace XXXX with the number
      assigned to the RFC once draft-ietf-teas-pm-telemetry-
      autonomics becomes an RFC.*/

   revision 2020-11-02 {
     description
       "Initial revision.";
     reference
       "RFC XXXX: YANG models for VN/TE Performance Monitoring
        Telemetry and Scaling Intent Autonomics";
   }

   identity telemetry-param-type {
     description
       "Base identity for telemetry param types";
   }

   identity one-way-delay {
     base telemetry-param-type;
     description
       "To specify average Delay in one (forward) direction.

        At the VN level, it is the max delay of the VN-members.";
     reference
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: Performance-Based Path Selection for Explicitly
        Routed Label Switched Paths (LSPs) Using TE Metric
        Extensions";
```

```
   }

   identity two-way-delay {
     base telemetry-param-type;
     description
       "To specify average Delay in both (forward and reverse)
        directions.

        At the VN level, it is the max delay of the VN-members.";
     reference
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: Performance-Based Path Selection for Explicitly
        Routed Label Switched Paths (LSPs) Using TE Metric
        Extensions";
   }

   identity one-way-delay-variation {
     base telemetry-param-type;
     description
       "To specify average Delay Variation in one (forward) direction.

        At the VN level, it is the max delay variation of the
        VN-members.";
     reference
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: Performance-Based Path Selection for Explicitly
        Routed Label Switched Paths (LSPs) Using TE Metric
        Extensions";
   }

   identity two-way-delay-variation {
     base telemetry-param-type;
     description
       "To specify average Delay Variation in both (forward and reverse)
        directions.

        At the VN level, it is the max delay variation of the
        VN-members.";
     reference
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: Performance-Based Path Selection for Explicitly
        Routed Label Switched Paths (LSPs) Using TE Metric
        Extensions";
   }
```

```
   identity utilized-bandwidth {
     base telemetry-param-type;
     description
       "To specify utilized bandwidth over the specified source
        and destination.";
     reference
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: Performance-Based Path Selection for Explicitly
        Routed Label Switched Paths (LSPs) Using TE Metric
        Extensions";
   }

   identity utilized-percentage {
     base telemetry-param-type;
     description
       "To specify utilization percentage of the entity
        (e.g., tunnel, link, etc.)";
   }

   /* Typedef */

   typedef telemetry-id {
     type inet:uri;
     description
       "Identifier for telemetry data.  The precise
        structure of the telemetry-id will be up to the
        implementation. The identifier SHOULD be chosen
        such that the same telemetry data will always be
        identified through the same identifier, even if
        the data model is instantiated in separate
        datastores.";
   }

   typedef scaling-criteria-operation {
     type enumeration {
       enum AND {
         description
           "AND operation";
       }
       enum OR {
         description
           "OR operation";
       }
     }
     description
       "Operations to analize list of scaling criterias";
   }
```

```
   grouping scaling-duration {
     description
       "Base scaling criteria durations";
     leaf threshold-time {
       type uint32;
       units "seconds";
       description
         "The duration for which the criteria must hold true";
     }
     leaf cooldown-time {
       type uint32;
       units "seconds";
       description
         "The duration after a scaling-in/scaling-out action has been
          triggered, for which there will be no further operation";
     }
   }

   grouping scaling-criteria {
     description
       "Grouping for scaling criteria";
     leaf performance-type {
       type identityref {
         base telemetry-param-type;
       }
       description
         "Reference to the tunnel level telemetry type";
     }
     leaf threshold-value {
       type string;
       description
         "Scaling threshold for the telemetry parameter type";
     }
   }

   grouping scaling-in-intent {
     description
       "Basic scaling in intent";
     uses scaling-duration;
     list scaling-condition {
       key "performance-type";
       description
         "Scaling conditions";
       uses scaling-criteria;
       leaf scale-in-operation-type {
         type scaling-criteria-operation;
         default "AND";
         description
```

```
            "Operation to be applied to check between scaling criterias
             to check if the scale in threshold condition has been met.
             Defaults to AND";
        }
      }
    }

    grouping scaling-out-intent {
      description
        "Basic scaling out intent";
      uses scaling-duration;
      list scaling-condition {
        key "performance-type";
        description
          "Scaling conditions";
        uses scaling-criteria;
        leaf scale-out-operation-type {
          type scaling-criteria-operation;
          default "OR";
          description
            "Operation to be applied to check between scaling criterias
             to check if the scale out threshold condition has been met.
             Defauls to OR";
        }
      }
    }

    augment "/te:te/te:tunnels/te:tunnel" {
      description
        "Augmentation parameters for config scaling-criteria TE
         tunnel topologies. Scale in/out criteria might be used
         for network autonomics in order the controller to react
         to a certain set of monitored params.";
      container te-scaling-intent {
        description
          "The scaling intent";
        container scale-in-intent {
          description
            "scale-in";
          uses scaling-in-intent;
        }
        container scale-out-intent {
          description
            "scale-out";
          uses scaling-out-intent;
        }
      }
      container te-telemetry {
```

```
       config false;
       description
         "Telemetry Data";
       leaf id {
         type telemetry-id;
         description
           "ID of telemetry data used for easy reference";
       }
       uses te-types:performance-metrics-attributes;
     }
   }
 }

 <CODE ENDS>
```

7.2. ietf-vn-kpi-telemetry model

The YANG code is as follows:

```
 <CODE BEGINS> file "ietf-vn-kpi-telemetry@2020-11-02.yang"
 module ietf-vn-kpi-telemetry {
   yang-version 1.1;
   namespace "urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry";
   prefix vn-kpi;

   /* Import VN */

   import ietf-vn {
     prefix vn;
     reference
       "I-D.ietf-teas-actn-vn-yang: A YANG Data Model for VN
        Operation";
   }

   /* Import TE */

   import ietf-te {
     prefix te;
     reference
       "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
        Engineering Tunnels and Interfaces";
   }

   /* Import TE Common types */

   import ietf-te-types {
     prefix te-types;
     reference
```

```
        "RFC 8776: Common YANG Data Types for Traffic Engineering";
    }

    /* Import TE KPI */

    import ietf-te-kpi-telemetry {
      prefix te-kpi;
      reference
        "RFC XXXX: YANG models for VN/TE Performance Monitoring
         Telemetry and Scaling Intent Autonomics";
    }

    /* Note: The RFC Editor will replace XXXX with the number
       assigned to this draft.*/

    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:  Young Lee <younglee.tx@gmail.com>
                Dhruv Dhody <dhruv.ietf@gmail.com>";
    description
      "This module describes YANG data models for performance
       monitoring telemetry for Virtual Network (VN).

       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
       (https://trustee.ietf.org/license-info).

       This version of this YANG module is part of RFC XXXX; see the
       RFC itself for full legal notices.

       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 (RFC 2119) (RFC 8174) when, and only when,
       they appear in all capitals, as shown here.";

    /* Note: The RFC Editor will replace XXXX with the number
       assigned to the RFC once draft-lee-teas-pm-telemetry-
```

```
       autonomics becomes an RFC.*/

    revision 2020-11-02 {
      description
        "Initial revision.";
      reference
        "RFC XXXX: YANG models for VN/TE Performance Monitoring
         Telemetry and Scaling Intent Autonomics";
    }

    typedef grouping-operation {
      type enumeration {
        enum MINIMUM {
          description
            "Select the minimum param";
        }
        enum MAXIMUM {
          description
            "Select the maximum param";
        }
        enum MEAN {
          description
            "Select the MEAN of the params";
        }
        enum STD_DEV {
          description
            "Select the standard deviation of the monitored params";
        }
        enum AND {
          description
            "Select the AND of the params";
        }
        enum OR {
          description
            "Select the OR of the params";
        }
      }
      description
        "Operations to analize list of monitored params";
    }

    grouping vn-telemetry-param {
      description
        "augment of te-kpi:telemetry-param for VN specific params";
      leaf-list te-grouped-params {
        type leafref {
          path
            "/te:te/te:tunnels/te:tunnel/te-kpi:te-telemetry/te-kpi:id";
```

```
        }
        description
          "Allows the definition of a vn-telemetry param
           as a grouping of underlying TE params";
      }
      leaf grouping-operation {
        type grouping-operation;
        description
          "describes the operation to apply to
           te-grouped-params";
      }
    }

    augment "/vn:vn/vn:vn-list" {
      description
        "Augmentation parameters for state TE VN topologies.";
      container vn-scaling-intent {
        description
          "scaling intent";
        container scale-in-intent {
          description
            "VN scale-in";
          uses te-kpi:scaling-in-intent;
        }
        container scale-out-intent {
          description
            "VN scale-out";
          uses te-kpi:scaling-out-intent;
        }
      }
      container vn-telemetry {
        config false;
        description
          "VN telemetry params";
        uses te-types:performance-metrics-attributes;
        leaf grouping-operation {
          type grouping-operation;
          description
            "describes the operation to apply to the VN-members";
        }
      }
    }

    augment "/vn:vn/vn:vn-list/vn:vn-member-list" {
      description
        "Augmentation parameters for state TE vn member topologies.";
      container vn-member-telemetry {
        config false;
```

```
        description
          "VN member telemetry params";
        uses te-types:performance-metrics-attributes;
        uses vn-telemetry-param;
      }
    }
  }
```

<CODE ENDS>

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 NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF protocol operations and content. The NETCONF Protocol over Secure Shell (SSH) [RFC6242] describes a method for invoking and running NETCONF within a Secure Shell (SSH) session as an SSH subsystem. 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.

A number of configuration data nodes defined in this document are writable/deletable (i.e., "config true"). These data nodes may be considered sensitive or vulnerable in some network environments.

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:

- /te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-in-intent
- /te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-out-intent
- /vn:vn/vn:vn-list/vn-scaling-intent/scale-in-intent

- /vn:vn/vn:vn-list/vn-scaling-intent/scale-out-intent

9. IANA Considerations

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```
---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------

---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry
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 [RFC7950]:

```
---------------------------------------------------------------------
name:         ietf-te-kpi-telemetry
namespace:    urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry
prefix:       te-tel
reference:    RFC XXXX (TBD)
---------------------------------------------------------------------

---------------------------------------------------------------------
name:         ietf-vn-kpi-telemetry
namespace:    urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry
prefix:       vn-tel
reference:    RFC XXXX (TBD)
---------------------------------------------------------------------
```

10. Acknowledgements

We thank Rakesh Gandhi, Tarek Saad, Igor Bryskin and Kenichi Ogaki for useful discussions and their suggestions for this work.

11. References

11.1. Normative References

[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-09 (work in progress), July 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.

[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>.

[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>.

[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>.

[RFC8233] Dhody, D., Wu, Q., Manral, V., Ali, Z., and K. Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", RFC 8233, DOI 10.17487/RFC8233, September 2017, <https://www.rfc-editor.org/info/rfc8233>.

[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>.

11.2. Informative References

[I-D.xu-actn-perf-dynamic-service-control]
Xu, Y., Zhang, G., Cheng, W., and z. zhenghaomian@huawei.com, "Use Cases and Requirements of Dynamic Service Control based on Performance Monitoring in ACTN Architecture", draft-xu-actn-perf-dynamic-service-control-03 (work in progress), April 2015.

[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>.

[RFC7823] Atlas, A., Drake, J., Giacalone, S., and S. Previdi, "Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions", RFC 7823, DOI 10.17487/RFC7823, May 2016, <https://www.rfc-editor.org/info/rfc7823>.

[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>.

[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>.

Authors' Addresses

Young Lee (editor)
Samsung Electronics

Email: younglee.tx@gmail.com

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Satish Karunanithi
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: satish.karunanithi@gmail.com

Ricard Vilalta
CTTC
Centre Tecnologic de Telecomunicacions de Catalunya (CTTC/CERCA)
Barcelona
Spain

Email: ricard.vilalta@cttc.es

Daniel King
Lancaster University

Email: d.king@lancaster.ac.uk

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

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: May 2021

November 2, 2020

Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Packet Optical Integration (POI)

draft-ietf-teas-actn-poi-applicability-01

Status of this Memo

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, identifying the YANG data models being defined by the IETF to support this deployment architecture as well as specific scenarios relevant for Service Providers.

Existing IETF protocols and data models are identified for each multi-layer (packet over optical) scenario with particular focus on the MPI (Multi-Domain Service Coordinator to Provisioning Network Controllers Interface)in the ACTN architecture.

Table of Contents

1. Introduction

The full automation of the management and control of Service Providers transport networks (IP/MPLS, Optical and also Microwave) is key for achieving the new challenges coming now with 5G as well as with the increased demand in terms of business agility and mobility in a digital world. ACTN architecture, by abstracting the network complexity from Optical and IP/MPLS networks towards MDSC and then from MDSC towards OSS/BSS or Orchestration layer through the use of standard interfaces and data models, is 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 possibly 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 of each other. There are technical differences between the technologies (e.g., routers vs. optical switches) and the corresponding network engineering and planning methods (e.g., inter-domain peering optimization in IP vs. 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. Last but not least, state-of-the-art packet and optical networks use sophisticated but complex technologies, and for a network engineer it may not be trivial to be

a full expert in both areas. As a result, packet and optical networks are often operated in technical and organizational silos.

This separation is inefficient for many reasons. Both capital expenditure (CAPEX) and operational expenditure (OPEX) could be significantly reduced by better 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, key scenarios for Packet Optical Integration (POI) are described from the packet service layer perspective. The objective is to explain the benefit and the impact for both the packet and the optical layer, and to identify the required coordination between both layers. 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 client of optical DWDM networks. The scenarios are ordered by increasing 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 to better 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 a number of 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:

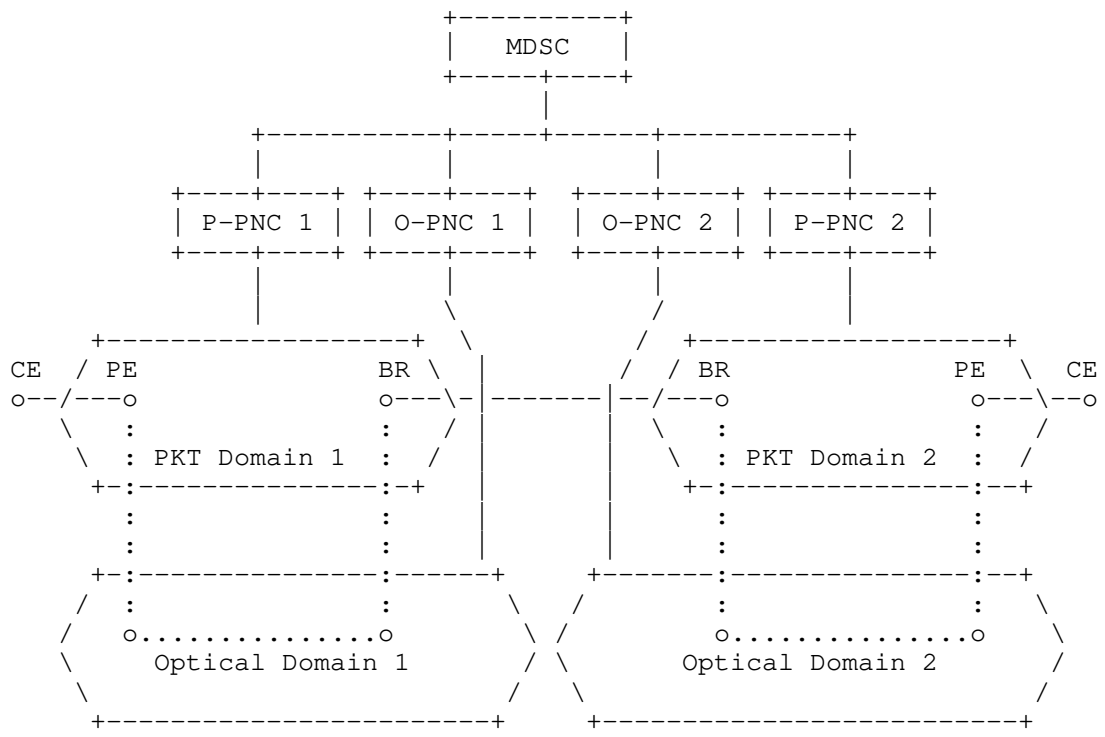


Figure 1 - Reference Scenario

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, and each Optical PNC (O-PNC) 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:

- 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;

- 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;

- The interfaces between the Routers and the Optical NEs are "Ethernet" physical interfaces;

- 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 technologies (e.g. QSFP-DD ZR 400G) are making convenient to fit the DWDM pluggable interfaces on the Routers, the deployment of those pluggable is not yet widely adopted by the operators. The reason is that most of operators are not yet ready to manage Packet and Transport networks in a unified single domain. As

a consequence, this draft is not addressing the unified scenario. This matter 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 the adaptation to the relevant network models. Such case is represented in Figure 2 of [RFC8453]
2. An implementation can choose to split the service-related and the network-related functions in 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 betweeh 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 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 key part of the architecture framework for a service provider:

- to abstract (through MDSC and PNCs) the underlying transport network complexity to the Business Systems Support layer

- to coordinate NFV, Transport (e.g. IP, Optical and Microwave networks), Fixed Acess, Core and Radio domains enabling full automation of end-to-end services to the end customers.

- 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 as well as the interface toward the MDSC are usually operator-specific and outside the scope of this draft. This document assumes that the OSS/Orchestrator requests MDSC to setup L2VPN/L3VPN services through mechanisms which are outside the scope of the draft.

There are two main cases when MDSC coordination of underlying PNCs in POI context is initiated:

- Initiated by a request from the OSS/Orchestration layer to setup L2VPN/L3VPN services that requires multi-layer/multi-domain coordination.

- Initiated by the MDSC itself to perform multi-layer/multi-domain optimizations and/or maintenance works, beyond discovery (e.g. rerouting LSPs with their associated services when putting a resource, like a fibre, in maintenance mode during a maintenance window). Different to service fulfillment, the workflows then are not related at all to a service provisioning request being received from the OSS/Orchestration layer.

Above two MDSC workflow cases are in the scope of this draft or in future versions.

2.1. L2/L3VPN Service Request in North Bound of MDSC

As explained in section 2, the OSS/Orchestration layer can request the MDSC to setup of L2/L3VPN services (with or without TE requirements).

Although the interface between the OSS/Orchestration layer is usually operator-specific, ideally it would be using a RESTCONF/YANG interface with more abstracted version of the MPI YANG data models used for network configuration (e.g. L3NM, L2NM).

Figure 2 shows an example of a possible control flow between the OSS/Orchestration layer and the MDSC to instantiate L2/L3VPN

services, using the YANG models under definition in [VN], [L2NM], [L3NM] and [TSM].

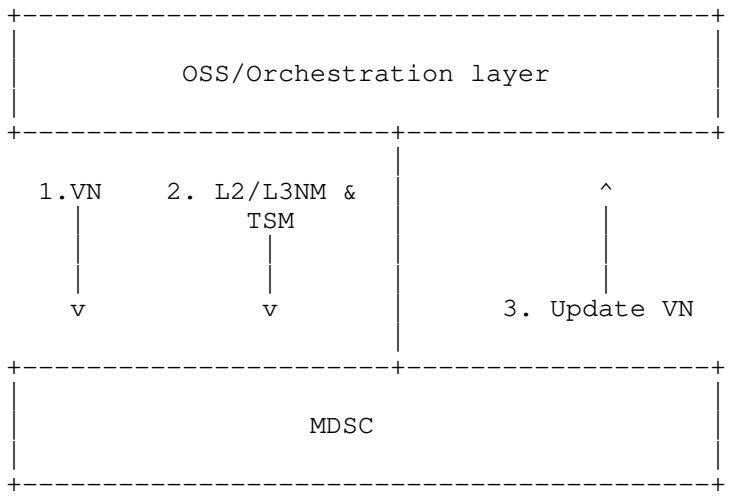


Figure 2 Service Request Process

- The VN YANG model [VN], whose primary focus is the CMI, can also be used to provide VN Service configuration from a orchestrated connectivity service point of view, when the L2/L3VPN service has TE requirements. This model is not used to setup L2/L3VPN service with no TE requirements.
  - 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 between the connectivity matrix with the VN member from which 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).
  - The model also provides VN-level preference information (e.g., VN member diversity) and VN-level admin-status and operational-status.
- 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.

- 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.

- The TE & Service Mapping YANG model [TSM] provides TE-service mapping as well as site mapping.

  - TE-service mapping provides the mapping between a L2/L3VPN instance and the corresponding VN instances.

  - The TE-service mapping also provides the service mapping requirement type as to how each L2/L3VPN/VN instance is created with respect to 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 detailed discussion on the mapping requirement types.

  - 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 decouples 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 which 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.

For the first three types, there could be additional TE binding requirements 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 fulfill the 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 opeerators for some references on real deployment]

2.2.1. Hard Isolation

For example, when "Hard Isolation with or w/o 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 lock 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 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 as well as 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 speakeers in each P-PNC, if needed.

2.3. IP/MPLS Domain Controller and NE Functions

IP/MPLS networks are assumed to have multiple domains, where 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:

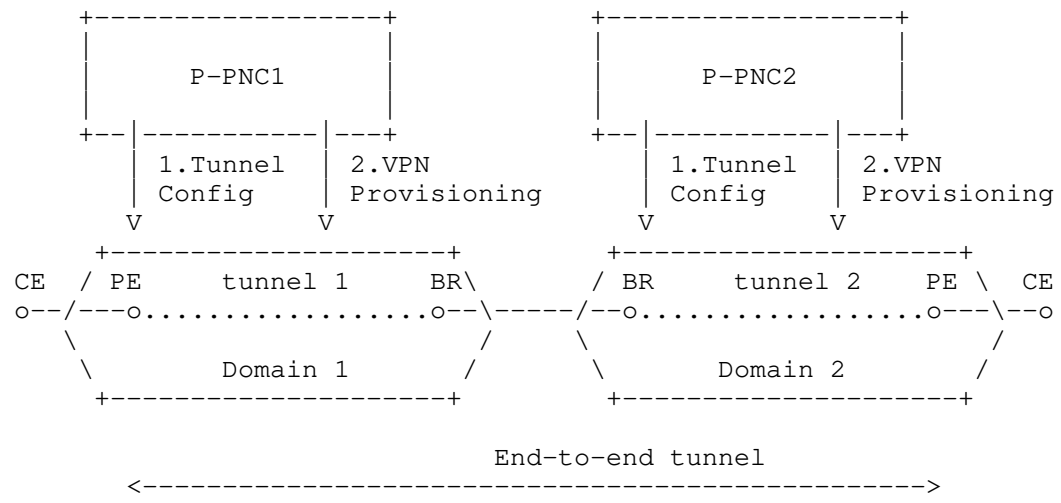


Figure 3 IP/MPLS Domain Controller & NE Functions

It is assumed that BGP is running in the inter-domain IP/MPLS networks for L2/L3VPN and that 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 label distribution of the end-to-end tunnel on PE and BR nodes. The MDSC is responsible for the selection of the BRs and of the intra-domain MPLS-TE Tunnels between PE/BR nodes.

If new MPLS-TE Tunnels are needed or mofications (e.g., bandwidth ingrease) 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 the selection of 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:

- VPN label: assigned by the egress PE and distributed by BGP;
- end-to-end LSP label: assigned by the egress BR, selected by the MDSC, and distributed by BGP;

- 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

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:

- provide to the MDSC an abstract TE topology view of its underlying optical network resources;
- perform single-domain local path computation, when requested by the MDSC;
- 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-speicific and outside the scope of this document.

Depending on the type of optical network, TE topology abstraction, path compution 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 which are applicable at all the MPI interfaces, between each PNC (Optical or Packet) and the MDSC, and also to 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. 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:

- The Base Network Model, defined in the "ietf-network" YANG module of [RFC8345]
- The Base Network Topology Model, defined in the "ietf-network-topology" YANG module of [RFC8345], which augments the Base Network Model
- 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:

- Dynamic Subscription to YANG Events and Datastores over RESTCONF as defined in [RFC8650]
- 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:

- 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].

- Optionally, when the OTN layer is used, the OTN Topology Model, as defined in the "ietf-otn-topology" YANG module of [OTN-TOPO].

- The Ethernet Topology Model, defined in the "ietf-eth-te-topology" YANG module of [CLIENT-TOPO].

- 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].

- 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:

- The TE Tunnel Model, defined in the "ietf-te" YANG module of [TE-TUNNEL]

- 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]

- Optionally, when the OTN layer is used, the OTN Tunnel Model, defined in the "ietf-otn-tunnel" YANG module of [OTN-TUNNEL].

- 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, alternatively, the Flexi-grid Media Channel Model are 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:

- The L3 Topology Model, defined in the "ietf-l3-unicast-topology" YANG modules of [RFC8346], which augments the Base Network Topology Model
- 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" in draft-ietf-teas-l3-te-topo [L3-TE-TOPO]
- 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).

- 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.
- 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.

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 IP PNCs are also identified.

4.1. Scenario 1: network and service 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 (NEs) 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).MDSC discovers also the IP/MPLS transport services (L2VPN/L3VPN) deployed, both intra-domain and inter-domain wise.

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, and 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 understand both the inter-AS domain links (seen by each controller as UNI interfaces but as I-NNI interfaces by the MDSC) as well as the cross-layer mapping between IP and WDM layer.

The MDSC also maintains an up-to-date network database of both IP and WDM layers (and optionally OTN layer) through the use of IETF notifications through MPI with the PNCs when any topology change occurs. It should be possible also to correlate information coming from IP and WDM layers (e.g.: which port, lambda/OTSi, 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 be able to correlate automatically 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 main to backup path etc.). As specified in [RFC7923] MDSC must be able to 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:

- Links between two IP domains (ASes)
- 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 understa how to merge the 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 together using the plug-id attribute defined in the TE Topology Model [RFC8795], as described in as described in section 4.3 of [RFC8795].

A more detailed description of how the plug-id can be used to discover inter-domain link 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. 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 as well as of the LLDP information within the plug-id value is implementation specific and needs to be consistent across all the PNCs.

4.1.2. IP Link Setup Procedure

The MDSC requires the O-PNC to setup a WDM Tunnel (either a WSON Tunnel or a Flexi grid Tunnel) within the DWDM network between the two Optical Transponders (OTs) associated with the two access links.

The Optical Transponders are reported by the O-PNC as Trail Termination Points (TTPs), defined in [TE TOPO], within the WDM Topology. The association between the Ethernet access link and the WDM TTP is reported by the Inter Layer Lock (ILL) identifiers, defined in [TE TOPO], reported by the O PNC within the Ethernet Topology and WDM Topology.

The MDSC also requires the O-PNC to steer the Ethernet client traffic between the two access Ethernet Links over the WDM Tunnel.

After the WDM 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.

4.2. L2VPN/L3VPN establishment

To be added

[Editor's Note] What mechanism would convey on the interface to the IP/MPLS domain controllers as well as on the SBI (between IP/MPLS domain controllers and IP/MPLS PE routers) the TE binding policy dynamically for the L3VPN? Typically, VRF is the function of the device that participate MP-BGP in MPLS VPN. With current MP-BGP implementation in MPLS VPN, the VRF's BGP next hop is the destination PE and the mapping to a tunnel (either an LDP or a BGP tunnel) toward the destination PE is done by automatically without any configuration. It is to be determined the impact on the PE VRF operation when the tunnel is an optical bypass tunnel which does not participate either LDP or BGP.

New text to answer the yellow part:

The MDSC Network-related function will then coordinate with the PNCs involved in the process to provide the provisioning information through ACTN MDSC to PNC (MPI) interface. The relevant data models used at the MPI may be in the form of L3NM, L2NM or others and are exchanged through MPI API calls. Through this process MDSC Network-related functions provide the configuration information to realize a VPN service to PNCs. For example, this process will inform PNCs on what PE routers compose a L3VPN, the topology requested, the VPN attributes, etc.

At the end of the process PNCs will deliver the actual configuration to the devices (either physical or virtual), through the ACTN Southbound Interface (SBI). In this case the configuration policies may be exchanged using a Netconf session delivering configuration commands associated to device-specific data models (e.g. BGP[], QOS [], etc.).

Having the topology information of the network domains under their control, PNCs will deliver all the information necessary to create, update, optimize or delete the tunnels connecting the PE nodes as requested by the VPN instantiation.

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 the collection of 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.

[WSON-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.

[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-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.

[CLIENT-SIGNAL] Zheng, H. et al., "A YANG Data Model for Transport Network Client Signals", draft-ietf-ccamp-client-signal-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. Fioccola, 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.

[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.

[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 backup 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-optimisation 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

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: Standards Track
Expires: May 5, 2021

Y. Lee, Ed.
Samsung Electronics
D. Dhody, Ed.
Huawei Technologies
D. Ceccarelli
Ericsson
I. Bryskin
Individual
B. Yoon
ETRI

November 1, 2020

A YANG Data Model for VN Operation
draft-ietf-teas-actn-vn-yang-10

Abstract

This document provides a YANG data model generally applicable to any mode of Virtual Network (VN) operation.

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 May 5, 2021.

Copyright Notice

Table of Contents

1. Introduction

This document provides a YANG [RFC7950] data model generally applicable to any mode of Virtual Network (VN) operation.

The VN model defined in this document is applicable in generic sense as an independent model in and of itself. The VN model defined in this document can also work together with other customer service models such as L3SM [RFC8299], L2SM [RFC8466] and L1CSM [I-D.ietf-ccamp-l1csm-yang] to provide a complete life-cycle service management and operations.

The YANG model discussed in this document basically provides the following:

- Characteristics of Access Points (APs) that describe customer's end point characteristics;
- Characteristics of Virtual Network Access Points (VNAP) that describe how an AP is partitioned for multiple VNs sharing the AP and its reference to a Link Termination Point (LTP) of the Provider Edge (PE) Node;
- Characteristics of Virtual Networks (VNs) that describe the customer's VN in terms of multiple VN Members comprising a VN, multi- source and/or multi-destination characteristics of the VN Member, the VN's reference to TE-topology's Abstract Node;

The actual VN instantiation and computation is performed with Connectivity Matrices sub-module of TE-Topology Model [RFC8795] which provides TE network topology abstraction and management operation. Once TE-topology Model is used in triggering VN instantiation over the networks, TE-tunnel [I-D.ietf-teas-yang-te] Model will inevitably interact with TE-Topology model for setting up actual tunnels and LSPs under the tunnels.

Abstraction and Control of Traffic Engineered Networks (ACTN) describes a set of management and control functions used to operate one or more TE networks to construct virtual networks that can be represented to customers and that are built from abstractions of the underlying TE networks [RFC8453]. ACTN is the primary example of the usage of the VN YANG model.

Sections 2 and 3 provide the discussion of how the VN YANG model is applicable to the ACTN context where Virtual Network Service (VNS) operation is implemented for the Customer Network Controller (CNC)-Multi-Domain Service Coordinator (MSDC) interface (CMI).

The YANG model on the CMI is also known as customer service model in [RFC8309]. The YANG model discussed in this document is used to operate customer-driven VNs during the VN instantiation, VN computation, and its life-cycle service management and operations.

The VN operational state is included in the same tree as the configuration consistent with Network Management Datastore Architecture (NMDA) [RFC8342]. The origin of the data is indicated as per the origin metadata annotation.

1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

1.1.1. Requirements Language

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. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

1.3. 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
vn	ietf-vn	[RFCXXXX]
inet	ietf-inet-types	[RFC6991]
nw	ietf-network	[RFC8345]
nt	ietf-network-topology	[RFC8345]
te-types	ietf-te-types	[RFC8776]
te-topo	ietf-te-topology	[RFC8795]

Table 1: Prefixes and corresponding YANG modules

Note: The RFC Editor will replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

2. Use-case of VN YANG Model in the ACTN context

In this section, ACTN is being used to illustrate the general usage of the VN YANG model. The model presented in this section has the following ACTN context.

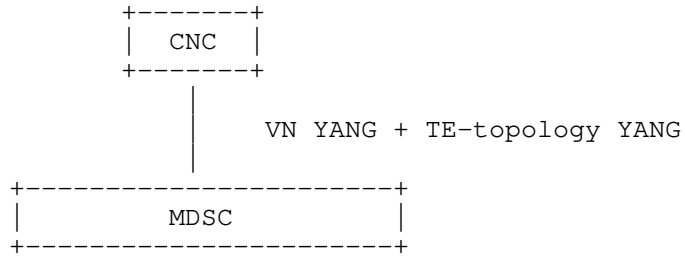


Figure 1: ACTN CMI

Both ACTN VN YANG and TE-topology models are used over the CMI to establish a VN over TE networks.

2.1. Type 1 VN

As defined in [RFC8453], a Virtual Network is a customer view of the TE network. To recapitulate VN types from [RFC8453], Type 1 VN is defined as follows:

The VN can be seen as a set of edge-to-edge abstract links (a Type 1 VN). Each abstract link is referred to as a VN member and is formed as an end-to-end tunnel across the underlying networks. Such tunnels may be constructed by recursive slicing or abstraction of paths in the underlying networks and can encompass edge points of the customer's network, access links, intra-domain paths, and inter-domain links.

If we were to create a VN where we have four VN-members as follows:

VN-Member 1	L1-L4
VN-Member 2	L1-L7
VN-Member 3	L2-L4
VN-Member 4	L3-L8

Where L1, L2, L3, L4, L7 and L8 correspond to a Customer End-Point, respectively.

This VN can be modeled as one abstract node representation as follows in Figure 2:

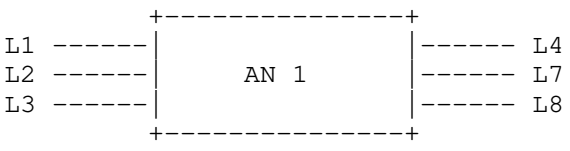


Figure 2: Abstract Node (One node topology)

Modeling a VN as one abstract node is the easiest way for customers to express their end-to-end connectivity; however, customers are not limited to express their VN only with one abstract node.

2.2. Type 2 VN

For some VN members of a VN, the customers are allowed to configure the actual path (i.e., detailed virtual nodes and virtual links) over the VN/abstract topology agreed mutually between CNC and MDSC prior to or a topology created by the MDSC as part of VN instantiation. Type 1 VN is a higher abstraction of a Type 2 VN.

If a Type 2 VN is desired for some or all of VN members of a type 1 VN (see the example in Section 2.1), the TE-topology model can provide the following abstract topology (that consists of virtual nodes and virtual links) which is built under the Type 1 VN.

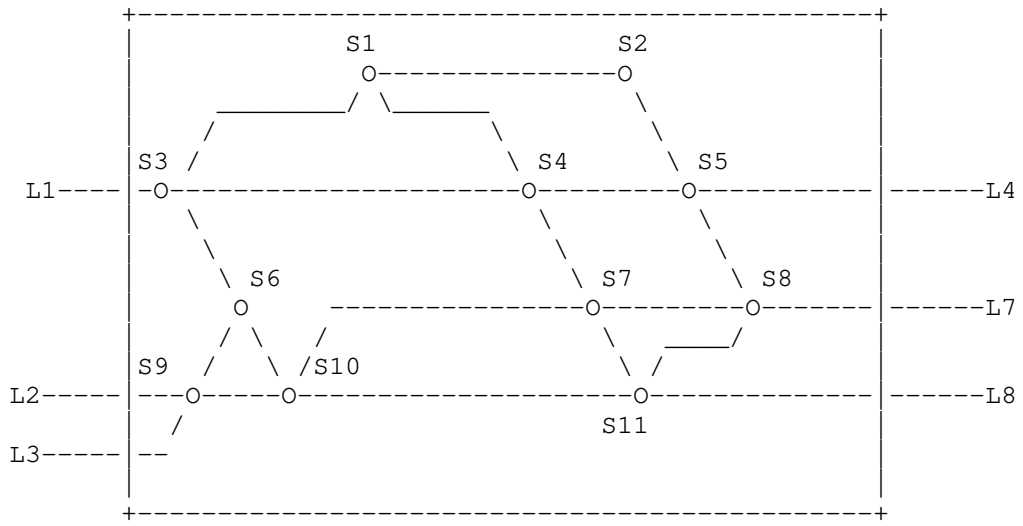


Figure 3: Type 2 topology

As you see from Figure 3, the Type 1 abstract node is depicted as a Type 1 abstract topology comprising of detailed virtual nodes and virtual links.

As an example, if VN-member 1 (L1-L4) is chosen to configure its own path over Type 2 topology, it can select, say, a path that consists of the ERO {S3,S4,S5} based on the topology and its service requirement. This capability is enacted via TE-topology configuration by the customer.

3. High-Level Control Flows with Examples

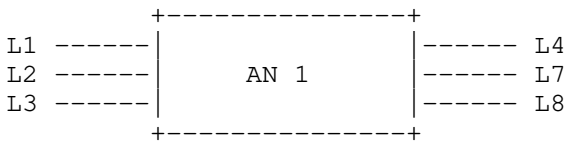
3.1. Type 1 VN Illustration

If we were to create a VN where we have four VN-members as follows:

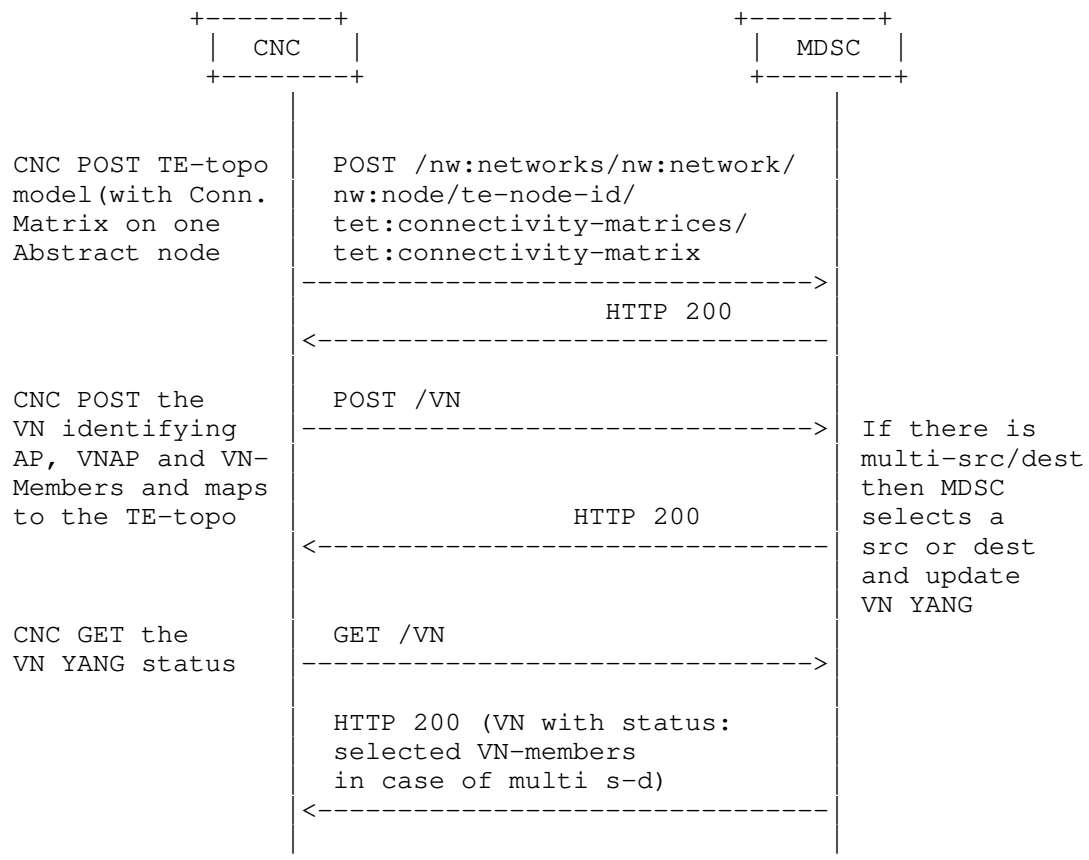
VN-Member 1	L1-L4
VN-Member 2	L1-L7
VN-Member 3	L2-L4
VN-Member 4	L3-L8

Where L1, L2, L3, L4, L7 and L8 correspond to Access Points.

This VN can be modeled as one abstract node representation as follows:



If this VN is Type 1, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using VN and TE-Topology Models.



3.2. Type 2 VN Illustration

For some VN members, the customer may want to "configure" explicit routes over the path that connects its two end-points. Let us consider the following example.

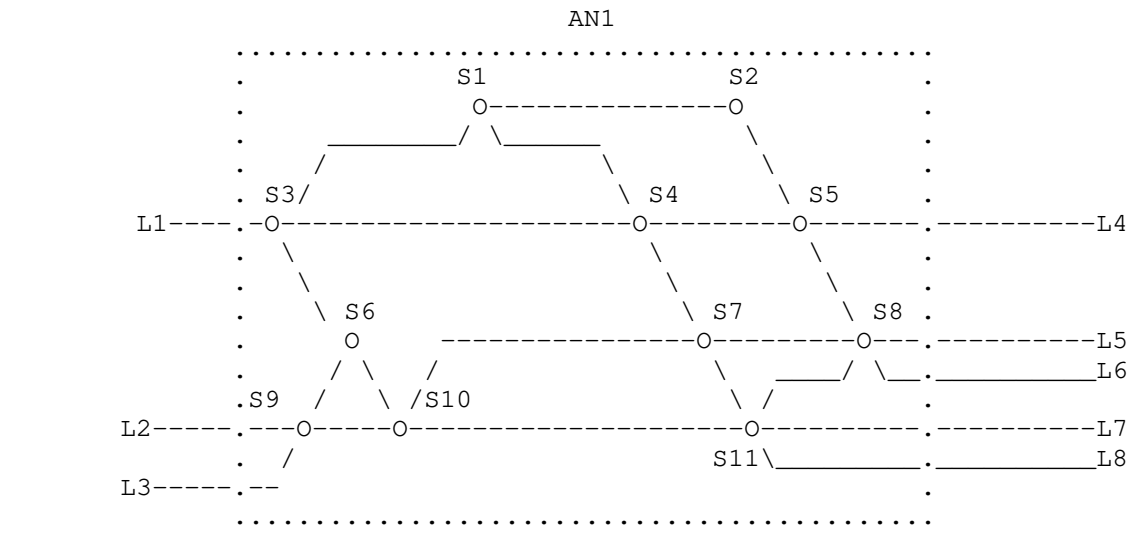
VN-Member 1 L1-L4 (via S3, S4, and S5)

VN-Member 2 L1-L7 (via S3, S4, S7 and S8)

VN-Member 3 L2-L7 (via S9, S10, and S11)

VN-Member 4 L3-L8 (via S9, S10 and S11)

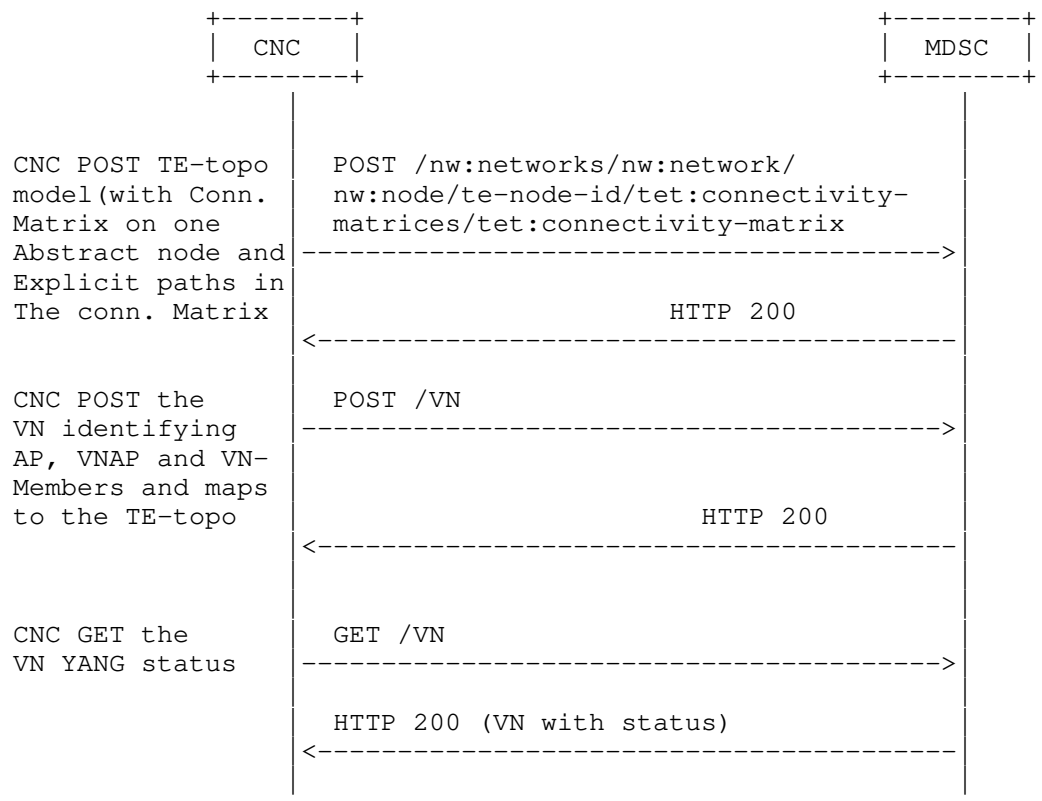
Where the following topology is the underlay for Abstraction Node 1 (AN1).



There are two options depending on whether CNC or MDSC creates the single abstract node topology.

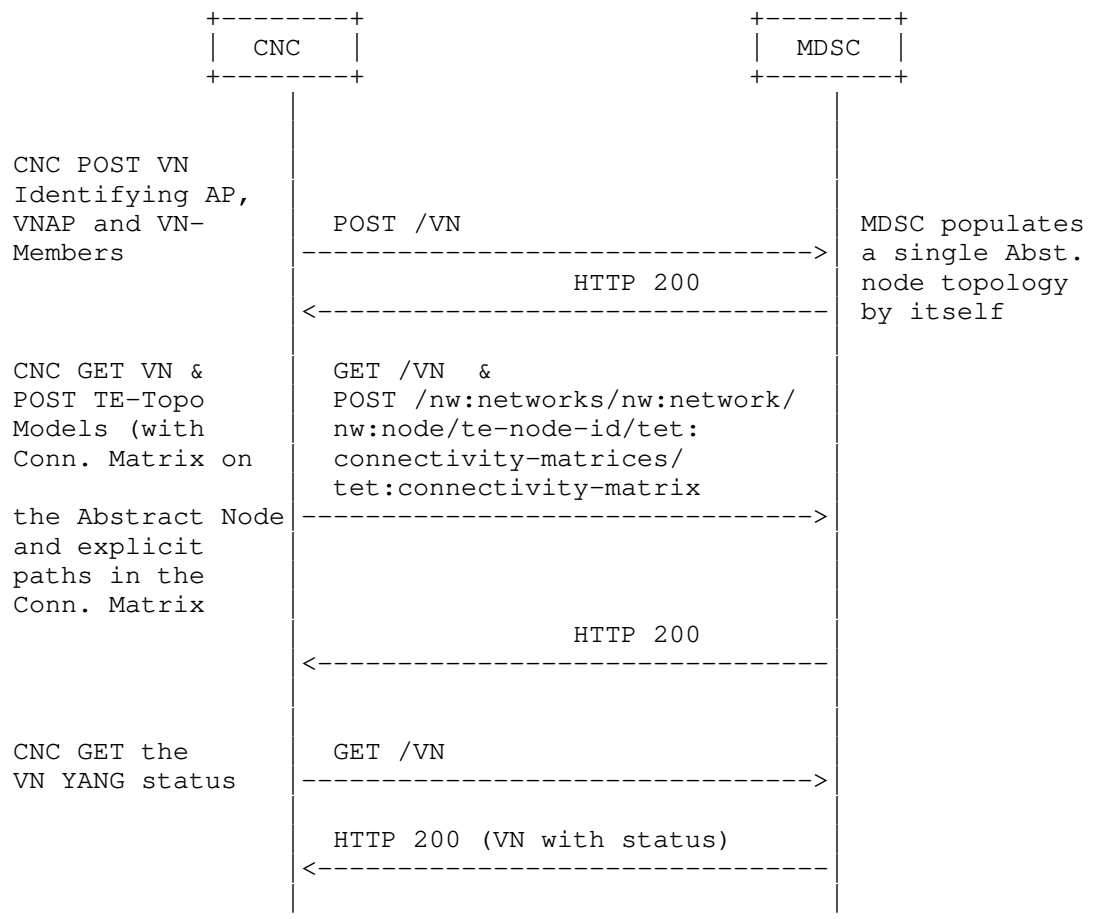
Case 1:

If CNC creates the single abstract node topology, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using VN and TE-Topology Model.



Case 2:

On the other hand, if MDSC create the single abstract node topology based VN YANG posted by the CNC, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using VN and TE-Topology Models.



Section 7 provides JSON examples for both VN model and TE-topology Connectivity Matrix sub-model to illustrate how a VN can be created by the CNC making use of the VN module as well as the TE-topology Connectivity Matrix module.

3.2.1. VN and AP Usage

The customer access information may be known at the time of VN creation. A shared logical AP identifier is used between the customer and the operator to identify the access link between Customer Edge (CE) and Provider Edge (PE) . This is described in Section 6 of [RFC8453].

In some VN operations, the customer access may not be known at the initial VN creation. The VN operation allow a creation of VN with

only PE identifier as well. The customer access information could be added later.

To achieve this the 'ap' container has a leaf for 'pe' node that allows AP to be created with PE information. The vn-member (and vn) could use APs that only have PE information initially.

4. VN Model Usage

4.1. Customer view of VN

The VN-YANG model allows to define a customer view, and allows the customer to communicate using the VN constructs as described in the [RFC8454]. It also allows to group the set of edge-to-edge links (i.e., VN members) under a common umbrella of VN. This allows the customer to instantiate and view the VN as one entity, making it easier for some customers to work on VN without worrying about the details of the provider based YANG models.

This is similar to the benefits of having a separate YANG model for the customer services as described in [RFC8309], which states that service models do not make any assumption of how a service is actually engineered and delivered for a customer.

4.2. Auto-creation of VN by MDSC

The VN could be configured at the MDSC explicitly by the CNC using the VN YANG model. In some other cases, the VN is not explicitly configured, but created automatically by the MDSC based on the customer service model and local policy, even in these case the VN YANG model can be used by the CNC to learn details of the underlying VN created to meet the requirements of customer service model.

4.3. Innovative Services

4.3.1. VN Compute

VN Model supports VN compute (pre-instantiation mode) to view the full VN as a single entity before instantiation. Achieving this via path computation or "compute only" tunnel setup does not provide the same functionality.

The VN compute RPC allow you to optionally include the constraints and the optimization criteria at the VN as well as at the individual VN-member level. Thus, the RPC can be used independently without creating an abstract topology first.

4.3.2. Multi-sources and Multi-destinations

In creating a virtual network, the list of sources or destinations or both may not be pre-determined by the customer. For instance, for a given source, there may be a list of multiple-destinations to which the optimal destination may be chosen depending on the network resource situations. Likewise, for a given destination, there may also be multiple-sources from which the optimal source may be chosen. In some cases, there may be a pool of multiple sources and destinations from which the optimal source-destination may be chosen. The following YANG module is shown for describing source container and destination container. The following YANG tree shows how to model multi-sources and multi-destinations.

```
  +--rw vn
     +--rw vn-list* [vn-id]
        +--rw vn-id                  vn-id
        +--rw vn-topology-id?        te-types:te-topology-id
        +--rw abstract-node?
        |       -> /nw:networks/network/node/tet:te-node-id
        +--rw vn-member-list* [vn-member-id]
        |  +--rw vn-member-id              vn-member-id
        |  +--rw src
        |  |  +--rw src?
        |  |  |       -> /ap/access-point-list/access-point-id
        |  |  +--rw src-vn-ap-id?
        |  |  |       -> /ap/access-point-list/vn-ap/vn-ap-id
        |  |  +--rw multi-src?      boolean {multi-src-dest}?
        |  +--rw dest
        |  |  +--rw dest?
        |  |  |       -> /ap/access-point-list/access-point-id
        |  |  +--rw dest-vn-ap-id?
        |  |  |       -> /ap/access-point-list/vn-ap/vn-ap-id
        |  |  +--rw multi-dest?      boolean {multi-src-dest}?
        |  +--rw connectivity-matrix-id?   leafref
        |  +--ro oper-status?              identityref
        +--ro if-selected?           boolean {multi-src-dest}?
        +--rw admin-status?          identityref
        +--ro oper-status?           identityref
        +--rw vn-level-diversity?    te-types:te-path-disjointness
```

4.3.3. Others

The VN YANG model can be easily augmented to support the mapping of VN to the Services such as L3SM and L2SM as described in [I-D.ietf-teas-te-service-mapping-yang].

The VN YANG model can be extended to support telemetry, performance monitoring and network autonomics as described in [I-D.ietf-teas-actn-pm-telemetry-autonomics].

4.3.4. Summary

This section summarizes the innovative service features of the VN YANG.

- Maintenance of AP and VNAP along with VN
- VN construct to group of edge-to-edge links
- VN Compute (pre-instantiate)
- Multi-Source / Multi-Destination
- Ability to support various VN and VNS Types
  - VN Type 1: Customer configures the VN as a set of VN Members. No other details need to be set by customer, making for a simplified operations for the customer.
  - VN Type 2: Along with VN Members, the customer could also provide an abstract topology, this topology is provided by the Abstract TE Topology YANG Model.

5. VN YANG Model (Tree Structure)

```
module: ietf-vn
  +--rw ap
  |  +--rw access-point-list* [access-point-id]
  |     +--rw access-point-id    access-point-id
  |     +--rw pe?
  |     |       -> /nw:networks/network/node/tet:te-node-id
  |     +--rw max-bandwidth?     te-types:te-bandwidth
  |     +--rw avl-bandwidth?     te-types:te-bandwidth
  |     +--rw vn-ap* [vn-ap-id]
  |        +--rw vn-ap-id         access-point-id
  |        +--rw vn?              -> /vn/vn-list/vn-id
  |        +--rw abstract-node?
  |        |       -> /nw:networks/network/node/tet:te-node-id
  |        +--rw ltp?             leafref
  |        +--ro max-bandwidth?   te-types:te-bandwidth
  +--rw vn
     +--rw vn-list* [vn-id]
        +--rw vn-id                 vn-id
        +--rw vn-topology-id?       te-types:te-topology-id
```

```
        +--rw abstract-node?
        |       -> /nw:networks/network/node/tet:te-node-id
        +--rw vn-member-list* [vn-member-id]
        |  +--rw vn-member-id              vn-member-id
        |  +--rw src
        |  |  +--rw src?
        |  |  |       -> /ap/access-point-list/access-point-id
        |  |  +--rw src-vn-ap-id?
        |  |  |       -> /ap/access-point-list/vn-ap/vn-ap-id
        |  |  +--rw multi-src?      boolean {multi-src-dest}?
        |  +--rw dest
        |  |  +--rw dest?
        |  |  |       -> /ap/access-point-list/access-point-id
        |  |  +--rw dest-vn-ap-id?
        |  |  |       -> /ap/access-point-list/vn-ap/vn-ap-id
        |  |  +--rw multi-dest?      boolean {multi-src-dest}?
        |  +--rw connectivity-matrix-id?   leafref
        |  +--ro oper-status?              identityref
        +--ro if-selected?          boolean {multi-src-dest}?
        +--rw admin-status?         identityref
        +--ro oper-status?          identityref
        +--rw vn-level-diversity?   te-types:te-path-disjointness

    rpcs:
      +---x vn-compute
         +---w input
         |  +---w abstract-node?
         |  |       -> /nw:networks/network/node/tet:te-node-id
         |  +---w path-constraints
         |  |  +---w te-bandwidth
         |  |  |  +---w (technology)?
         |  |  |          ...
         |  |  +---w link-protection?          identityref
         |  |  +---w setup-priority?           uint8
         |  |  +---w hold-priority?            uint8
         |  |  +---w signaling-type?           identityref
         |  |  +---w path-metric-bounds
         |  |  |  +---w path-metric-bound* [metric-type]
         |  |  |          ...
         |  |  +---w path-affinities-values
         |  |  |  +---w path-affinities-value* [usage]
         |  |  |          ...
         |  |  +---w path-affinity-names
         |  |  |  +---w path-affinity-name* [usage]
         |  |  |          ...
         |  |  +---w path-srlgs-lists
         |  |  |  +---w path-srlgs-list* [usage]
         |  |  |          ...
```

```
|   |  +---w path-srlgs-names
|   |  |  +---w path-srlgs-name* [usage]
|   |  |  |       ...
|   |  +---w disjointness?             te-path-disjointness
|   +---w optimizations
|   |  +---w (algorithm)?
|   |     +--:(metric) {path-optimization-metric}?
|   |     |     ...
|   |     +--:(objective-function)
|   |              {path-optimization-objective-function}?
|   |           ...
|   +---w vn-member-list* [vn-member-id]
|   |  +---w vn-member-id              vn-member-id
|   |  +---w src
|   |  |  +---w src?
|   |  |  |       -> /ap/access-point-list/access-point-id
|   |  |  +---w src-vn-ap-id?
|   |  |  |       -> /ap/access-point-list/vn-ap/vn-ap-id
|   |  |  +---w multi-src?      boolean {multi-src-dest}?
|   |  +---w dest
|   |  |  +---w dest?
|   |  |  |       -> /ap/access-point-list/access-point-id
|   |  |  +---w dest-vn-ap-id?
|   |  |  |       -> /ap/access-point-list/vn-ap/vn-ap-id
|   |  |  +---w multi-dest?      boolean {multi-src-dest}?
|   |  +---w connectivity-matrix-id?   leafref
|   |  +---w path-constraints
|   |  |  +---w te-bandwidth
|   |  |  |     ...
|   |  |  +---w link-protection?          identityref
|   |  |  +---w setup-priority?           uint8
|   |  |  +---w hold-priority?            uint8
|   |  |  +---w signaling-type?           identityref
|   |  |  +---w path-metric-bounds
|   |  |  |     ...
|   |  |  +---w path-affinities-values
|   |  |  |     ...
|   |  |  +---w path-affinity-names
|   |  |  |     ...
|   |  |  +---w path-srlgs-lists
|   |  |  |     ...
|   |  |  +---w path-srlgs-names
|   |  |  |     ...
|   |  |  +---w disjointness?             te-path-disjointness
|   |  +---w optimizations
|   |     +---w (algorithm)?
|   |           ...
|   +---w vn-level-diversity?   te-types:te-path-disjointness
```

```
         +--ro output
            +--ro vn-member-list* [vn-member-id]
               +--ro vn-member-id              vn-member-id
               +--ro src
               |  +--ro src?
               |  |       -> /ap/access-point-list/access-point-id
               |  +--ro src-vn-ap-id?
               |  |       -> /ap/access-point-list/vn-ap/vn-ap-id
               |  +--ro multi-src?      boolean {multi-src-dest}?
               +--ro dest
               |  +--ro dest?
               |  |       -> /ap/access-point-list/access-point-id
               |  +--ro dest-vn-ap-id?
               |  |       -> /ap/access-point-list/vn-ap/vn-ap-id
               |  +--ro multi-dest?      boolean {multi-src-dest}?
               +--ro connectivity-matrix-id?   leafref
               +--ro if-selected?              boolean
               |       {multi-src-dest}?
               +--ro compute-status?           identityref
```

6. VN YANG Model

The YANG model is as follows:

```
<CODE BEGINS> file "ietf-vn@2020-11-02.yang"
module ietf-vn {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vn";
  prefix vn;

  /* Import inet-types */

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }

  /* Import network */

  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
```

```
     /* Import network topology */

     import ietf-network-topology {
       prefix nt;
       reference
         "RFC 8345: A YANG Data Model for Network Topologies";
     }

     /* Import TE Common types */

     import ietf-te-types {
       prefix te-types;
       reference
         "RFC 8776: Common YANG Data Types for Traffic Engineering";
     }

     /* Import TE Topology */

     import ietf-te-topology {
       prefix tet;
       reference
         "RFC 8795: YANG Data Model for Traffic Engineering (TE)
          Topologies";
     }

     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: Young Lee <younglee.tx@gmail.com>
              : Dhruv Dhody <dhruv.ietf@gmail.com>";
     description
       "This module contains a YANG module for the VN. It describes a
        VN operation module that takes place in the context of the
        CNC-MDSC Interface (CMI) of the ACTN architecture where the
        CNC is the actor of a VN Instantiation/modification/deletion
        as per RFC 8453.

        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
```

```
        (https://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC XXXX; see the
        RFC itself for full legal notices.

        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 (RFC 2119) (RFC 8174) when, and only when,
        they appear in all capitals, as shown here.";

    revision 2020-11-02 {
      description
        "initial version.";
      reference
        "RFC XXXX: A YANG Data Model for VN Operation";
    }

    /* Features */

    feature multi-src-dest {
      description
        "Support for selection of one src or destination
         among multiple.";
      reference
        "RFC 8453: Framework for Abstraction and Control of TE
         Networks (ACTN)";
    }

    /* Identity VN State*/

    identity vn-state-type {
      description
        "Base identity for VN state";
    }

    identity vn-state-up {
      base vn-state-type;
      description
        "VN state up";
    }

    identity vn-state-down {
      base vn-state-type;
      description
        "VN state down";
    }
```

```
     /* Identity VN Admin State*/

     identity vn-admin-state-type {
       description
         "Base identity for VN admin states";
     }

     identity vn-admin-state-up {
       base vn-admin-state-type;
       description
         "VN administratively state up";
     }

     identity vn-admin-state-down {
       base vn-admin-state-type;
       description
         "VN administratively state down";
     }

     /* Identity VN Compute State*/

     identity vn-compute-state-type {
       description
         "Base identity for compute states";
     }

     identity vn-compute-state-computing {
       base vn-compute-state-type;
       description
         "State path compute in progress";
     }

     identity vn-compute-state-computation-ok {
       base vn-compute-state-type;
       description
         "State path compute successful";
     }

     identity vn-compute-state-computation-failed {
       base vn-compute-state-type;
       description
         "State path compute failed";
     }

     /* Typedef */

     typedef vn-id {
       type inet:uri;
```

```
    description
      "Identifier for a VN.  The precise structure of the
       vn-id will be up to the implementation. The
       identifier SHOULD be chosen such that the same VN
       will always be identified through the same
       identifier, even if the data model is instantiated
       in separate datastores.";
  }

  typedef access-point-id {
    type inet:uri;
    description
      "Identifier for an AP.  The precise structure of the
       access-point-id will be up to the implementation.
       The identifier SHOULD be chosen such that the same AP
       will always be identified through the same
       identifier, even if the data model is instantiated
       in separate datastores. This type is used for both AP
       and VNAP";
  }

  typedef vn-member-id {
    type inet:uri;
    description
      "Identifier for a VN member.  The precise structure of
       the vn-member-id will be up to the implementation.
       The identifier SHOULD be chosen such that the same VN
       member will always be identified through the same
       identifier, even if the data model is instantiated
       in separate datastores. ";
  }

  /* Groupings */

  grouping vn-ap {
    description
      "VNAP related information";
    leaf vn-ap-id {
      type access-point-id;
      description
        "A unique identifier for the referred VNAP";
    }
    leaf vn {
      type leafref {
        path "/vn/vn-list/vn-id";
      }
      description
        "A reference to the VN";
```

```
      }
      leaf abstract-node {
        type leafref {
          path "/nw:networks/nw:network/nw:node/tet:te-node-id";
        }
        description
          "A reference to the abstract node in TE Topology that
           represent the VN";
      }
      leaf ltp {
        type leafref {
          path "/nw:networks/nw:network/nw:node/"
             + "nt:termination-point/tet:te-tp-id";
        }
        description
          "A reference LTP in the TE-topology";
        reference
          "RFC 8795: YANG Data Model for Traffic Engineering (TE)
           Topologies";
      }
      leaf max-bandwidth {
        type te-types:te-bandwidth;
        config false;
        description
          "The max bandwidth of the VNAP";
      }
      reference
        "RFC 8453: Framework for Abstraction and Control of TE
         Networks (ACTN)";
    } //vn-ap

    grouping access-point {
      description
        "AP related information";
      leaf access-point-id {
        type access-point-id;
        description
          "A unique identifier for the referred access point";
      }
      leaf pe {
        type leafref {
          path "/nw:networks/nw:network/nw:node/tet:te-node-id";
        }
        description
          "A reference to the PE node in the native TE Topology";
      }
      leaf max-bandwidth {
        type te-types:te-bandwidth;
```

```
        description
          "The max bandwidth of the AP";
      }
      leaf avl-bandwidth {
        type te-types:te-bandwidth;
        description
          "The available bandwidth of the AP";
      }
      /*add details and any other properties of AP,
      not associated by a VN
      CE port, PE port etc.
       */
      list vn-ap {
        key "vn-ap-id";
        uses vn-ap;
        description
          "List of VNAP in this AP";
      }
      reference
        "RFC 8453: Framework for Abstraction and Control of TE
         Networks (ACTN)";
    } //access-point

    grouping vn-member {
      description
        "The vn-member is described by this grouping";
      leaf vn-member-id {
        type vn-member-id;
        description
          "A vn-member identifier";
      }
      container src {
        description
          "The source of VN Member";
        leaf src {
          type leafref {
            path "/ap/access-point-list/access-point-id";
          }
          description
            "A reference to source AP";
        }
        leaf src-vn-ap-id {
          type leafref {
            path "/ap/access-point-list/vn-ap/vn-ap-id";
          }
          description
            "A reference to source VNAP";
        }
```

```
          leaf multi-src {
            if-feature "multi-src-dest";
            type boolean;
            description
              "Is the source part of multi-source, where
               only one of the source is enabled";
          }
        }
        container dest {
          description
            "the destination of VN Member";
          leaf dest {
            type leafref {
              path "/ap/access-point-list/access-point-id";
            }
            description
              "A reference to destination AP";
          }
          leaf dest-vn-ap-id {
            type leafref {
              path "/ap/access-point-list/vn-ap/vn-ap-id";
            }
            description
              "A reference to dest VNAP";
          }
          leaf multi-dest {
            if-feature "multi-src-dest";
            type boolean;
            description
              "Is destination part of multi-destination, where only one
               of the destination is enabled";
          }
        }
        leaf connectivity-matrix-id {
          type leafref {
            path "/nw:networks/nw:network/nw:node/tet:te/"
               + "tet:te-node-attributes/"
               + "tet:connectivity-matrices/"
               + "tet:connectivity-matrix/tet:id";
          }
          description
            "A reference to connectivity-matrix";
          reference
            "RFC 8795: YANG Data Model for Traffic Engineering (TE)
             Topologies";
        }
        reference
          "RFC 8454: Information Model for Abstraction and Control of TE
```

```
          Networks (ACTN)";
      } //vn-member

      grouping vn-policy {
        description
          "policy for VN-level diverisity";
        leaf vn-level-diversity {
          type te-types:te-path-disjointness;
          description
            "The type of disjointness on the VN level (i.e., across all
             VN members)";
        }
      }

      /* Configuration data nodes */

      container ap {
        description
          "AP configurations";
        list access-point-list {
          key "access-point-id";
          description
            "access-point identifier";
          uses access-point {
            description
              "The access-point information";
          }
        }
        reference
          "RFC 8453: Framework for Abstraction and Control of TE
           Networks (ACTN)";
      }
      container vn {
        description
          "VN configurations";
        list vn-list {
          key "vn-id";
          description
            "A virtual network is identified by a vn-id";
          leaf vn-id {
            type vn-id;
            description
              "A unique VN identifier";
          }
          leaf vn-topology-id {
            type te-types:te-topology-id;
            description
              "An optional identifier to the TE Topology Model where the
```

```
            abstract nodes and links of the Topology can be found for
            Type 2 VNS";
        }
        leaf abstract-node {
          type leafref {
            path "/nw:networks/nw:network/nw:node/tet:te-node-id";
          }
          description
            "A reference to the abstract node in TE Topology";
        }
        list vn-member-list {
          key "vn-member-id";
          description
            "List of vn-members in a VN";
          uses vn-member;
          leaf oper-status {
            type identityref {
              base vn-state-type;
            }
            config false;
            description
              "The vn-member operational state.";
          }
        }
        leaf if-selected {
          if-feature "multi-src-dest";
          type boolean;
          default "false";
          config false;
          description
            "Is the vn-member is selected among the multi-src/dest
             options";
        }
        leaf admin-status {
          type identityref {
            base vn-admin-state-type;
          }
          default "vn-admin-state-up";
          description
            "VN administrative state.";
        }
        leaf oper-status {
          type identityref {
            base vn-state-type;
          }
          config false;
          description
            "VN operational state.";
```

```
        }
        uses vn-policy;
      } //vn-list
      reference
        "RFC 8453: Framework for Abstraction and Control of TE
         Networks (ACTN)";
    } //vn

    /* RPC */

    rpc vn-compute {
      description
        "The VN computation without actual instantiation";
      input {
        leaf abstract-node {
          type leafref {
            path "/nw:networks/nw:network/nw:node/tet:te-node-id";
          }
          description
            "A reference to the abstract node in TE Topology";
        }
        uses te-types:generic-path-constraints;
        uses te-types:generic-path-optimization;
        list vn-member-list {
          key "vn-member-id";
          description
            "List of VN-members in a VN";
          uses vn-member;
          uses te-types:generic-path-constraints;
          uses te-types:generic-path-optimization;
        }
        uses vn-policy;
      }
      output {
        list vn-member-list {
          key "vn-member-id";
          description
            "List of VN-members in a VN";
          uses vn-member;
          leaf if-selected {
            if-feature "multi-src-dest";
            type boolean;
            default "false";
            description
              "Is the vn-member is selected among the multi-src/dest
               options";
          }
          leaf compute-status {
```

```
            type identityref {
              base vn-compute-state-type;
            }
            description
              "The VN-member compute state.";
          }
        }
      }
    } //vn-compute

  }
```

<CODE ENDS>

7. JSON Example

This section provides json implementation examples as to how VN YANG model and TE topology model are used together to instantiate virtual networks.

The example in this section includes following VN

- VN1 (Type 1): Which maps to the single node topology abstract1 (node D1) and consist of VN Members 104 (L1 to L4), 107 (L1 to L7), 204 (L2 to L4), 308 (L3 to L8) and 108 (L1 to L8). We also show how disjointness (node, link, srlg) is supported in the example on the global level (i.e., connectivity matrices level).

- VN2 (Type 2): Which maps to the single node topology abstract2 (node D2), this topology has an underlay topology (absolute) (see figure in section 3.2). This VN has a single VN member 105 (L1 to L5) and an underlay path (S4 and S7) has been set in the connectivity matrix of abstract2 topology;

- VN3 (Type 1): This VN has a multi-source, multi-destination feature enable for VN Member 104 (L1 to L4)/107 (L1 to L7) {multi-src} and VN Member 204 (L2 to L4)/304 (L3 to L4) {multi-dest} usecase. The selected VN-member is known via the field "if-selected" and the corresponding connectivity-matrix-id.

Note that the VN YANG model also include the AP and VNAP which shows various VN using the same AP.

7.1. VN JSON

```
{
      "ap":{
         "access-point-list": [
```

```
                  {
                   "access-point-id": 101,
                   "access-point-name": "101",
                   "vn-ap": [
                      {
                         "vn-ap-id": 10101,
                         "vn": 1,
                         "abstract-node": "D1",
                         "ltp": "1-0-1"
                      },
                      {
                         "vn-ap-id": 10102,
                         "vn": 2,
                         "abstract-node": "D2",
                         "ltp": "1-0-1"
                      },
                      {
                         "vn-ap-id": 10103,
                         "vn": 3,
                         "abstract-node": "D3",
                         "ltp": "1-0-1"
                      },
                   ]
                },
                {
                   "access-point-id": 202,
                   "access-point-name": "202",
                   "vn-ap": [
                      {
                         "vn-ap-id": 20201,
                         "vn": 1,
                         "abstract-node": "D1",
                         "ltp": "2-0-2"
                      }
                   ]
                },
                {
                   "access-point-id": 303,
                   "access-point-name": "303",
                   "vn-ap": [
                      {
                         "vn-ap-id": 30301,
                         "vn": 1,
                         "abstract-node": "D1",
                         "ltp": "3-0-3"
                      },
                      {
                         "vn-ap-id": 30303,
```

```
                    "vn": 3,
                    "abstract-node": "D3",
                    "ltp": "3-0-3"
                }
            ]
        },
        {
            "access-point-id": 440,
            "access-point-name": "440",
            "vn-ap": [
                {
                    "vn-ap-id": 44001,
                    "vn": 1,
                    "abstract-node": "D1",
                    "ltp": "4-4-0"
                }
            ]
        },
        {
            "access-point-id": 550,
            "access-point-name": "550",
            "vn-ap": [
                {
                    "vn-ap-id": 55002,
                    "vn": 2,
                    "abstract-node": "D2",
                    "ltp": "5-5-0"
                }
            ]
        },
        {
            "access-point-id": 770,
            "access-point-name": "770",
            "vn-ap": [
                {
                    "vn-ap-id": 77001,
                    "vn": 1,
                    "abstract-node": "D1",
                    "ltp": "7-7-0"
                },
                {
                    "vn-ap-id": 77003,
                    "vn": 3,
                    "abstract-node": "D3",
                    "ltp": "7-7-0"
                }
            ]
        },
```

```
                {
                   "access-point-id": 880,
                   "access-point-name": "880",
                   "vn-ap": [
                      {
                         "vn-ap-id": 88001,
                         "vn": 1,
                         "abstract-node": "D1",
                         "ltp": "8-8-0"
                      },
                      {
                         "vn-ap-id": 88003,
                         "vn": 3,
                         "abstract-node": "D3",
                         "ltp": "8-8-0"
                      }
                   ]
                }
             ]
          },
          "vn":{
             "vn-list": [
                {
                   "vn-id": 1,
                   "vn-name": "vn1",
                   "vn-topology-id": "te-topology:abstract1",
                   "abstract-node": "D1",
                   "vn-member-list": [
                      {
                         "vn-member-id": 104,
                         "src": {
                            "src": 101,
                            "src-vn-ap-id": 10101,
                         },
                         "dest": {
                            "dest": 440,
                            "dest-vn-ap-id": 44001,
                         },
                         "connectivity-matrix-id": 104
                      },
                      {
                         "vn-member-id": 107,
                         "src": {
                            "src": 101,
                            "src-vn-ap-id": 10101,
                         },
                         "dest": {
                            "dest": 770,
```

```
                        "dest-vn-ap-id": 77001,
                     },
                     "connectivity-matrix-id": 107
                  },
                  {
                     "vn-member-id": 204,
                     "src": {
                        "src": 202,
                        "dest-vn-ap-id": 20401,
                     },
                     "dest": {
                        "dest": 440,
                        "dest-vn-ap-id": 44001,
                     },
                     "connectivity-matrix-id": 204
                  },
                  {
                     "vn-member-id": 308,
                     "src": {
                        "src": 303,
                        "src-vn-ap-id": 30301,
                     },
                     "dest": {
                        "dest": 880,
                        "src-vn-ap-id": 88001,
                     },
                     "connectivity-matrix-id": 308
                  },
                  {
                     "vn-member-id": 108,
                     "src": {
                        "src": 101,
                        "src-vn-ap-id": 10101,
                     },
                     "dest": {
                        "dest": 880,
                        "dest-vn-ap-id": 88001,
                     },
                     "connectivity-matrix-id": 108
                  }
               ]
            },
            {
               "vn-id": 2,
               "vn-name": "vn2",
               "vn-topology-id": "te-topology:abstract2",
               "abstract-node": "D2",
               "vn-member-list": [
```

```
                {
                   "vn-member-id": 105,
                   "src": {
                      "src": 101,
                      "src-vn-ap-id": 10102,
                   },
                   "dest": {
                      "dest": 550,
                      "dest-vn-ap-id": 55002,
                   },
                   "connectivity-matrix-id": 105
                }
             ]
          },
          {
             "vn-id": 3,
             "vn-name": "vn3",
             "vn-topology-id": "te-topology:abstract3",
             "abstract-node": "D3",
             "vn-member-list": [
                {
                   "vn-member-id": 104,
                   "src": {
                      "src": 101,
                   },
                   "dest": {
                      "dest": 440,
                      "multi-dest": true
                   }
                },
                {
                   "vn-member-id": 107,
                   "src": {
                      "src": 101,
                      "src-vn-ap-id": 10103,
                   },
                   "dest": {
                      "dest": 770,
                      "dest-vn-ap-id": 77003,
                      "multi-dest": true
                   },
                   "connectivity-matrix-id": 107,
                   "if-selected":true,
                },
                {
                   "vn-member-id": 204,
                   "src": {
                      "src": 202,
```

```
                              "multi-src": true,
                           },
                           "dest": {
                              "dest": 440,
                           },
                        },
                        {
                           "vn-member-id": 304,
                           "src": {
                              "src": 303,
                              "src-vn-ap-id": 30303,
                              "multi-src": true,
                           },
                           "dest": {
                              "dest": 440,
                              "src-vn-ap-id": 44003,
                           },
                           "connectivity-matrix-id": 304,
                           "if-selected":true,
                        },
                     ]
                  },

               ]
            }

         }
      }
```

7.2. TE-topology JSON

```
      {
           "networks": {
             "network": [
               {
                 "network-types": {
                   "te-topology": {}
                 },
                 "network-id": "abstract1",
                 "provider-id": 201,
                 "client-id": 600,
                 "te-topology-id": "te-topology:abstract1",
                 "node": [
                   {
                     "node-id": "D1",
                     "te-node-id": "2.0.1.1",
                     "te": {
```

```
                    "te-node-attributes": {
                      "domain-id" : 1,
                      "is-abstract": [null],
                      "connectivity-matrices": {
                        "is-allowed": true,
                        "path-constraints": {
                          "bandwidth-generic": {
                            "te-bandwidth": {
                              "generic": [
                                {
                                  "generic": "0x1p10",
                                }
                              ]
                            }
                          }
                          "disjointness": "node link srlg",

                        },
                        "connectivity-matrix": [
                          {
                            "id": 104,
                            "from": "1-0-1",
                            "to": "4-4-0"
                          },
                          {
                            "id": 107,
                            "from": "1-0-1",
                            "to": "7-7-0"
                          },
                          {
                            "id": 204,
                            "from": "2-0-2",
                            "to": "4-4-0"
                          },
                          {
                            "id": 308,
                            "from": "3-0-3",
                            "to": "8-8-0"
                          },
                          {
                            "id": 108,
                            "from": "1-0-1",
                            "to": "8-8-0"
                          },
                        ]
                      }
                    }
                  },
```

```
                    "termination-point": [

                      {
                        "tp-id": "1-0-1",
                        "te-tp-id": 10001,
                        "te": {
                          "interface-switching-capability": [
                            {
                              "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                              }
                          ]
                        }
                      },
                      {
                        "tp-id": "1-1-0",
                        "te-tp-id": 10100,
                        "te": {
                          "interface-switching-capability": [
                            {
                              "switching-capability": "switching-otn",
                                "encoding": "lsp-encoding-oduk"
                            }
                          ]
                        }
                      },
                      {
                        "tp-id": "2-0-2",
                        "te-tp-id": 20002,
                        "te": {
                          "interface-switching-capability": [
                            {
                              "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                            }
                          ]
                        }
                      },
                      {
                        "tp-id": "2-2-0",
                        "te-tp-id": 20200,
                        "te": {
                          "interface-switching-capability": [
                            {
                              "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                            }
                          ]
```

```
        }
      },
      {

        "tp-id": "3-0-3",
        "te-tp-id": 30003,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "3-3-0",
        "te-tp-id": 30300,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "4-0-4",
        "te-tp-id": 40004,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "4-4-0",
        "te-tp-id": 40400,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
```

```
          ]
        }
      },
      {
        "tp-id": "5-0-5",
        "te-tp-id": 50005,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "5-5-0",
        "te-tp-id": 50500,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "6-0-6",
        "te-tp-id": 60006,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "6-6-0",
        "te-tp-id": 60600,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
```

```
          ]
        }
      },
      {
        "tp-id": "7-0-7",
        "te-tp-id": 70007,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "7-7-0",
        "te-tp-id": 70700,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "8-0-8",
        "te-tp-id": 80008,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "8-8-0",
        "te-tp-id": 80800,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
```

```
                        ]
                      }
                    }
                  ]
                }
              ]
            },
            {
              "network-types": {
                "te-topology": {}
              },
              "network-id": "abstract2",
              "provider-id": 201,
              "client-id": 600,
              "te-topology-id": "te-topology:abstract2",
              "node": [
               {
                  "node-id": "D2",
                  "te-node-id": "2.0.1.2",
                  "te": {
                    "te-node-attributes": {
                      "domain-id" : 1,
                      "is-abstract": [null],
                      "connectivity-matrices": {
                        "is-allowed": true,
                        "underlay": {
                           "enabled": true
                        },
                        "path-constraints": {
                          "bandwidth-generic": {
                            "te-bandwidth": {
                              "generic": [
                                {
                                  "generic": "0x1p10"
                                }
                              ]
                            }
                          }
                        },
                        "optimizations": {
                           "objective-function": {
                               "objective-function-type":
                               "of-maximize-residual-bandwidth"
                           }
                        },
                        "connectivity-matrix": [
                          {
                            "id": 105,
```

```
                        "from": "1-0-1",
                        "to": "5-5-0",
                        "underlay": {
                           "enabled": true,
                           "primary-path": {
                               "network-ref": "absolute",
                               "path-element": [
                                 {

                                   "path-element-id": 1,
                                   "index": 1,
                                   "numbered-hop": {
                                     "address": "4.4.4.4",
                                     "hop-type": "STRICT"
                                    }
                                  },
                                  {
                                   "path-element-id": 2,
                                   "index": 2,
                                   "numbered-hop": {
                                     "address": "7.7.7.7",
                                     "hop-type": "STRICT"
                                   }
                                  }
                               ]
                           }
                        }
                      ]
                    }
                  }
                },
                "termination-point": [
                  {
                    "tp-id": "1-0-1",
                    "te-tp-id": 10001,
                    "te": {
                      "interface-switching-capability": [
                        {
                          "switching-capability": "switching-otn",
                          "encoding": "lsp-encoding-oduk"
                          }
                      ]
                    }
                  },
                  {
                    "tp-id": "1-1-0",
                    "te-tp-id": 10100,
```

```
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                  "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {

          "tp-id": "2-0-2",
          "te-tp-id": 20002,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "2-2-0",
          "te-tp-id": 20200,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "3-0-3",
          "te-tp-id": 30003,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "3-3-0",
```

```
          "te-tp-id": 30300,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "4-0-4",
          "te-tp-id": 40004,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "4-4-0",
          "te-tp-id": 40400,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "5-0-5",
          "te-tp-id": 50005,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "5-5-0",
```

```
          "te-tp-id": 50500,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "6-0-6",
          "te-tp-id": 60006,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "6-6-0",
          "te-tp-id": 60600,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "7-0-7",
          "te-tp-id": 70007,
          "te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-otn",
                "encoding": "lsp-encoding-oduk"
              }
            ]
          }
        },
        {
          "tp-id": "7-7-0",
```

```
                        "te-tp-id": 70700,
                        "te": {
                          "interface-switching-capability": [
                            {
                              "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                            }
                          ]
                        }
                      },
                      {
                        "tp-id": "8-0-8",
                        "te-tp-id": 80008,
                        "te": {

                          "interface-switching-capability": [
                            {
                              "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                            }
                          ]
                        }
                      },
                      {
                        "tp-id": "8-8-0",
                        "te-tp-id": 80800,
                        "te": {
                          "interface-switching-capability": [
                            {
                              "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                            }
                          ]
                        }
                      }
                    ]
                  }
                ]
              },
              {
                "network-types": {
                  "te-topology": {}
                },
                "network-id": "abstract3",
                "provider-id": 201,
                "client-id": 600,
                "te-topology-id": "te-topology:abstract3",
                "node": [
```

```
{
  "node-id": "D3",
  "te-node-id": "3.0.1.1",
  "te": {
    "te-node-attributes": {
      "domain-id" : 3,
      "is-abstract": [null],
      "connectivity-matrices": {
        "is-allowed": true,
        "path-constraints": {
          "bandwidth-generic": {
            "te-bandwidth": {
              "generic": [
                {
                  "generic": "0x1p10",
                }

              ]
            }
          }
        },
        "connectivity-matrix": [
          {
            "id": 107,
            "from": "1-0-1",
            "to": "7-7-0"
          },
          {
            "id": 308,
            "from": "3-0-3",
            "to": "8-8-0"
          },
        ]
      }
    }
  },
  "termination-point": [
    {
      "tp-id": "1-0-1",
      "te-tp-id": 10001,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
            }
        ]
      }
```

```
                    },
                    {
                      "tp-id": "1-1-0",
                      "te-tp-id": 10100,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "2-0-2",
                      "te-tp-id": 20002,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "2-2-0",
                      "te-tp-id": 20200,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "3-0-3",
                      "te-tp-id": 30003,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
```

```
                    },
                    {
                      "tp-id": "3-3-0",
                      "te-tp-id": 30300,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "4-0-4",
                      "te-tp-id": 40004,
                      "te": {
                        "interface-switching-capability": [
                          {

                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "4-4-0",
                      "te-tp-id": 40400,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "5-0-5",
                      "te-tp-id": 50005,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
```

```
        }
      },
      {
        "tp-id": "5-5-0",
        "te-tp-id": 50500,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "6-0-6",
        "te-tp-id": 60006,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "6-6-0",
        "te-tp-id": 60600,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
        }
      },
      {
        "tp-id": "7-0-7",
        "te-tp-id": 70007,
        "te": {
          "interface-switching-capability": [
            {
              "switching-capability": "switching-otn",
              "encoding": "lsp-encoding-oduk"
            }
          ]
```

```
                    }
                  },
                  {
                    "tp-id": "7-7-0",
                    "te-tp-id": 70700,
                    "te": {
                      "interface-switching-capability": [
                        {
                          "switching-capability": "switching-otn",
                          "encoding": "lsp-encoding-oduk"
                        }
                      ]
                    }
                  },
                  {
                    "tp-id": "8-0-8",
                    "te-tp-id": 80008,
                    "te": {
                      "interface-switching-capability": [
                        {
                          "switching-capability": "switching-otn",
                          "encoding": "lsp-encoding-oduk"
                        }
                      ]
                    }
                  },
                  {
                    "tp-id": "8-8-0",
                    "te-tp-id": 80800,
                    "te": {
                      "interface-switching-capability": [
                        {
                          "switching-capability": "switching-otn",
                          "encoding": "lsp-encoding-oduk"
                        }
                      ]
                    }
                  }
                ]
              }
            ]
          },
        ]
      }
    }
```

8. Security Considerations

The configuration, state, and action data defined in this document are designed to be accessed via a management protocol with a secure transport layer, 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 users to a preconfigured subset of all available NETCONF protocol operations and content.

The model presented in this document is used in the interface between the Customer Network Controller (CNC) and Multi-Domain Service Coordinator (MDSC), which is referred to as CNC-MDSC Interface (CMI). Therefore, many security risks such as malicious attack and rogue elements attempting to connect to various ACTN components. Furthermore, some ACTN components (e.g., MSDC) represent a single point of failure and threat vector and must also manage policy conflicts and eavesdropping of communication between different ACTN components.

A number of configuration data nodes defined in this document are writable/deletable (i.e., "config true") These data nodes may be considered sensitive or vulnerable in some network environments.

These are the subtrees and data nodes and their sensitivity/ vulnerability:

- access-point-list:
  - access-point-id
  - max-bandwidth
  - avl-bandwidth
- vn-ap:
  - vn-ap-id
  - vn
  - abstract-node
  - ltp

- vn-list
  - vn-id
  - vn-topology-id
  - abstract-node
- vn-member-id
  - src
  - src-vn-ap-id
  - dest
  - dest-vn-ap-id
  - connectivity-matrix-id

9. IANA Considerations

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```
---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-vn
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-vn
namespace:    urn:ietf:params:xml:ns:yang:ietf-vn
prefix:       vn
reference:    RFC XXXX (TBD)
---------------------------------------------------------------------
```

10. Acknowledgments

The authors would like to thank Xufeng Liu, Adrian Farrel, and Tom Petch for their helpful comments and valuable suggestions.

11. References

11.1. Normative References

[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.

[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>.

[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>.

[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>.

[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>.

11.2. Informative References

[I-D.ietf-ccamp-l1csm-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-l1csm-yang-12 (work in progress), September 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-03 (work in progress), July 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-04 (work in progress), July 2020.

[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>.

[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>.

Appendix A. Performance Constraints

At the time of creation of VN, it is natural to provide VN level constraints and optimization criteria. It should be noted that this YANG model rely on the TE-Topology Model [RFC8795] by using a reference to an abstract node to achieve this. Further, connectivity-matrix structure is used to assign the constraints and optimization criteria include delay, jitter etc. [RFC8776] define some of the metric-types already and future documents are meant to augment it.

Note that the VN compute allows inclusion of the constraints and the optimization criteria directly in the RPC to allow it to be used independently.

Appendix B. Contributors Addresses

Qin Wu
Huawei Technologies
Email: bill.wu@huawei.com

Peter Park
KT
Email: peter.park@kt.com

Haomian Zheng
Huawei Technologies
Email: zhenghaomian@huawei.com

Xian Zhang
Huawei Technologies
Email: zhang.xian@huawei.com

Sergio Belotti
Nokia
Email: sergio.belotti@nokia.com

Takuya Miyasaka
KDDI
Email: ta-miyasaka@kddi.com

Kenichi Ogaki
KDDI
Email: ke-oogaki@kddi.com

Authors' Addresses

Young Lee (editor)
Samsung Electronics

Email: younglee.tx@gmail.com

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Igor Bryskin
Individual

Email: i_bryskin@yahoo.com

Bin Yeong Yoon
ETRI

Email: byyun@etri.re.kr

TEAS Working Group
Internet-Draft
Obsoletes: 3272 (if approved)
Intended status: Informational
Expires: May 6, 2021

A. Farrel, Ed.
Old Dog Consulting

November 2, 2020

Overview and Principles of Internet Traffic Engineering
draft-ietf-teas-rfc3272bis-02

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.

Table of Contents

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 online and offline 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 by the Internet is the routing 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 the routing function, 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, scheduling

2. other functions that regulate traffic flow through the network or that arbitrate access to network resources between different packets or between different traffic streams.

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 task keeping 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.

1.2. Components of Traffic Engineering

As mentioned in Section 1.1, Internet traffic engineering provides performance optimization of operational 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 next hops and paths 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 [I-D.ietf-idr-rfc5575bis]. 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 polices, 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], and Segment Routing [RFC8402]. 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 southbound (i.e., controller to router) control protocol.

Resource management provides resource aware control and, in some cases, forwarding. Examples of resources are bandwidth, buffers, and queues, which in turn 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 to be used by a particular flow. This determination may be done on a very course or very fine level. Note that this consensus exists at the network control or controller level, not within the data plane. It may be purely composed 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 a static understanding of resource requirements, or using dynamic mechanisms to collect requirements (e.g., via [RFC3209]) and resource availability (e.g., via [RFC4203]), or any combination of the two.

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 for 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 implementing [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, in Section 7, an overview of the high level considerations pertaining to inter-domain traffic engineering will be provided. 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 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 conjunctively to accomplish traffic engineering objectives.

Traffic flow: A stream of packets between two end-points that can be characterized in a certain way. A micro-flow has a more specific definition A micro-flow is a stream of packets with the same 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 must 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 4.1.4), 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 streams belonging to the same service class (intra-class contention resolution) and traffic streams 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:

- 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 three significant characteristics:

- they provide real-time services
- they are mission critical
- 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 streams 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. RFC 2475 provides an architecture for Differentiated Services (DiffServ) and makes this requirement clear [RFC2475]. The RFC allows packets to 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 and QoS constraints.

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 large 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:

- 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 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 reasonable 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.

- 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 traffic engineering system itself.

- A set of quantitative and qualitative techniques and methodologies for abstracting, formulating, and solving traffic engineering problems.

- A set of administrative control parameters which may be manipulated through a Configuration Management (CM) system. The CM 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 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:

- 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 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:

- manually
- automatically, online using constraint-based routing processes implemented on label switching routers
- 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 equipement, 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 forcasts 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 medium time scale response schemes rely on a measurement system. The measurement system monitors changes in traffic distribution, traffic shifts, and network resource utilization. The measurement system then provides feedback to the online and/or offline traffic engineering mechanisms and tools which employ this feedback information to trigger certain control actions to occur within the network. The traffic engineering mechanisms and tools can be implemented in a distributed or centralized fashion, and may have a hierarchical or flat structure. The comparative merits of distributed and centralized control structures for networks are well known. A centralized scheme may have global visibility into the network state and may produce potentially 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 may 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. This 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 on the order of several round trip times. It includes router mechanisms such as passive and active buffer management. These mechanisms are used to control congestion and/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]. RED supports congestion avoidance by controlling the average queue size. 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. For a router that does not utilize explicit congestion notification (ECN) see e.g., [FLOY94], the marked packets can simply be dropped to signal the inception of congestion to end systems. On the other hand, if the router supports ECN, then 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], e.g., RED with In and Out (RIO) and Weighted RED. There is general consensus that RED provides congestion avoidance performance which is not worse than traditional Tail-Drop (TD) queue management (drop arriving packets only when the queue is full). Importantly, however, RED reduces the possibility of global synchronization and improves fairness among different TCP sessions. However, RED by itself can not 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 these schemes were proposed as theoretical frameworks and are typically not available in existing commercial products. Two such schemes are Longest Queue Drop (LQD) and Dynamic Soft Partitioning with Random Drop (RND) [SLDC98].

2. Congestion Management: Reactive versus Preventive Schemes

* Reactive: Reactive (recovery) congestion management policies react to existing congestion problems to improve it. All the policies described in the long and medium time scales above can be categorized as being reactive especially if the

policies are based on monitoring and identifying existing congestion problems, and on the initiation of relevant actions to ease a situation.

* Preventive: Preventive (predictive/avoidance) policies take proactive action to prevent congestion based on estimates and predictions of future potential congestion problems. Some of the policies described in the long and medium time scales fall into this category. They 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 congestion problems in the future. The schemes described in the short time scale (e.g., RED and its variations, ECN, LQD, and RND) are also used for congestion avoidance since dropping or marking packets before queues actually overflow would trigger corresponding TCP sources to slow down.

3. Congestion Management: Supply-Side versus Demand-Side Schemes

* Supply-side: 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. Another way to accomplish this is to minimize congestion by having a relatively balanced distribution of traffic over the network. For example, capacity planning should aim to provide a physical topology and associated link bandwidths that match estimated traffic workload and traffic distribution. This may be based on forecasting and subject to budgetary or other constraints. If actual traffic distribution does not match the topology derived from capacity panning, then the traffic can be mapped onto the existing topology using routing control mechanisms, using path oriented technologies (e.g., MPLS LSPs and optical channel trails) to modify the logical topology, or by using some other load redistribution mechanisms.

* Demand-side: 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 (such as RED and its variations, ECN, LQD, and RND) as well as policing and rate-shaping mechanisms attempt to regulate the offered load in various ways. Tariffs may also be applied as a demand side instrument. To date, however, tariffs have not been used as a means of demand-side congestion management within the Internet.

In summary, a variety of mechanisms can be used to address congestion problems in IP networks. These mechanisms may operate at multiple time-scales and at multiple traffic aggregation levels.

2.5. Implementation and Operational Context

The operational context of Internet traffic engineering is characterized by constant changes which 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 a traffic engineer, or more generally a traffic engineering system, must perform to optimize the performance of an operational network (see also [RFC2702], AWD2]). This process model may be enacted explicitly or implicitly, by an automaton and/or by a human.

The traffic engineering process model is iterative [AWD2]. The four phases of the process model described below are repeated continually.

- Define the relevant control policies that govern the operation of the network.

- A feedback mechanism involving the acquisition of measurement data from the operational network.

- Analyze the network state and to characterize traffic workload. Performance analysis may be proactive and/or reactive. Proactive performance analysis identifies potential problems that do not exist, but could manifest in the future. Reactive performance analysis identifies existing problems, determines their cause through diagnosis, and valuates alternative approaches to remedy the problem, if necessary.

- Performance optimization of the network. It involves a decision process which selects and implements a set of actions from a set of alternatives. Optimization actions may include the use of

appropriate techniques to either control the offered traffic or 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:

1. Measurement is crucial to the traffic engineering function. The operational state of a network can be conclusively determined only 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. Review of TE Techniques

This section briefly reviews different traffic engineering approaches proposed and implemented in telecommunications and computer networks. The discussion is not intended to be comprehensive. It is primarily intended to illuminate pre-existing perspectives and prior art

concerning traffic engineering in the Internet and in legacy telecommunications networks. A historic overview is provided in Appendix A.

4.1. Overview of IETF Projects Related to Traffic Engineering

This subsection reviews a number of IETF activities pertinent to Internet traffic engineering. These activities are primarily intended to evolve the IP architecture to support new service definitions which allow preferential or differentiated treatment to be accorded to certain types of traffic.

4.1.1. 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 setting, 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 4.1.5.

Unlike QoS routing (for example, see [RFC2386] and [MA]) 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.

4.1.2. Integrated Services

The IETF Integrated Services working group developed the integrated services (Intserv) model. This model 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 micro-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 next.

4.1.3. 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 for applications to communicate QoS requirements to the network and for the network to reserve relevant resources to satisfy the QoS requirements [RFC2205].

Under RSVP, the sender or source node sends a PATH message to the receiver with the same source and destination addresses as the traffic which the sender will generate. The PATH message contains: (1) a sender Tspec specifying the characteristics of the traffic, (2) a sender Template specifying the format of the traffic, and (3) an optional Adspec 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 micro-flows. These issues are described in [RFC2961].

Recently, RSVP has been modified and extended in several ways to mitigate the scaling problems. As a result, it is becoming 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, and to perform other signaling functions within the Internet. There are also a number of proposals to reduce the amount of refresh messages required to maintain established RSVP sessions [RFC2961].

A number of IETF working groups have been engaged in activities related to the RSVP protocol. These include the original RSVP working group, the MPLS working group, the CCAMP working group, the TEAS working group, the Resource Allocation Protocol working group, and the Policy Framework working group.

4.1.4. Differentiated Services

The goal of the Differentiated Services (Diffserv) effort within the IETF is 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 the Diffserv effort was to devise alternative mechanisms for service differentiation in the Internet that mitigate the scalability issues encountered with the Intserv model.

The IETF Diffserv working group has defined a Differentiated Services field in the IP header (DS field). The DS field consists of six bits of the part of the IP header formerly known as the TOS octet. The DS field is used to indicate the forwarding treatment that a packet should receive at a node [RFC2474]. The Diffserv working group has also standardized a number 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 receive Differentiated Services from its Internet Service Provider (ISP), it may be necessary for the user to have a Service Level Agreement (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.

It should be obvious from the previous discussion that the Diffserv model essentially 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].

4.1.5. 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].

4.1.6. Generalized MPLS

GMPLS extends MPLS control protocols to encompass time-division (e.g., SONET/SDH, PDH, G.709), wavelength (lambdas), 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 4.1.5).

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.

4.1.7. IP Performance Metrics

The IETF IP Performance Metrics (IPPM) working group has been developing 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 WG 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 WG are useful for specifying Service Level Agreements (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.

4.1.8. Flow Measurement

The IETF Real Time Flow Measurement (RTFM) working group has produced an architecture document defining 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 certain groups, accumulates certain 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 a user application, a host, a network, a group of 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 specification, meter control parameters, and sampling techniques. The instructions received by a meter reader from a manager include the address of the meter whose date is to be collected, the frequency of data collection, and the types of flows to be collected.

4.1.9. Endpoint Congestion Management

[RFC3124] is intended to provide a set of congestion control mechanisms that transport protocols can use. It is also intended to develop 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.

4.1.10. TE Extensions to the IGPs

TBD

4.1.11. 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 4.1.10.

The Border Gateway Protocol (BGP) 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 4.1.17) for use by the Path Computation Element (PCE, see Section 4.1.12), or may be used by Application-Layer Traffic Optimization (ALTO) servers (see Section 4.1.13).

4.1.12. 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 4.1.17) 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, such as the Application of the Path Computation Element Architecture [RFC6805], the Applicability of a Stateful Path Computation Element [RFC8051], Abstraction and Control of TE Networks (ACTN) Section 4.1.15, Centralized Network Control [RFC8283], and Software Defined Networking (SDN) Section 5.3.2.

4.1.13. Application-Layer Traffic Optimization

TBD

4.1.14. Segment Routing with MPLS encapsuation (SR-MPLS)

Segment Routing (SR) leverages the source routing and tunneling paradigms: The path packet takes is defined at the ingress and 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, label stack in MPLS case.

A segment can represent any instruction, topological or service-based, thanks to the MPLS architecture [RFC3031]. Labels cand be looked up in a global context (platform wide) as well as in some other context (see "context labels" in section 3 of [RFC5331]).

4.1.14.1. Base Segment Routing Identifier Types

Segments are identified by Segment Identifiers (SIDs). There are four types of SID that are relevant for traffic engineering.

Prefix SID: Uses SR Global Block (SRGB), must be unique within the routing domain SRGB, and is advertised by an IGP. The Prefix-SID can be configured as an absolute value or an index.

Node SID: A Node SID is a prefix SID with the 'N' (node) bit set, it is associated with a host prefix (/32 or /128) that identifies the node. More than 1 Node SID can be configured per node.

Adjacency SID: An Adjacency SID is locally significant (by default). It can be made globally significant through use of the 'L' flag. It identifies unidirectional adjacency. In most implementations Adjacency SIDs are automatically allocated for each adjacency. They are always encoded as an absolute (not indexed) value.

Binding SID: A Binding SID has two purposes

1. Mapping Server in ISIS

ISIS:The SID/Label Binding TLV is used to advertise prefixes to SID/Label mappings. This functionality is called the Segment Routing Mapping Server (SRMS). The behavior of the SRMS is defined in [RFC8661]

2. Cross-connect (label to FEC mapping)

This is fundamental for multi-domain/multi-layer operation. The Binding SID identifies a new (could be SR or hierarchical, at another OSI Layer) path available at the anchor point. Is always local to the originator (must not be at the top of the stack), must be looked up in the context of the nodal SID. It could be provisioned through Netconf/Restconf, PCEP, BGP, or the CLI.

4.1.15. 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 [RFC3945] and the Path Computation Element (PCE) [RFC4655]. 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 an 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 aspects like abstraction, virtualization and presentation. In particular it deals with:

- Abstraction of the underlying network resources and how they are provided to higher-layer applications and customers.
- Virtualization of underlying resources, whose selection criterion is the allocation of those resources for the customer, application, or service. The creation of a virtualized environment allowis 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 are 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 ACTN type of network virtualization provides customers and applications (tenants) to utilise 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.

4.1.16. Deterministic Networking

TBD

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

4.1.18. 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 programable 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.

4.2. Content Distribution

The Internet is dominated by client-server interactions, especially Web traffic (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. The exact parameters needed are not yet defined.

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.

5. Taxonomy of Traffic Engineering Systems

This section presents a short taxonomy of traffic engineering systems. A taxonomy of traffic engineering systems can be constructed based on traffic engineering styles and views as listed below:

- Time-dependent vs State-dependent vs Event-dependent
- Offline vs Online
- Centralized vs Distributed

- Local vs Global Information
- Prescriptive vs Descriptive
- Open Loop vs Closed Loop
- Tactical vs Strategic

These classification systems are described in greater detail in the following subsections of this document.

5.1. Time-Dependent Versus State-Dependent Versus Event Dependent

Traffic engineering methodologies can be classified as time-dependent, or 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.

In the time-dependent TE, historical information based on periodic variations in traffic, (such as time of day), is used to pre-program routing plans 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., diurnal). Time-dependent algorithms do not attempt to adapt to random 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]. Data mining 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 for packets based on the current state of the network. The current state of the network 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 such as utilization, packet delay, packet loss, etc. These parameters can be obtained in several ways. For example, each router may flood these

parameters periodically or by means of some kind of trigger to other 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].

Expeditious and accurate gathering and distribution of state information is critical for adaptive TE due to the dynamic nature of network conditions. State-dependent algorithms may be applied to increase network efficiency and resilience. Time-dependent algorithms are more suitable for predictable traffic variations. On the other hand, state-dependent algorithms are more suitable for adapting 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].

5.2. 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.

5.3. Centralized Versus Distributed

Centralized control has a central authority which determines routing plans and perhaps other TE control parameters on behalf of each router. The central authority collects the network-state information from all routers periodically and returns the routing information to the routers. The routing update cycle 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 routers 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 exceptional conditions. Examples of protocol extensions used to advertise network link state information are defined in [RFC5305], [RFC6119], [RFC7471], [RFC7810], and [RFC8571].

5.3.1. Hybrid Systems

TBD

5.3.2. Considerations for Software Defined Networking

TBD

5.4. Local Versus Global

Traffic engineering algorithms may require local or global network-state information.

Local information pertains to the state of a portion of the domain. Examples include the bandwidth and packet loss rate of a particular path. Local state information may be sufficient for certain instances of distributed-controlled TEs.

Global information pertains to the state of the entire domain undergoing traffic engineering. 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.

5.5. 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.

5.5.1. Intent-Based Networking

TBD

5.6. 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.

5.7. Tactical vs 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.

6. Recommendations for Internet Traffic Engineering

This section describes high-level recommendations for traffic engineering in the Internet. These recommendations are presented 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.

Functional recommendations for Internet traffic engineering describe the functions that a traffic engineering system should perform. These functions are needed to realize traffic engineering objectives by addressing traffic engineering problems.

Non-functional recommendations for Internet traffic engineering 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 include: usability, automation, scalability, stability, visibility, simplicity, efficiency, reliability, correctness, maintainability, extensibility, interoperability, and security. 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 (or components thereof) to tailor it to a specific operational context.

In the following paragraphs, some of the aspects of the non-functional recommendations for Internet traffic engineering are summarized.

Usability: Usability is a human factor aspect of traffic engineering systems. Usability 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 traffic engineering system should automate as many traffic engineering functions as possible to minimize the amount of human effort needed to control and analyze operational networks. Automation is particularly imperative 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 traffic engineering system.

Scalability: Contemporary public networks are growing very fast with respect to network size and traffic volume. Therefore, a TE system should be scalable to remain applicable as the network evolves. 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 much network resources when collecting and distributing state information or when exerting control.

Stability: Stability is a very important consideration in traffic engineering systems that respond to changes in the state of the network. State-dependent traffic engineering methodologies typically mandate a tradeoff between responsiveness and stability. It is strongly recommended that when tradeoffs are warranted between responsiveness and stability, that the tradeoff should be made in favor of stability (especially in public IP backbone networks).

Flexibility: A TE system should be flexible to allow for changes in optimization policy. In particular, a TE system should provide sufficient configuration options so that a network administrator can tailor the TE 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: As part of the TE system, mechanisms should exist 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. Other examples of status information which should be observed include existing functional routing information (additionally, in the context of MPLS existing LSP routes), etc.

Simplicity: Generally, a TE system should be as simple as possible. More importantly, the TE system should be relatively easy to use (i.e., clean, convenient, and intuitive user interfaces). Simplicity in user interface does not necessarily imply that the TE system will

use naive algorithms. When complex algorithms and internal structures are used, such complexities should be hidden as much as possible from the network administrator through the user interface.

Interoperability: Whenever feasible, traffic engineering 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 traffic engineering systems. Such traffic engineering systems typically exert control over certain functional aspects of the network to achieve the desired performance objectives. Therefore, adequate measures must be taken to safeguard the integrity of the traffic engineering system. Adequate measures must also be taken to protect the network from vulnerabilities that originate from security breaches and other impairments within the traffic engineering system.

The remainder of this section will 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 routing system is one that takes traffic characteristics and network constraints into account during route selection while maintaining stability.

Traditional shortest path first (SPF) interior gateway protocols are based on shortest path algorithms and have limited control capabilities for traffic engineering [RFC2702], [AWD2]. These limitations include:

1. The well known issues with pure SPF protocols, which do not take network constraints and traffic characteristics into account during route selection. For example, since IGPs always use the shortest paths (based on administratively assigned link metrics) to forward traffic, load sharing cannot be accomplished among paths of different costs. Using shortest paths to forward traffic conserves network resources, but may cause the following problems: 1) If traffic from a source to a destination exceeds the capacity of a link along the shortest path, the link (hence the shortest path) becomes congested while a longer path between these two nodes may be under-utilized; 2) 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.

3. Modifying IGP metrics to control traffic routing tends to have network-wide effect. Consequently, undesirable and unanticipated traffic shifts can be triggered as a result. Recent 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 have been described elsewhere and are summarized below.

Constraint-based routing is desirable to evolve the routing architecture of IP networks, especially public IP backbones with complex topologies [RFC2702]. Constraint-based routing computes routes to fulfill requirements subject to constraints. 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 subject to network and administrative policy constraints. Routes computed through 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 redistribute traffic onto the infrastructure (even for best effort traffic). For example, if the bandwidth requirements for path selection and reservable bandwidth attributes of network links are appropriately defined and configured, then congestion problems caused by uneven traffic distribution may be avoided or reduced. In this way, the performance and efficiency of the network can be improved.

A number of enhancements are needed to conventional link state IGPs, such as OSPF and IS-IS, to allow them to distribute additional state information required for constraint-based routing. These extensions to OSPF were described in [RFC3630] and to IS-IS in [RFC5305]. Essentially, these enhancements require the propagation of additional information in link state advertisements. Specifically, in addition to normal link-state information, an enhanced IGP is required to propagate topology state information needed for constraint-based routing. Some of the additional topology state information include link attributes such as reservable bandwidth and link resource class attribute (an administratively specified property of the link). The resource class attribute concept was 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 tradeoff is typically required between the timeliness of the information flooded and the flooding frequency to avoid excessive consumption of link bandwidth and computational resources, and more importantly, to avoid instability.

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].

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 from a source to a destination away from its original path to another path (without affecting other traffic paths) allows traffic to be moved from resource-poor network segments to resource-rich segments. Path oriented technologies such as MPLS 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 pertains to 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 selected 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 5.1.

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. This requirement was 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 micro-flow) at the destination node of the parallel paths.

As a general rule, 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 rely on short time scale congestion control mechanisms (such as queue management, scheduling, etc.) to mitigate congestion. Thus, mechanisms that perform the traffic mapping functions should complement existing congestion control mechanisms. In an operational network, it is generally desirable to map the traffic onto the infrastructure such that intra-class and inter-class resource contention are minimized.

When traffic mapping techniques that depend on dynamic state feedback (e.g., 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. Mechanisms should be provided to measure and collect statistics from the network to support the traffic engineering 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 timescale traffic statistics are very useful for traffic engineering. Long-term time scale traffic statistics may capture or reflect periodicity in network workload (such as hourly, daily, and weekly variations in traffic profiles) as well as traffic trends. Aspects of the monitored traffic statistics may also depict class of service characteristics for a network supporting multiple classes of service. Analysis of the long-term traffic statistics may yield secondary statistics 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 of 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.

Measured 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 techniques, FTP, IGP link state advertisements, etc.

6.5. 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 has become an issue of great concern within the Internet community due to the increasing demands 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 Generalized Multi-Protocol Label Switching (GMPLS) Recovery [RFC4872], and GMPLS Segment Recovery [RFC4873].

Failure protection and restoration capabilities have become available from multiple layers as network technologies have continued to improve. At the bottom of the layered stack, optical networks are now capable of providing dynamic ring and mesh restoration functionality at the wavelength level as well as traditional protection functionality. 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 ATM (generally with slower mean restoration times). Rerouting is traditionally 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. Some new developments in the MPLS context make it possible to achieve recovery at the IP layer prior to convergence [RFC3469].

To support advanced survivability requirements, path-oriented technologies such a MPLS can be used to enhance the survivability of IP networks in a potentially cost effective manner. The advantages of path oriented technologies such as MPLS for IP restoration becomes even more evident when class based protection and restoration capabilities are required.

Recently, a common suite of control plane protocols has been proposed for both MPLS and optical transport networks under the acronym Multi-protocol Lambda Switching [AWD1]. This new paradigm of Multi-protocol Lambda Switching will support even more sophisticated mesh restoration capabilities at the optical layer for the emerging IP over WDM network architectures.

Another important aspect regarding multi-layer survivability is that technologies at different layers provide protection and restoration capabilities at different temporal 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.

The impact of service outages varies significantly for different service classes depending upon the effective duration of the outage. The duration of an outage 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).

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 whenever feasible and appropriate to provide network survivability in a flexible and cost effective manner. Minimization of function duplication across 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. A temporal order of restoration trigger timing at different layers is another way to coordinate multi-layer protection/restoration.

- Spare capacity at higher layers is often regarded as working traffic at lower layers. Placing protection/restoration functions in many layers may increase redundancy and robustness, but it should not result in significant and avoidable inefficiencies in network resource utilization.

- It is generally desirable to have protection and restoration schemes that are bandwidth efficient.

- Failure notification throughout the network should be timely and reliable.

- Alarms and other fault monitoring and reporting capabilities should be provided at appropriate layers.

6.5.1. Survivability in MPLS Based Networks

MPLS is an important emerging technology that enhances IP networks in terms of features, capabilities, and services. Because MPLS is path-oriented, it can potentially provide faster and more predictable

protection and restoration capabilities than conventional hop by hop routed IP systems. This subsection describes some of the basic aspects and recommendations for MPLS networks regarding protection and restoration. See [RFC3469] for a more comprehensive discussion on MPLS based recovery.

Protection types for MPLS networks can be categorized as link protection, node protection, path protection, and segment protection.

- Link Protection: The objective for link protection is to protect an LSP from a given link failure. Under link protection, the path of the protection or backup LSP (the secondary LSP) is disjoint from the path of the working or operational LSP at the particular link over which protection is required. When the protected link fails, traffic on the working LSP is switched over to the protection LSP at the head-end of the failed link. This is a local repair method which can be fast. It might be more appropriate in situations where some network elements along a given path are less reliable than others.

- Node Protection: The objective of LSP node protection is to protect an LSP from a given node failure. Under node protection, the path of the protection LSP is disjoint from the path of the working LSP at the particular node to be protected. The secondary path is also disjoint from the primary path at all links associated with the node to be protected. When the node fails, traffic on the working LSP is switched over to the protection LSP at the upstream LSR directly connected to the failed node.

- Path Protection: The goal of LSP path protection is to protect an LSP from failure at any point 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 that might occur at the ingress and egress LSRs, or for correlated failures that might impact both working and backup paths simultaneously. Additionally, since the path selection is end-to-end, path protection might be more efficient in terms of resource usage than link or node protection. However, path protection may be slower than link and node protection in general.

- Segment Protection: An MPLS domain may be partitioned into multiple protection domains whereby a failure in a protection domain is rectified within that 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 path protection because recovery generally occurs closer to the fault.

6.5.2. Protection Option

Another issue to consider is the concept of protection options. The protection option uses the notation m:n protection, where m is the number of protection LSPs used to protect n working LSPs. Feasible protection options follow.

- 1:1: one working LSP is protected/restored by one protection LSP.
- 1:n: one protection LSP is used to protect/restore n working LSPs.
- n:1: one working LSP is protected/restored by n protection LSPs, possibly with configurable load splitting ratio. When more than one protection LSP is used, it may be desirable to share the traffic across the protection LSPs when the working LSP fails to satisfy the bandwidth requirement of the traffic trunk associated with the working LSP. This may be especially useful when it is not feasible to find one path that can satisfy the bandwidth requirement of the primary LSP.
- 1+1: traffic is sent concurrently on both the working LSP and the protection LSP. In this case, the egress LSR selects one of the two LSPs based on a local traffic integrity decision process, which compares the traffic received from both the working and the protection LSP and identifies discrepancies. It is unlikely that this option would be used extensively in IP networks due to its resource utilization inefficiency. However, if bandwidth becomes plentiful and cheap, then this option might become quite viable and attractive in IP networks.

6.6. Traffic Engineering in Diffserv Environments

This section provides an overview of the traffic engineering features and recommendations that are specifically pertinent to Differentiated Services (Diffserv) [RFC2475] capable IP networks.

Increasing requirements to support multiple classes of traffic, such as best effort and mission critical data, in the Internet calls for IP networks to differentiate traffic according to some criteria, and to accord preferential treatment to certain types of traffic. Large numbers of flows can be aggregated into a few behavior aggregates based on some criteria in terms of common performance requirements in terms of packet loss ratio, delay, and jitter; or in terms of common fields within the IP packet headers.

As Diffserv evolves and becomes deployed in operational networks, traffic engineering will be critical to ensuring that SLAs defined within a given Diffserv service model are met. Classes of service (CoS) can be supported in a Diffserv environment by concatenating per-hop behaviors (PHBs) along the routing path, using service provisioning mechanisms, and by appropriately configuring edge functionality such as traffic classification, marking, policing, and shaping. PHB is the forwarding behavior that a packet receives at a DS node (a Diffserv-compliant node). This is accomplished by means of buffer management and packet scheduling mechanisms. In this context, packets belonging to a class are those that are members of a corresponding ordering aggregate.

Traffic engineering can be used as a compliment to Diffserv mechanisms to improve utilization of network resources, but not as a necessary element in general. When traffic engineering is used, it 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 aggregate 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 certain 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:

- traffic engineering mechanisms on a per service class basis that enforce the desired 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 by, for example, 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 vary significantly from router to router, it may not suffice to rely exclusively on conventional IGP routing protocols or on traffic engineering mechanisms that are insensitive 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 certain per-class parameters to be distributed. Note that it is common to have some classes share some aggregate constraint (e.g., maximum bandwidth requirement) without enforcing the constraint on each individual class. These classes then can be grouped into a class-type and per-class-type parameters can be distributed instead to improve scalability. It 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 individual class in the class-type. Note that it is still possible, nevertheless, 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.

An example of the class-type can be a low-loss class-type that includes both AF1-based and AF2-based Ordering Aggregates. With such a class-type, one may implement some priority policy which assigns higher preemption priority to AF1-based traffic trunks over AF2-based ones, vice versa, or the same priority.

See [RFC4124] for detailed requirements on Diffserv-aware traffic engineering.

6.7. Network Controllability

Off-line (and on-line) traffic engineering considerations would be of limited utility if the network could not be controlled effectively to implement the results of TE decisions and to achieve desired network performance objectives. Capacity augmentation is a coarse grained solution to traffic engineering issues. However, it is simple and may be advantageous if bandwidth is abundant and cheap or if the

current or expected network workload demands it. However, bandwidth is not always abundant and cheap, and the workload may not always demand additional capacity. Adjustments of administrative weights and other parameters associated with routing protocols provide finer grained control, but is difficult to use and imprecise because of the routing interactions that occur across the network. In certain network contexts, more flexible, finer grained approaches which provide more precise control over the mapping of traffic to routes and over the selection and placement of routes may be appropriate and useful.

Control mechanisms can be manual (e.g., administrative configuration), partially-automated (e.g., scripts) or fully-automated (e.g., policy based management systems). Automated mechanisms are particularly required in large scale networks. Multi-vendor interoperability can be facilitated by developing and deploying standardized management systems (e.g., standard MIBs) and policies (PIBs) to support the control functions required to address traffic engineering objectives such as load distribution and protection/restoration.

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 traffic engineering is concerned with the performance optimization for traffic that originates in one administrative domain and terminates in a different one.

Traffic exchange between autonomous systems in the Internet occurs through exterior gateway protocols. Currently, BGP [RFC4271] is the standard exterior gateway protocol for the Internet. BGP provides a number of attributes and capabilities (e.g., route filtering) that can be used for inter-domain traffic engineering. More specifically, BGP permits the control of routing information and traffic exchange between Autonomous Systems (ASes) in the Internet. BGP incorporates a sequential decision process which calculates the degree of preference for various routes to a given destination network. There are two fundamental aspects to inter-domain traffic engineering using BGP:

- Route Redistribution: controlling the import and export of routes between AS's, 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. Best path selection is performed by the BGP decision process based on a sequential procedure, taking a number of different considerations into account. Ultimately, best path selection under BGP boils down to selecting preferred exit points out of an AS towards specific destination networks. The BGP path selection process can be influenced by manipulating the attributes associated with the BGP decision process. These attributes include: NEXT-HOP, WEIGHT (Cisco proprietary which is also implemented by some other vendors), 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. In particular, Route-maps 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 looking at possible strategies for inter-domain TE with BGP, it must be noted that the outbound traffic exit point is controllable, whereas the interconnection point where inbound traffic is received from an EBGP peer typically is not, unless a special arrangement is made with the peer sending the traffic. Therefore, it is up to each individual network to implement sound TE strategies that deal with the efficient delivery of outbound traffic from one's customers to one's peering points. The vast majority of TE policy is based upon a "closest exit" strategy, which offloads interdomain traffic at the nearest outbound peer point towards the destination autonomous system. Most methods of manipulating the point at which inbound traffic enters a network from an EBGP peer (inconsistent route announcements between peering points, AS pre-pending, and sending MEDs) 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. A systematic approach for inter-domain traffic engineering is yet to be devised.

Inter-domain TE is inherently more difficult than intra-domain TE under the current Internet architecture. The reasons for this are both technical and administrative. Technically, while topology and link state information are helpful for mapping traffic more effectively, BGP does not propagate such information across domain

boundaries for stability and scalability reasons. Administratively, there are differences in operating costs and network capacities between domains. Generally, what may be considered a good solution in one domain may not necessarily be a good solution in another domain. Moreover, it would generally be considered inadvisable for one domain to permit another domain to influence the routing and management of traffic in its network.

MPLS TE-tunnels (explicit LSPs) can potentially add a degree of flexibility in the selection of exit points for inter-domain routing. The concept of relative and absolute metrics can be applied to this purpose. The idea is that 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 to 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. If a peer accepts and processes MEDs, then a similar MPLS TE-tunnel based scheme can be applied to cause certain entrance points to be preferred by setting MED to be an IGP cost, which has been modified by the tunnel metric.

Similar 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 autonomous system to another.

Generally, redistribution of inter-domain traffic requires coordination between peering partners. An export policy in one domain that results in load redistribution across peer points with another domain can significantly affect the local traffic matrix inside the domain of the peering partner. This, in turn, will affect the intra-domain TE due to changes in the spatial distribution of traffic. Therefore, it is mutually beneficial for peering partners to coordinate with each other before attempting any policy changes that may result in significant shifts in inter-domain traffic. In certain contexts, this coordination can be quite challenging due to technical and non- technical reasons.

It is a matter of speculation as to whether MPLS, or similar technologies, can be extended to allow selection of constrained paths across domain boundaries.

8. Overview of Contemporary TE Practices in Operational IP Networks

This section provides an overview of some contemporary traffic engineering practices in IP networks. The focus is primarily on the aspects that pertain to the 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.

Currently, service providers apply many of the traffic engineering mechanisms discussed 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 for medium timescales, the overlay model also for medium timescales, and traffic management mechanisms for short timescale.

When a service provider plans to build an IP network, or expand the capacity of an existing network, effective capacity planning should be an important component of the process. Such plans may take the following aspects into account: location of new nodes if any, 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 then analyzed and used to trigger various traffic engineering mechanisms. Tools that perform what-if analysis can also be used to assist the TE process by allowing various scenarios to be reviewed before a new set of configurations are implemented in the operational network.

Traditionally, intra-domain real-time TE with IGP is done by increasing the OSPF or IS-IS metric of a congested link until enough traffic has been diverted from that link. This approach has some limitations as discussed in Section 6.2. Recently, some new intra-domain TE approaches/tools have been proposed [RR94] [FT00] [FT01] [WANG]. Such approaches/tools take traffic matrix, network topology, and network performance objectives as input, and produce some link metrics and possibly some unequal load-sharing ratios to be set at the head-end routers of some ECMPs as output. These new progresses open new possibility for intra-domain TE with IGP to be done in a more systematic way.

The overlay model (IP over ATM, or IP over Frame Relay) is another approach which was commonly used [AWD2], but has been replaced by MPLS and router hardware technology.

Deployment of MPLS for traffic engineering applications has commenced in some service provider networks. One operational scenario is to deploy MPLS in conjunction with an IGP (IS-IS-TE or OSPF-TE) that supports the traffic engineering extensions, in conjunction with

constraint-based routing for explicit route computations, and a signaling protocol (e.g., RSVP-TE) for LSP instantiation.

In contemporary MPLS traffic engineering contexts, network administrators specify and configure link attributes and resource constraints such as maximum reservable bandwidth and resource class attributes for links (interfaces) within the MPLS domain. A link state protocol that supports TE extensions (IS-IS-TE or OSPF-TE) is used to propagate information about network topology and link attribute to all routers in the routing area. Network administrators also specify all the LSPs that are to originate each router. For each LSP, the network administrator specifies the destination node and the attributes of the LSP which indicate the requirements that to be satisfied during the path selection process. Each router then uses a local constraint-based routing process to compute explicit paths for all LSPs originating from it. Subsequently, a signaling protocol is used to instantiate the LSPs. By assigning proper bandwidth values to links and LSPs, congestion caused by uneven traffic distribution can generally be avoided or mitigated.

The bandwidth attributes of LSPs used for traffic engineering can be updated periodically. The basic concept is that the bandwidth assigned to an LSP should relate in some manner to the bandwidth requirements of traffic that actually flows through the LSP. The traffic attribute of an LSP can be modified to accommodate traffic growth and persistent traffic shifts. 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.

In an MPLS domain, a traffic matrix 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. Current practice suggests that deploying an MPLS network consisting of hundreds of routers and thousands of LSPs is feasible. In summary, recent deployment experience suggests that MPLS approach is very effective for traffic engineering in IP networks [XIAO].

As mentioned previously in Section 7, one usually has no direct control over the distribution of inbound traffic. Therefore, the main goal of contemporary 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 usually begins with the placement of multiple peering interconnection points in locations that have high peer density, are in close proximity to originating/terminating traffic locations on one's own network, and are lowest in cost. There are generally several locations in each region of the world where the vast majority of major networks congregate and interconnect. Some location-decision problems that arise in association with inter-domain routing are discussed in [AWD5].

Once the locations of the interconnects are determined, and circuits are implemented, one decides how best to handle the routes heard from the peer, as well as how to propagate the peers' routes within one's own network. One way to engineer outbound traffic flows on a network with many EBGP peers is to create a hierarchy of peers. Generally, the Local Preferences of all peers are set to the same value so that the shortest AS paths will be chosen to forward traffic. Then, by over-writing the inbound MED metric (Multi-exit-discriminator metric, also referred to as "BGP metric". Both terms are used interchangeably in this document) with BGP metrics to routes received at different peers, the hierarchy can be formed. For example, all Local Preferences can be set to 200, preferred private peers can be assigned a BGP metric of 50, the rest of the private peers can be assigned a BGP metric of 100, and public peers can be assigned a BGP metric of 600. "Preferred" peers might be defined as those peers with whom the most available capacity exists, whose customer base is larger in comparison to other peers, whose interconnection costs are the lowest, and with whom upgrading existing capacity is the easiest. In a network with low utilization at the edge, this works well. The same concept could be applied to a network with higher edge utilization by creating more levels of BGP metrics between peers, allowing for more granularity in selecting the exit points for traffic bound for a dual homed customer on a peer's network.

By only replacing inbound MED metrics with BGP metrics, only equal AS-Path length routes' exit points are being changed. (The BGP decision considers Local Preference first, then AS-Path length, and then BGP metric). For example, assume a network has two possible egress points, peer A and peer B. Each peer has 40% of the Internet's routes exclusively on its network, while the remaining 20% of the Internet's routes are from customers who dual home between A and B. Assume that both peers have a Local Preference of 200 and a BGP metric of 100. If the link to peer A is congested, increasing its BGP metric while leaving the Local Preference at 200 will ensure that the 20% of total routes belonging to dual homed customers will prefer peer B as the exit point. The previous example would be used

in a situation where all exit points to a given peer were close to congestion levels, and traffic needed to be shifted away from that peer entirely.

When there are multiple exit points to 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 circuit. This can be achieved by using passive IGP-metrics, AS-path filtering, or prefix filtering.

Occasionally, more drastic changes are needed, for example, in dealing 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 Local Preference to that peer can be reduced below the level of other peers. This effectively reduces the amount of traffic sent to that peer to only originating traffic (assuming no transit providers are involved). 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.

Although it is not much of an issue in regional networks, the propagation of a peer's routes back through the network must be considered when a network is peering on a global scale. Sometimes, business considerations can influence the choice of BGP policies in a given context. For example, it may be imprudent, from a business perspective, to operate a global network and provide full access to the global customer base to a small network in a particular country. However, for the purpose of providing one's own customers with quality service in a particular region, good connectivity to that in-country network may still be necessary. This can be achieved by assigning a set of communities at the edge of the network, which have a known behavior when routes tagged with those communities are propagating back through the core. Routes heard from local peers will be prevented from propagating back to the global network, whereas routes learned from larger peers may be allowed to propagate freely throughout the entire global network. By implementing a flexible community strategy, the benefits of using a single global AS Number (ASN) can be realized, while the benefits of operating regional networks can also be taken advantage of. An alternative to doing this is to use different ASNs in different regions, with the consequence that the AS path length for routes announced by that service provider will increase.

9. Conclusion

This document described principles for traffic engineering in the Internet. It presented an overview of some of the basic issues surrounding traffic engineering in IP networks. The context of TE

was described, a TE process models and a taxonomy of TE styles were presented. A brief historical review of pertinent developments related to traffic engineering was provided. A survey of contemporary TE techniques in operational networks was presented. Additionally, the document specified a set of generic requirements, recommendations, and options for Internet traffic engineering.

10. Security Considerations

This document does not introduce new security issues.

11. IANA Considerations

This draft makes no requests for IANA action.

12. 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 orginal authors should be considered as Contributors to this work. They were:

Daniel O. Awduche
Movaz Networks
7926 Jones Branch Drive, Suite 615
McLean, VA 22102

Phone: 703-298-5291
EMail: awduche@movaz.com

Angela Chiu
Celion Networks
1 Sheila Dr., Suite 2
Tinton Falls, NJ 07724

Phone: 732-747-9987
EMail: angela.chiu@celion.com

Anwar Elwalid
Lucent Technologies
Murray Hill, NJ 07974

Phone: 908 582-7589
EMail: anwar@lucent.com

Indra Widjaja
Bell Labs, Lucent Technologies
600 Mountain Avenue
Murray Hill, NJ 07974

Phone: 908 582-0435
EMail: iwidjaja@research.bell-labs.com

XiPeng Xiao
Redback Networks
300 Holger Way
San Jose, CA 95134

Phone: 408-750-5217
EMail: xipeng@redback.com

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 vs 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 authors of this document would also like to thank Dhurv Dhody for review comments.

13. 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

14. 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.

[AWD1] Awduche, D. and Y. Rekhter, "Multiprocotol Lambda Switching - Combining MPLS Traffic Engineering Control with Optical Crossconnects", Article IEEE Communications Magazine, March 2001.

[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-idr-rfc5575bis] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", draft-ietf-idr-rfc5575bis-27 (work in progress), October 2020.

[I-D.ietf-teas-yang-te-topo] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.

[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.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, DOI 10.17487/RFC7810, May 2016, <https://www.rfc-editor.org/info/rfc7810>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC8661] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., and S. Litkowski, "Segment Routing MPLS Interworking with LDP", RFC 8661, DOI 10.17487/RFC8661, December 2019, <https://www.rfc-editor.org/info/rfc8661>.

[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 incepted 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:

- 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 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 Sonet/SDH, OTN, or 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., PCR, SCR, and MBS 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

This section is a place-holder. It is expected that once work on this document is nearly complete, this section will be updated to provide an overview of the structural and substantive changed from RFC 3272.

Author's Address

Adrian Farrel (editor)
Old Dog Consulting

Email: adrian@olddog.co.uk

TEAS Working Group
Internet-Draft
Obsoletes: 3272 (if approved)
Intended status: Informational
Expires: May 31, 2021

A. Farrel, Ed.
Old Dog Consulting

November 27, 2020

Overview and Principles of Internet Traffic Engineering
draft-ietf-teas-rfc3272bis-08

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.

Table of Contents

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 online and offline 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 by the Internet is the routing 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 the routing function, 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, scheduling

2. other functions that regulate traffic flow through the network or that arbitrate access to network resources between different packets or between different traffic streams.

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 task keeping 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.

1.2. Components of Traffic Engineering

As mentioned in Section 1.1, Internet traffic engineering provides performance optimization of operational 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 next hops and paths 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 [I-D.ietf-idr-rfc5575bis]. 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 polices, 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], and Segment Routing [RFC8402]. 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 southbound (i.e., controller to router) 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 course 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, in Section 7, an overview of the high level considerations pertaining to inter-domain traffic engineering will be provided. 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 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 micro-flow has a more specific definition A micro-flow is a stream of packets with the same 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 must 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 4.1.4), 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 streams belonging to the same service class (intra-class contention resolution) and traffic streams 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:

- 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 three significant characteristics:

- they provide real-time services
- they are mission critical
- 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 streams 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. RFC 2475 provides an architecture for Differentiated Services (Diffserv) and makes this requirement clear [RFC2475]. The RFC allows packets to 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 and QoS constraints.

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 large 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:

- 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 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 reasonable 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.

- 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 traffic engineering system itself.

- A set of quantitative and qualitative techniques and methodologies for abstracting, formulating, and solving traffic engineering problems.

- A set of administrative control parameters which may be manipulated through a Configuration Management (CM) system. The CM 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 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:

- 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 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:

- manually
- automatically, online using constraint-based routing processes implemented on label switching routers
- 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. Congestion Management: Reactive Versus Preventive Schemes

* Reactive: 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: Preventive (predictive/avoidance) policies take proactive action to prevent congestion based on estimates and predictions of future congestion problems. 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. Congestion Management: Supply-Side Versus Demand-Side Schemes

* Supply-side: 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 panning, 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: 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.

- 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 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. 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.

4.1. Overview of IETF Projects Related to Traffic Engineering

This subsection reviews a number of IETF activities pertinent to Internet traffic engineering.

4.1.1. 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 4.1.6.

Unlike QoS routing (for example, see [RFC2386] and [MA]) 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.

4.1.2. 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 micro-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 4.1.3.

4.1.3. 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 4.1.2) 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 micro-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 4.1.6), 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.

4.1.4. 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].

4.1.5. QUIC

TBD

4.1.6. 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].

4.1.7. Generalized MPLS (GMPLS)

GMPLS extends MPLS control protocols to encompass time-division (e.g., SONET/SDH, PDH, G.709), wavelength (lambdas), 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 4.1.6).

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.

4.1.8. 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.

4.1.9. 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:

- understanding the behavior of existing networks
- 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 date is to be collected, the frequency of data collection, and the types of flows to be collected.

4.1.10. Endpoint Congestion Management

[RFC3124] provides a set of congestion control mechanisms for the use of transport protocols. It is 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.

4.1.11. 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 [RFC2370] 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 [RFC7810] for IS-IS and in [RFC7471] for OSPFv2 ([RFC5329] for OSPFv3).

4.1.12. 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 4.1.11.

The Border Gateway Protocol (BGP) 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 4.1.19) for use by the Path Computation Element (PCE, see Section 4.1.13), or may be used by Application-Layer Traffic Optimization (ALTO) servers (see Section 4.1.14).

4.1.13. 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 4.1.19) 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, such as the Application of the Path Computation Element Architecture [RFC6805], the Applicability of a Stateful Path Computation Element ([RFC8051]), Abstraction and Control of TE Networks (ACTN) (Section 4.1.16), Centralized Network Control [RFC8283], and Software Defined Networking (SDN) (Section 5.3.2).

4.1.14. 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 [RFC7159] 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 4.1.12) or IGP-TE (see Section 4.1.11), 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.

4.1.15. Segment Routing with MPLS Encapsulation (SR-MPLS)

Segment Routing (SR) 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]).

4.1.15.1. Base Segment Routing Identifier Types

Segments are identified by Segment Identifiers (SIDs). There are four types of SID that are relevant for traffic engineering.

Prefix SID: Uses the SR Global Block (SRGB), must be unique within the routing domain SRGB, and is advertised by an IGP. The Prefix-SID can be configured as an absolute value or an index.

Node SID: A Prefix SID with the 'N' (node) bit set. It is associated with a host prefix (/32 or /128) that identifies the node. More than 1 Node SID can be configured per node.

Adjacency SID: Locally significant by default, an Adjacency SID can be made globally significant through use of the 'L' flag. It identifies a unidirectional adjacency. In most implementations Adjacency SIDs are automatically allocated for each adjacency. They are always encoded as an absolute (not indexed) value.

Binding SID: A Binding SID has two purposes:

1. Mapping Server in ISIS

The SID/Label Binding TLV is used to advertise the mappings of prefixes to SIDs/Labels. This functionality is called the Segment Routing Mapping Server (SRMS). The behavior of the SRMS is defined in [RFC8661]

2. Cross-connect (label to FEC mapping)

This is fundamental for multi-domain/multi-layer operation. The Binding SID identifies a new path available at the anchor point. It is always local to the originator, must not be present at the top of the stack, and must be looked up in the context of the Node SID. It could be provisioned through Netconf/Restconf, PCEP, BGP, or the CLI.

4.1.15.2. Segment Routing Policy

TBD : Boris Hassanov

4.1.16. 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 4.1.6 and Section 4.1.7) and the Path Computation Element (PCE) (Section 4.1.13). 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:

- 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.

4.1.17. 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.nsdt-teas-ietf-network-slice-definition]. The resources are used to satisfy specific Service Level Objectives (SLOs) specified by the consumer.

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 same network infrastructure. IETF Network Slices are defined such that they are independent of the underlying infrastructure connectivity and technologies used. This is to allow an IETF Network Slice consumer to describe their network connectivity and relevant objectives in a common format, independent of the underlying technologies used.

An IETF Network Slice is a well-defined composite of a set of endpoints, the connectivity requirements between subsets of these endpoints, and associated service requirements. The service requirements are expressed in terms of quantifiable characteristics or service level objectives (SLOs). SLOs along with terms Service Level Indicator (SLI) and Service Level Agreement (SLA) are used to define the performance of a service at different levels [I-D.nsdt-teas-ietf-network-slice-definition].

The concept of an IETF network slice is consistent with an enhanced VPN (VPN+) [I-D.ietf-teas-enhanced-vpn]. That is, from a consumer's perspective it looks like a VPN connectivity matrix with additional information about the level of service required between endpoints, while from an operator's perspective it 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.

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.

4.1.18. 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 [I-D.ietf-detnet-data-plane-framework]. 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.

4.1.19. 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].

4.1.20. 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.

4.2. 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.

5. 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.

- 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

5.1. 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].

5.2. 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.

5.3. 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 4.1.11.

5.3.1. 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 5.1.

5.3.2. Considerations for Software Defined Networking

As discussed in Section 4.1.16, 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 4.1.19). The PCE is used to compute a path (see Section 4.1.13) 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.

5.4. 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.

5.5. 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.

5.5.1. Intent-Based Networking

TBD : Jeff Tantsura

5.6. 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.

5.7. 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.

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.

- 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.

- 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 and easy to use (i.e., have clean, convenient, and intuitive user interfaces). 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, 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 5.1.

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 micro-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 periodicity 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. 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 a 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.

- 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 survivablity 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.

6.5.1. 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 jop 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.

6.5.2. 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.

6.6. 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 compliment 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:

- 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 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:

- 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 traffic engineering.

6.7. Network Controllability

Offline and online (see Section 5.2) 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 interdomain 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 rhe 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 and 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 forms 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 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.

TBD : Need some discussion of the security and privacy of TE

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 orginal 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 orriginal text resulting from an Errata Report by Jean-Michel Grimaldi.

The authors 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., Lee, Y., Dhody, D., Randriamasy, S., and L. Contreras, "ALTO Performance Cost Metrics", draft-ietf-alto-performance-metrics-12 (work in progress), July 2020.

[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-07 (work in progress), October 2020.

[I-D.ietf-detnet-data-plane-framework] Varga, B., Farkas, J., Berger, L., Malis, A., and S. Bryant, "DetNet Data Plane Framework", draft-ietf-detnet-data-plane-framework-06 (work in progress), May 2020.

[I-D.ietf-detnet-ip-over-tsn] Varga, B., Farkas, J., Malis, A., and S. Bryant, "DetNet Data Plane: IP over IEEE 802.1 Time Sensitive Networking (TSN)", draft-ietf-detnet-ip-over-tsn-04 (work in progress), November 2020.

[I-D.ietf-idr-rfc5575bis] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", draft-ietf-idr-rfc5575bis-27 (work in progress), October 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-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.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-01 (work in progress), November 2020.

[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>.

[RFC2370] Coltun, R., "The OSPF Opaque LSA Option", RFC 2370, DOI 10.17487/RFC2370, July 1998, <https://www.rfc-editor.org/info/rfc2370>.

[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>.

[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, JL., 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>.

[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>.

[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>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[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>.

[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>.

[RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, DOI 10.17487/RFC7810, May 2016, <https://www.rfc-editor.org/info/rfc7810>.

[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>.

[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>.

[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>.

[RFC8661] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., and S. Litkowski, "Segment Routing MPLS Interworking with LDP", RFC 8661, DOI 10.17487/RFC8661, December 2019, <https://www.rfc-editor.org/info/rfc8661>.

[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>.

[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 triggers 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:

- 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 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 Sonet/SDH, OTN, or 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., PCR, SCR, and MBS 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

This section is a place-holder. It is expected that once work on this document is nearly complete, this section will be updated to provide an overview of the structural and substantive changed from RFC 3272.

TBD

Author's Address

Adrian Farrel (editor)
Old Dog Consulting

Email: adrian@olddog.co.uk

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2021

Y. Lee, Ed.
Samsung Electronics
D. Dhody, Ed.
G. Fioccola
Q. Wu, Ed.
Huawei Technologies
D. Ceccarelli
Ericsson
J. Tantsura
Apstra

November 2, 2020

Traffic Engineering (TE) and Service Mapping Yang Model
draft-ietf-teas-te-service-mapping-yang-05

Abstract

This document provides a YANG data model to map customer service models (e.g., the L3VPN Service Model (L3SM)) to Traffic Engineering (TE) models (e.g., the TE Tunnel or the Virtual Network (VN) model). This model is referred to as TE Service Mapping Model and is applicable generically to the operator's need for seamless control and management of their VPN services with TE tunnel support.

The model is principally used to allow monitoring and diagnostics of the management systems to show how the service requests are mapped onto underlying network resource and TE models.

Table of Contents

1. Introduction

Data models are a representation of objects that can be configured or monitored within a system. Within the IETF, YANG [RFC7950] is the language of choice for documenting data models, and YANG models have been produced to allow configuration or modelling of a variety of network devices, protocol instances, and network services. YANG data models have been classified in [RFC8199] and [RFC8309].

Framework for Abstraction and Control of Traffic Engineered Networks (ACTN) [RFC8453] introduces an architecture to support virtual network services and connectivity services.
[I-D.ietf-teas-actn-vn-yang] defines a YANG model and describes how customers or end-to-end orchestrator can request and/or instantiate a generic virtual network service. [I-D.ietf-teas-actn-yang] describes the way IETF YANG models of different classifications can be applied to the ACTN interfaces. In particular, it describes how customer service models can be mapped into the CNC-MDSC Interface (CMI) of the ACTN architecture.

The models presented in this document are also applicable in generic context [RFC8309] as part of Customer Service Model used between Service Orchestrator and Customer.

[RFC8299] provides a L3VPN service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

[RFC8466] provides a L2VPN service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

[I-D.ietf-ccamp-l1csm-yang] provides a L1 connectivity service delivery YANG model for PE-based VPNs. The scope of that draft is

limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

While the IP/MPLS Provisioning Network Controller (PNC) is responsible for provisioning the VPN service on the Provider Edge (PE) nodes, the Multi-Domain Service Coordinator (MDSC) can coordinate how to map the VPN services onto Traffic Engineering (TE) tunnels. This is consistent with the two of the core functions of the MDSC specified in [RFC8453]:

- Customer mapping/translation function: This function is to map customer requests/commands into network provisioning requests that can be sent to the PNC according to the business policies that have been provisioned statically or dynamically. Specifically, it provides mapping and translation of a customer's service request into a set of parameters that are specific to a network type and technology such that the network configuration process is made possible.

- Virtual service coordination function: This function translates customer service-related information into virtual network service operations in order to seamlessly operate virtual networks while meeting a customer's service requirements. In the context of ACTN, service/virtual service coordination includes a number of service orchestration functions such as multi-destination load balancing, guarantees of service quality, bandwidth and throughput. It also includes notifications for service fault and performance degradation and so forth.

Section 2 describes a set of TE and service related parameters that this document addresses as "new and advanced parameters" that are not included in generic service models. Section 3 discusses YANG modelling approach.

Apart from the service model, the TE mapping is equally applicable to the Network Models (L3 VPN Service Network Model (L3NM) [I-D.ietf-opsawg-l3sm-l3nm], L2 VPN Service Network Model (L2NM) [I-D.ietf-opsawg-l2nm] etc.). See Section 3.2 for details.

1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

The terminology for describing YANG data models is found in [RFC7950].

1.1.1. Requirements Language

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. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

1.3. 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
inet	ietf-inet-types	[RFC6991]
tsm-types	ietf-te-service-mapping-types	[RFCXXXX]
l1csm	ietf-l1csm	[I-D.ietf-ccamp-l1csm-yang]
l2vpn-svc	ietf-l2vpn-svc	[RFC8466]
l3vpn-svc	ietf-l3vpn-svc	[RFC8299]
l1-tsm	ietf-l1csm-te-service-mapping	[RFCXXXX]
l2-tsm	ietf-l2sm-te-service-mapping	[RFCXXXX]
l3-tsm	ietf-l3sm-te-service-mapping	[RFCXXXX]
vn	ietf-vn	[I-D.ietf-teas-actn-vn-yang]
nw	ietf-network	[RFC8345]
te-types	ietf-te-types	[RFC8776]
te	ietf-te	[I-D.ietf-teas-yang-te]
l2vpn-ntw	ietf-l2vpn-ntw	[I-D.ietf-opsawg-l2nm]
l3vpn-ntw	ietf-l3vpn-ntw	[I-D.ietf-opsawg-l3sm-l3nm]
rt	ietf-routing	[RFC8349]
sr-policy	ietf-sr-policy	[I-D.ietf-spring-sr-policy-yang]

Table 1: Prefixes and corresponding YANG modules

Note: The RFC Editor should replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

2. TE and Service Related Parameters

While L1/L2/L3 service models (L1CSM, L2SM, L3SM) are intended to provide service-specific parameters for VPN service instances, there are a number of TE Service related parameters that are not included in these service models.

Additional 'service parameters and policies' that are not included in the aforementioned service models are addressed in the YANG models defined in this document.

2.1. VN/Tunnel Selection Requirements

In some cases, the service requirements may need addition TE tunnels to be established. This may occur when there are no suitable existing TE tunnels that can support the service requirements, or when the operator would like to dynamically create and bind tunnels to the VPN such that they are not shared by other VPNs, for example, for network slicing. The establishment of TE tunnels is subject to the network operator's policies.

To summarize, there are three modes of VN/Tunnel selection operations to be supported as follows. Additional modes may be defined in the future.

- New VN/Tunnel Binding - A customer could request a VPN service based on VN/Tunnels that are not shared with other existing or future services. This might be to meet VPN isolation requirements. Further, the YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the ACTN VN. The VN (and TE tunnels) could be bound to the VPN and not used for any other VPN. Under this mode, the following sub-categories can be supported:

  1. Hard Isolation with deterministic characteristics: A customer could request a VPN service using a set of TE Tunnels with deterministic characteristics requirements (e.g., no latency variation) and where that set of TE Tunnels must not be shared with other VPN services and must not compete for bandwidth or other network resources with other TE Tunnels.

  2. Hard Isolation: This is similar to the above case but without the deterministic characteristics requirements.

  3. Soft Isolation: The customer requests a VPN service using a set of TE tunnels which can be shared with other VPN services.

- VN/Tunnel Sharing - A customer could request a VPN service where new tunnels (or a VN) do not need to be created for each VPN and can be shared across multiple VPNs. Further, the mapping YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the tunnels in use. No modification of the properties of a tunnel (or VN) is allowed in this mode: an existing tunnel can only be selected.

- VN/Tunnel Modify - This mode allows the modification of the properties of the existing VN/tunnel (e.g., bandwidth).

- o TE Mapping Template - This mode allows a VPN service to be bound to a mapping template containing constraints and optimization criteria. This allows mapping with the underlay TE characteristics without first creating a VN or tunnels to map. The VPN service could be mapped to a template first. Once the VN/ Tunnels are actually created/selected for the VPN service, this mode is no longer used and replaced with the above modes.

2.2. Availability Requirement

Availability is another service requirement or intent that may influence the selection or provisioning of TE tunnels or a VN to support the requested service. Availability is a probabilistic measure of the length of time that a VPN/VN instance functions without a network failure.

The availability level will need to be translated into network specific policies such as the protection/reroute policy associated with a VN or Tunnel. The means by which this is achieved is not in the scope of this document.

3. YANG Modeling Approach

This section provides how the TE and Service mapping parameters are supported using augmentation of the existing service models (i.e., [I-D.ietf-ccamp-l1csm-yang], [RFC8466], and [RFC8299]). Figure 1 shows the scope of the Augmented LxSM Model.

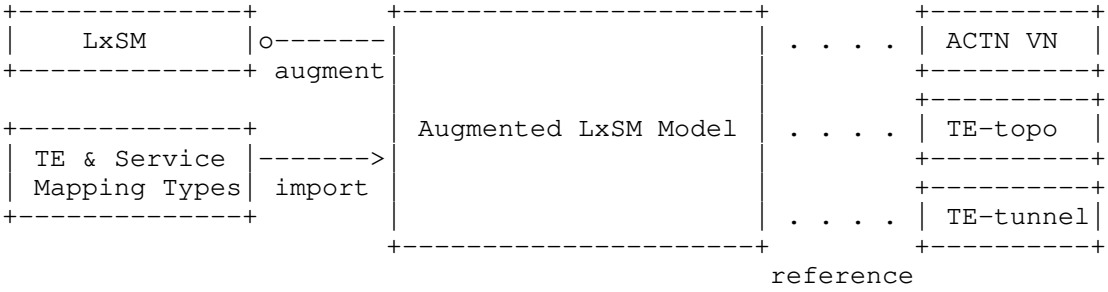


Figure 1: Augmented LxSM Model

The Augmented LxSM model (where x=1,2,3) augments the basic LxSM model while importing the common TE and Service related parameters (defined in Section 2) grouping information from TE and Service Mapping Types. The TE and Service Mapping Types (ietf-te-service-mapping-types) module is the repository of all common groupings imported by each augmented LxSM model. Any future service models would import this mapping-type common model.

The role of the augmented LxSm service model is to expose the mapping relationship between service models and TE models so that VN/VPN service instantiations provided by the underlying TE networks can be viewed outside of the MDSC, for example by an operator who is diagnosing the behaviour of the network. It also allows for the customers to access operational state information about how their services are instantiated with the underlying VN, TE topology or TE tunnels provided that the MDSC operator is willing to share that information. This mapping will facilitate a seamless service management operation with underlay-TE network visibility.

As seen in Figure 1, the augmented LxSM service model records a mapping between the customer service models and the ACTN VN YANG model. Thus, when the MDSC receives a service request it creates a VN that meets the customer's service objectives with various constraints via TE-topology model [RFC8795], and this relationship is recorded by the Augmented LxSM Model. The model also supports a mapping between a service model and TE-topology or a TE-tunnel.

The YANG models defined in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

3.1. Forward Compatibility

The YANG module defined in this document supports three existing service models via augmenting while sharing the common TE and Service Mapping Types.

It is possible that new service models will be defined at some future time and that it will be desirable to map them to underlying TE constructs in the same way as the three existing models are augmented.

3.2. TE and Network Models

The L2/L3 network models (L2NM, L3NM) are intended to describe a VPN Service in the Service Provider Network. It containts information of the Service Provider network and might include allocated resources. It can be used by network controllers to manage and control the VPN Service configuration in the Service Provider network.

Similar to service model, the existing network models (i.e., [I-D.ietf-opsawg-l3sm-l3nm], and [I-D.ietf-opsawg-l2nm]) are augmented to include the TE and Service mapping parameters. Figure 2 shows the scope of the Augmented LxNM Model.

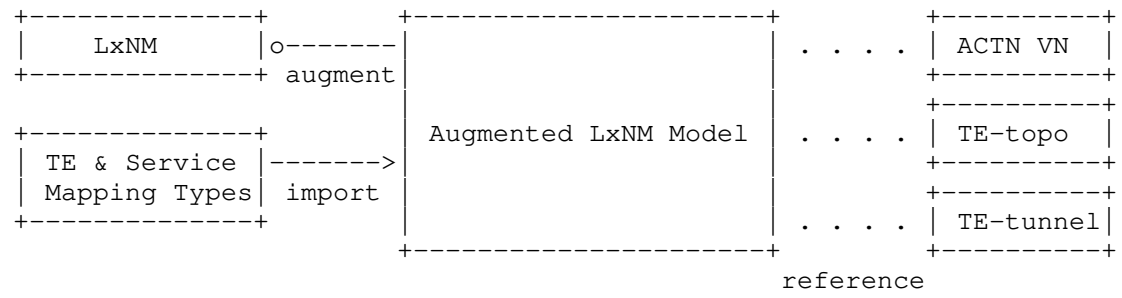
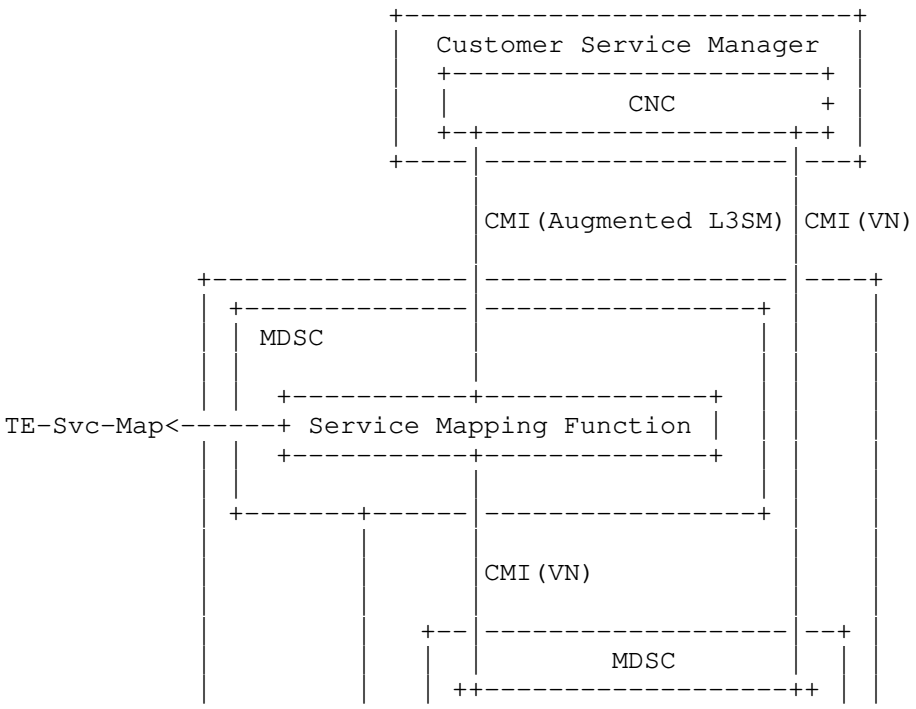


Figure 2: Augmented LxNM Model

The Augmented LxNM model (where x=2,3) augments the basic LxNM model while importing the common TE mapping related parameters (defined in Section 2) grouping information from TE and Service Mapping Types. The role of the augmented LxNM network model is to expose the mapping relationship between network models and TE models.

4. L3VPN Architecture in the ACTN Context

Figure 3 shows the architectural context of this document referencing the ACTN components and interfaces.



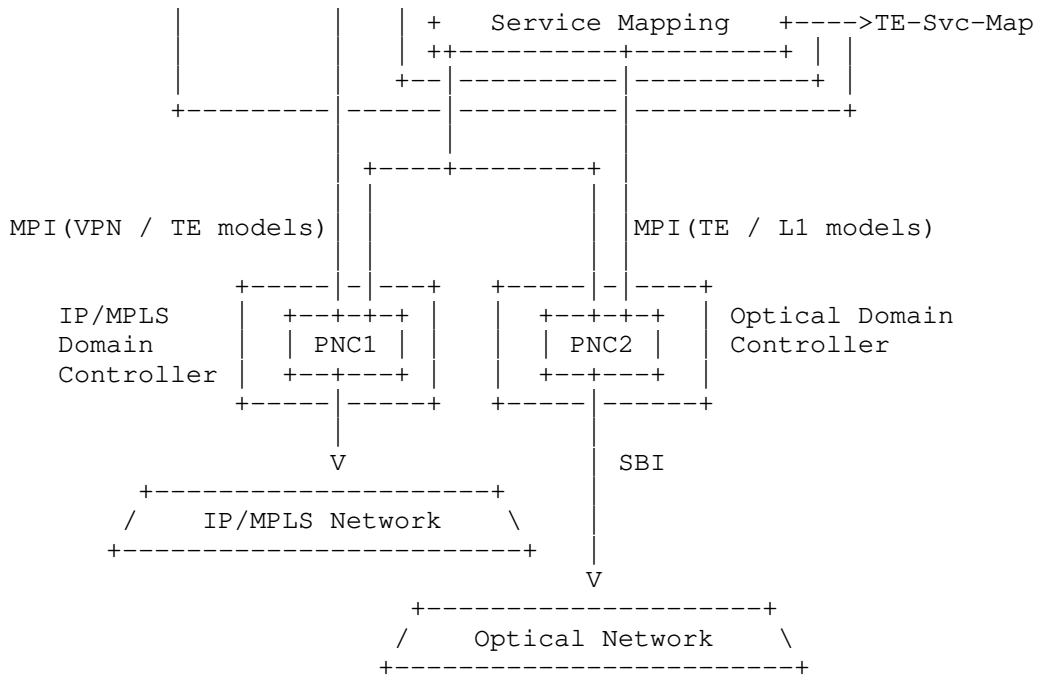


Figure 3: L3VPN Architecture from the IP+Optical Network Perspective

There are three main entities in the ACTN architecture and shown in Figure 3.

- CNC: The Customer Network Controller is responsible for generating service requests. In the context of an L3VPN, the CNC uses the Augmented L3SM to express the service request and communicate it to the network operator.

- MDSC: This entity is responsible for coordinating a L3VPN service request (expressed via the Augmented L3SM) with the IP/MPLS PNC and the Transport PNC. For TE services, one of the key responsibilities of the MDSC is to coordinate with both the IP PNC and the Transport PNC for the mapping of the Augmented L3VPN Service Model to the ACTN VN model. In the VN/TE-tunnel binding case, the MDSC will need to coordinate with the Transport PNC to dynamically create the TE-tunnels in the transport network as needed. These tunnels are added as links in the IP/MPLS Layer topology. The MDSC coordinates with IP/MPLS PNC to create the TE-tunnels in the IP/MPLS layer, as part of the ACTN VN creation.

- PNC: The Provisioning Network Controller is responsible for configuring and operating the network devices. Figure 2 shows two distinct PNCs.

  * IP/MPLS PNC (PNC1): This entity is responsible for device configuration to create PE-PE L3VPN tunnels for the VPN customer and for the configuration of the L3VPN VRF on the PE nodes. Each network element would select a tunnel based on the configuration.

  * Transport PNC (PNC2): This entity is responsible for device configuration for TE tunnels in the transport networks.

There are four main interfaces shown in Figure 2.

- CMI: The CNC-MDSC Interface is used to communicate service requests from the customer to the operator. The requests may be expressed as Augmented VPN service requests (L2SM, L3SM), as connectivity requests (L1CSM), or as virtual network requests (ACTN VN).

- MPI: The MDSC-PNC Interface is used by the MDSC to orchestrate networks under the control of PNCs. The requests on this interface may use TE tunnel models, TE topology models, VPN network configuration models or layer one connectivity models.

- SBI: The Southbound Interface is used by the PNC to control network devices and is out of scope for this document.

The TE Service Mapping Model as described in this document can be used to see the mapping between service models and VN models and TE Tunnel/Topology models. That mapping may occur in the CNC if a service request is mapped to a VN request. Or it may occur in the MDSC where a service request is mapped to a TE tunnel, TE topology, or VPN network configuration model. The TE Service Mapping Model may be read from the CNC or MDSC to understand how the mapping has been made and to see the purpose for which network resources are used.

As shown in Figure 2, the MDSC may be used recursively. For example, the CNC might map a L3SM request to a VN request that it sends to a recursive MDSC.

The high-level control flows for one example are as follows:

1. A customer asks for an L3VPN between CE1 and CE2 using the Augmented L3SM model.

2. The MDSC considers the service request and local policy to determine if it needs to create a new VN or any TE Topology, and if that is the case, ACTN VN YANG [I-D.ietf-teas-actn-vn-yang] is used to configure a new VN based on this VPN and map the VPN service to the ACTN VN. In case an existing tunnel is to be used, each device will select which tunnel to use and populate this mapping information.

3. The MDSC interacts with both the IP/MPLS PNC and the Transport PNC to create a PE-PE tunnel in the IP network mapped to a TE tunnel in the transport network by providing the inter-layer access points and tunnel requirements. The specific service information is passed to the IP/MPLS PNC for the actual VPN configuration and activation.

   A. The Transport PNC creates the corresponding TE tunnel matching with the access point and egress point.
   B. The IP/MPLS PNC maps the VPN ID with the corresponding TE tunnel ID to bind these two IDs.

4. The IP/MPLS PNC creates/updates a VRF instance for this VPN customer. This is not in the scope of this document.

4.1. Service Mapping

Augmented L3SM and L2SM can be used to request VPN service creation including the creation of sites and corresponding site network access connection between CE and PE. A VPN-ID is used to identify each VPN service ordered by the customer. The ACTN VN can be used further to establish PE-to-PE connectivity between VPN sites belonging to the same VPN service. A VN-ID is used to identify each virtual network established between VPN sites.

Once the ACTN VN has been established over the TE network (maybe a new VN, maybe modification of an existing VN, or maybe the use of an unmodified existing VN), the mapping between the VPN service and the ACTN VN service can be created.

4.2. Site Mapping

The elements in Augmented L3SM and L2SM define site location parameters and constraints such as distance and access diversity that can influence the placement of network attachment points (i.e, virtual network access points (VNAP)). To achieve this, a central directory can be set up to establish the mapping between location parameters and constraints and network attachment point location. Suppose multiple attachment points are matched, the management system

can use constraints or other local policy to select the best candidate network attachment points.

After a network attachment point is selected, the mapping between VPN site and VNAP can be established as shown in Table 1.

Site	Site Network Access	Location (Address, Postal Code, State, City,Country Code)	Access Diversity (Constraint-Type, Group-id,Target Group-id)	PE
SITE1	ACCESS1	(,,US,NewYork,)	(10,PE-Diverse,10)	PE1
SITE2	ACCESS2	(,,CN,Beijing,)	(10,PE-Diverse,10)	PE2
SITE3	ACCESS3	(,,UK,London,)	(12,same-PE,12)	PE4
SITE4	ACCESS4	(,,FR,Paris,)	(20,Bearer-Diverse,20)	PE7

Table 2: : Mapping Between VPN Site and VNAP

5. Applicability of TE-Service Mapping in Generic context

As discussed in the Introduction Section, the models presented in this document are also applicable generically outside of the ACTN architecture. [RFC8309] defines Customer Service Model between Customer and Service Orchestrator and Service Delivery Model between Service Orchestrator and Network Orchestrator(s). TE-Service mapping models defined in this document can be regarded primarily as Customer Service Model and secondarily as Service Deliver Model.

6. YANG Data Trees

6.1. Service Mapping Types

```
module: ietf-te-service-mapping-types
  +--rw te-mapping-templates
     +--rw te-mapping-template* [id]
        +--rw id                  te-mapping-template-id
        +--rw description?        string
        +--rw map-type?           identityref
        +--rw path-constraints
        |  +--rw te-bandwidth
        |  |  +--rw (technology)?
        |  |     +--:(generic)
```

```
|  |        +--rw generic?   te-bandwidth
|  +--rw link-protection?          identityref
|  +--rw setup-priority?           uint8
|  +--rw hold-priority?            uint8
|  +--rw signaling-type?           identityref
|  +--rw path-metric-bounds
|  |  +--rw path-metric-bound* [metric-type]
|  |     +--rw metric-type    identityref
|  |     +--rw upper-bound?   uint64
|  +--rw path-affinities-values
|  |  +--rw path-affinities-value* [usage]
|  |     +--rw usage    identityref
|  |     +--rw value?   admin-groups
|  +--rw path-affinity-names
|  |  +--rw path-affinity-name* [usage]
|  |     +--rw usage            identityref
|  |     +--rw affinity-name* [name]
|  |        +--rw name    string
|  +--rw path-srlgs-lists
|  |  +--rw path-srlgs-list* [usage]
|  |     +--rw usage     identityref
|  |     +--rw values*   srlg
|  +--rw path-srlgs-names
|  |  +--rw path-srlgs-name* [usage]
|  |     +--rw usage    identityref
|  |     +--rw names*   string
|  +--rw disjointness?             te-path-disjointness
+--rw optimizations
   +--rw (algorithm)?
      +--:(metric) {path-optimization-metric}?
      |  +--rw optimization-metric* [metric-type]
      |  |  +--rw metric-type
      |  |  |       identityref
      |  |  +--rw weight?                              uint8
      |  |  +--rw explicit-route-exclude-objects
      |  |  |     ...
      |  |  +--rw explicit-route-include-objects
      |  |        ...
      |  +--rw tiebreakers
      |     +--rw tiebreaker* [tiebreaker-type]
      |        ...
      +--:(objective-function)
               {path-optimization-objective-function}?
         +--rw objective-function
            +--rw objective-function-type?   identityref
```

6.2. Service Models

6.2.1. L3SM

```
   module: ietf-l3sm-te-service-mapping
     augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services
               /l3vpn-svc:vpn-service:
       +--rw te-service-mapping!
          +--rw te-mapping
             +--rw map-type?                          identityref
             +--rw availability-type?                 identityref
             +--rw (te)?
                +--:(vn)
                |  +--rw vn-list*
                |           -> /vn:vn/vn-list/vn-id
                +--:(te-topo)
                |  +--rw vn-topology-id?
                |  |       te-types:te-topology-id
                |  +--rw abstract-node?
                |           -> /nw:networks/network/node/node-id
                +--:(te-tunnel)
                |  +--rw te-tunnel-list*            te:tunnel-ref
                |  +--rw sr-policy*
                |           [policy-color-ref policy-endpoint-ref]
                |           {sr-policy}?
                |     +--rw policy-color-ref       leafref
                |     +--rw policy-endpoint-ref    leafref
                +--:(te-mapping-template) {template}?
                   +--rw te-mapping-template-ref?   leafref
     augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site
               /l3vpn-svc:site-network-accesses
               /l3vpn-svc:site-network-access:
       +--rw (te)?
          +--:(vn)
          |  +--rw ap-list*
          |           -> /vn:ap/access-point-list/access-point-id
          +--:(te)
             +--rw ltp?       te-types:te-tp-id
```

6.2.2. L2SM

```
   module: ietf-l2sm-te-service-mapping
     augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services
               /l2vpn-svc:vpn-service:
       +--rw te-service-mapping!
          +--rw te-mapping
             +--rw map-type?                          identityref
             +--rw availability-type?                 identityref
             +--rw (te)?
                +--:(vn)
                |  +--rw vn-list*
                |           -> /vn:vn/vn-list/vn-id
                +--:(te-topo)
                |  +--rw vn-topology-id?
                |  |       te-types:te-topology-id
                |  +--rw abstract-node?
                |           -> /nw:networks/network/node/node-id
                +--:(te-tunnel)
                |  +--rw te-tunnel-list*            te:tunnel-ref
                |  +--rw sr-policy*
                |           [policy-color-ref policy-endpoint-ref]
                |           {sr-policy}?
                |     +--rw policy-color-ref       leafref
                |     +--rw policy-endpoint-ref    leafref
                +--:(te-mapping-template) {template}?
                   +--rw te-mapping-template-ref?   leafref
     augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site
               /l2vpn-svc:site-network-accesses
               /l2vpn-svc:site-network-access:
       +--rw (te)?
          +--:(vn)
          |  +--rw ap-list*
          |           -> /vn:ap/access-point-list/access-point-id
          +--:(te)
             +--rw ltp?       te-types:te-tp-id
```

6.2.3. L1CSM

```
module: ietf-l1csm-te-service-mapping
  augment /l1csm:l1-connectivity/l1csm:services/l1csm:service:
    +--rw te-service-mapping!
       +--rw te-mapping
          +--rw map-type?                          identityref
          +--rw availability-type?                 identityref
          +--rw (te)?
             +--:(vn)
             |  +--rw vn-list*
             |           -> /vn:vn/vn-list/vn-id
             +--:(te-topo)
             |  +--rw vn-topology-id?
             |  |       te-types:te-topology-id
             |  +--rw abstract-node?
             |           -> /nw:networks/network/node/node-id
             +--:(te-tunnel)
             |  +--rw te-tunnel-list*            te:tunnel-ref
             |  +--rw sr-policy*
             |           [policy-color-ref policy-endpoint-ref]
             |           {sr-policy}?
             |     +--rw policy-color-ref       leafref
             |     +--rw policy-endpoint-ref    leafref
             +--:(te-mapping-template) {template}?
                +--rw te-mapping-template-ref?   leafref
  augment /l1csm:l1-connectivity/l1csm:access/l1csm:unis/l1csm:uni:
    +--rw (te)?
       +--:(vn)
       |  +--rw ap-list*
       |           -> /vn:ap/access-point-list/access-point-id
       +--:(te)
          +--rw ltp?       te-types:te-tp-id
```

6.3. Network Models

6.3.1. L3NM

```
   module: ietf-l3nm-te-service-mapping
     augment /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services
               /l3vpn-ntw:vpn-service:
       +--rw te-service-mapping!
          +--rw te-mapping
             +--rw map-type?                          identityref
             +--rw availability-type?                 identityref
             +--rw (te)?
                +--:(vn)
                |  +--rw vn-list*
                |           -> /vn:vn/vn-list/vn-id
                +--:(te-topo)
                |  +--rw vn-topology-id?
                |  |       te-types:te-topology-id
                |  +--rw abstract-node?
                |           -> /nw:networks/network/node/node-id
                +--:(te-tunnel)
                |  +--rw te-tunnel-list*            te:tunnel-ref
                |  +--rw sr-policy*
                |           [policy-color-ref policy-endpoint-ref]
                |           {sr-policy}?
                |     +--rw policy-color-ref       leafref
                |     +--rw policy-endpoint-ref    leafref
                +--:(te-mapping-template) {template}?
                   +--rw te-mapping-template-ref?   leafref
     augment /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services
               /l3vpn-ntw:vpn-service/l3vpn-ntw:vpn-nodes
               /l3vpn-ntw:vpn-node/l3vpn-ntw:vpn-network-accesses
               /l3vpn-ntw:vpn-network-access:
       +--rw (te)?
          +--:(vn)
          |  +--rw ap-list*
          |           -> /vn:ap/access-point-list/access-point-id
          +--:(te)
             +--rw ltp?       te-types:te-tp-id
```

6.3.2. L2NM

```
   module: ietf-l2nm-te-service-mapping
     augment /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services
               /l2vpn-ntw:vpn-service:
       +--rw te-service-mapping!
          +--rw te-mapping
             +--rw map-type?                          identityref
             +--rw availability-type?                 identityref
             +--rw (te)?
                +--:(vn)
                |  +--rw vn-list*
                |           -> /vn:vn/vn-list/vn-id
                +--:(te-topo)
                |  +--rw vn-topology-id?
                |  |       te-types:te-topology-id
                |  +--rw abstract-node?
                |           -> /nw:networks/network/node/node-id
                +--:(te-tunnel)
                |  +--rw te-tunnel-list*            te:tunnel-ref
                |  +--rw sr-policy*
                |           [policy-color-ref policy-endpoint-ref]
                |           {sr-policy}?
                |     +--rw policy-color-ref       leafref
                |     +--rw policy-endpoint-ref    leafref
                +--:(te-mapping-template) {template}?
                   +--rw te-mapping-template-ref?   leafref
     augment /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services
               /l2vpn-ntw:vpn-service/l2vpn-ntw:vpn-nodes
               /l2vpn-ntw:vpn-node/l2vpn-ntw:vpn-network-accesses
               /l2vpn-ntw:vpn-network-access:
       +--rw (te)?
          +--:(vn)
          |  +--rw ap-list*
          |           -> /vn:ap/access-point-list/access-point-id
          +--:(te)
             +--rw ltp?       te-types:te-tp-id
```

7. YANG Data Models

The YANG codes are as follows:

7.1. ietf-te-service-mapping-types

```
 <CODE BEGINS> file "ietf-te-service-mapping-types@2020-11-02.yang"
 module ietf-te-service-mapping-types {
   yang-version 1.1;
   namespace
```

```
    "urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types";
  prefix tsm-types;

  /* Import inet-types */

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }

  /* Import inet-types */

  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering";
  }

  /* Import network model */

  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }

  /* Import TE model */

  import ietf-te {
    prefix te;
    reference
      "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
       Engineering Tunnels and Interfaces";
  }

  /* Import VN model */

  import ietf-vn {
    prefix vn;
    reference
      "I-D.ietf-teas-actn-vn-yang: A Yang Data Model for VN Operation";
  }

  /* Import Routing */

  import ietf-routing {
    prefix rt;
```

```
    reference
      "RFC 8349: A YANG Data Model for Routing Management";
  }

  /* Import SR Policy */

  import ietf-sr-policy {
    prefix sr-policy;
    reference
      "I-D.ietf-spring-sr-policy-yang: YANG Data Model for Segment
       Routing Policy";
  }

  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:   Young Lee
               <mailto:younglee.tx@gmail.com>
     Editor:   Dhruv Dhody
               <mailto:dhruv.ietf@gmail.com>
     Editor:   Qin Wu
               <mailto:bill.wu@huawei.com>";
  description
    "This module contains a YANG module for TE & Service mapping
     parameters and policies as a common grouping applicable to
     variuous service models (e.g., L1CSM, L2SM, L3SM, etc.)

     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
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see the
     RFC itself for full legal notices.

     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 (RFC 2119) (RFC 8174) when, and only when,
```

```
      they appear in all capitals, as shown here.";

   revision 2020-11-02 {
     description
       "Initial revision.";
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }

   /*
    * Features
    */

   feature template {
     description
       "Support TE mapping templates.";
   }

   feature sr-policy {
     description
       "Support SR Policy.";
   }

   /*
    * Identity for map-type
    */

   identity map-type {
     description
       "Base identity from which specific map types are derived.";
   }

   identity new {
     base map-type;
     description
       "The new VN/tunnels are binded to the service.";
   }

   identity hard-isolation {
     base new;
     description
       "Hard isolation.";
   }

   identity detnet-hard-isolation {
     base hard-isolation;
     description
       "Hard isolation with deterministic characteristics.";
```

```
   }

   identity soft-isolation {
     base new;
     description
       "Soft-isolation.";
   }

   identity select {
     base map-type;
     description
       "The VPN service selects an existing tunnel with no
        modification.";
   }

   identity modify {
     base map-type;
     description
       "The VPN service selects an existing tunnel and allows to modify
        the properties of the tunnel (e.g., b/w)";
   }

   identity template {
     base map-type;
     description
       "The VPN service selects an TE mapping template with path
        constraints and optimization criteria";
   }

   /*
    * Identity for availability-type
    */

   identity availability-type {
     description
       "Base identity from which specific map 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 te-mapping-template-id {
     type inet:uri;
     description
       "Identifier for a TE mapping template.  The precise
        structure of the te-mapping-template-id will be up
        to the implementation. The identifier SHOULD be
        chosen such that the same template will always be
        identified through the same identifier, even if the
        data model is instantiated in separate datastores.";
   }

   /*
    * Groupings
    */

   grouping te-ref {
     description
       "The reference to TE.";
     choice te {
       description
         "The TE";
       case vn {
         leaf-list vn-list {
           type leafref {
```

```
            path "/vn:vn/vn:vn-list/vn:vn-id";
          }
          description
            "The reference to VN";
          reference
            "RFC 8453: Framework for Abstraction and Control of TE
             Networks (ACTN)";
        }
      }
      case te-topo {
        leaf vn-topology-id {
          type te-types:te-topology-id;
          description
            "An identifier to the TE Topology Model where the abstract
             nodes and links of the Topology can be found for Type 2
             VNS";
          reference
            "RFC 8795: YANG Data Model for Traffic Engineering (TE)
             Topologies";
        }
        leaf abstract-node {
          type leafref {
            path "/nw:networks/nw:network/nw:node/nw:node-id";
          }
          description
            "A reference to the abstract node in TE Topology";
          reference
            "RFC 8795: YANG Data Model for Traffic Engineering (TE)
             Topologies";
        }
      }
      case te-tunnel {
        leaf-list te-tunnel-list {
          type te:tunnel-ref;
          description
            "Reference to TE Tunnels";
          reference
            "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
             Engineering Tunnels and Interfaces";
        }
        list sr-policy {
          if-feature "sr-policy";
          key "policy-color-ref policy-endpoint-ref";
          description
            "SR Policy";
          leaf policy-color-ref {
            type leafref {
              path
```

```
                "/rt:routing/sr-policy:segment-routing"
              + "/sr-policy:traffic-engineering/sr-policy:policies"
              + "/sr-policy:policy/sr-policy:color";
            }
            description
              "Reference to sr-policy color";
          }
          leaf policy-endpoint-ref {
            type leafref {
              path
                "/rt:routing/sr-policy:segment-routing"
              + "/sr-policy:traffic-engineering/sr-policy:policies"
              + "/sr-policy:policy/sr-policy:endpoint";
            }
            description
              "Reference to sr-policy endpoint";
          }
        }
      }
      case te-mapping-template {
        if-feature "template";
        leaf te-mapping-template-ref {
          type leafref {
            path "/tsm-types:te-mapping-templates/"
               + "tsm-types:te-mapping-template/tsm-types:id";
          }
          description
            "An identifier to the TE Mapping Template where the TE
             constraints and optimization criteria are specified.";
        }
      }
    }
  }

  //grouping

  grouping te-endpoint-ref {
    description
      "The reference to TE endpoints.";
    choice te {
      description
        "The TE";
      case vn {
        leaf-list ap-list {
          type leafref {
            path "/vn:ap/vn:access-point-list/vn:access-point-id";
          }
          description
```

```
            "The reference to VN AP";
          reference
            "RFC 8453: Framework for Abstraction and Control of TE
             Networks (ACTN)";
        }
      }
      case te {
        leaf ltp {
          type te-types:te-tp-id;
          description
            "Reference LTP in the TE-topology";
          reference
            "RFC 8795: YANG Data Model for Traffic Engineering (TE)
             Topologies";
        }
      }
    }
  }

  //grouping

  grouping te-mapping {
    description
      "Mapping between Services and TE";
    container te-mapping {
      description
        "Mapping between Services and TE";
      leaf map-type {
        type identityref {
          base map-type;
        }
        description
          "Isolation Requirements, Tunnel Bind or
           Tunnel Selection";
      }
      leaf availability-type {
        type identityref {
          base availability-type;
        }
        description
          "Availability Requirement for the Service";
      }
      uses te-ref;
    }
  }

  //grouping
```

```
   container te-mapping-templates {
     description
       "The TE constraints and optimization criteria";
     list te-mapping-template {
       key "id";
       leaf id {
         type te-mapping-template-id;
         description
           "Identification of the Template to be used.";
       }
       leaf description {
         type string;
         description
           "Description of the template.";
       }
       leaf map-type {
         type identityref {
           base map-type;
         }
         must "0 = derived-from-or-self(.,'template')" {
           error-message "The map-type must be other than "
                       + "TE mapping template";
         }
         description
           "Map type for the VN/Tunnel creation/
            selection.";
       }
       uses te-types:generic-path-constraints;
       uses te-types:generic-path-optimization;
       description
         "List for templates.";
     }
   }
 }


 <CODE ENDS>
```

7.2. Service Models

7.2.1. ietf-l3sm-te-service-mapping

```
 <CODE BEGINS> file "ietf-l3sm-te-service-mapping@2020-11-02.yang"
 module ietf-l3sm-te-service-mapping {
   yang-version 1.1;
   namespace
     "urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping";
   prefix l3-tsm;
```

```
   import ietf-te-service-mapping-types {
     prefix tsm-types;
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }
   import ietf-l3vpn-svc {
     prefix l3vpn-svc;
     reference
       "RFC 8299: YANG Data Model for L3VPN Service Delivery";
   }

   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:   Young Lee
                <mailto:younglee.tx@gmail.com>
      Editor:   Dhruv Dhody
                <mailto:dhruv.ietf@gmail.com>
      Editor:   Qin Wu
                <mailto:bill.wu@huawei.com>";
   description
     "This module contains a YANG module for the mapping of Layer 3
      Service Model (L3SM) to the TE and VN.

      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
      (https://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX; see the
      RFC itself for full legal notices.

      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 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.";

   revision 2020-11-02 {
```

```
     description
       "Initial revision.";
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }

   /*
    * Augmentation to L3SM
    */

   augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services"
         + "/l3vpn-svc:vpn-service" {
     description
       "L3SM augmented to include TE parameters and mapping";
     container te-service-mapping {
       presence "Indicates L3 service to TE mapping";
       description
         "Container to augment l3sm to TE parameters and mapping";
       uses tsm-types:te-mapping;
     }
   }

   //augment

   augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site"
         + "/l3vpn-svc:site-network-accesses"
         + "/l3vpn-svc:site-network-access" {
     description
       "This augment is only valid for TE mapping of L3SM network-access
        to TE endpoints";
     uses tsm-types:te-endpoint-ref;
   }

   //augment
 }

 <CODE ENDS>
```

7.2.2. ietf-l2sm-te-service-mapping

```
 <CODE BEGINS> file "ietf-l2sm-te-service-mapping@2020-11-02.yang"
 module ietf-l2sm-te-service-mapping {
   yang-version 1.1;
   namespace
     "urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping";
   prefix l2-tsm;

   import ietf-te-service-mapping-types {
```

```
    prefix tsm-types;
    reference
      "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
  }
  import ietf-l2vpn-svc {
    prefix l2vpn-svc;
    reference
      "RFC 8466: A YANG Data Model for Layer 2 Virtual Private Network
       (L2VPN) Service Delivery";
  }

  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:   Young Lee
               <mailto:younglee.tx@gmail.com>
     Editor:   Dhruv Dhody
               <mailto:dhruv.ietf@gmail.com>
     Editor:   Qin Wu
               <mailto:bill.wu@huawei.com>";
  description
    "This module contains a YANG module for the mapping of Layer 2
     Service Model (L2SM) to the TE and VN.

     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
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see the
     RFC itself for full legal notices.

     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 (RFC 2119) (RFC 8174) when, and only when,
     they appear in all capitals, as shown here.";

  revision 2020-11-02 {
```

```
    description
      "Initial revision.";
    reference
      "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
  }

  /*
   * Augmentation to L2SM
   */

  augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/"
        + "l2vpn-svc:vpn-service" {
    description
      "L2SM augmented to include TE parameters and mapping";
    container te-service-mapping {
      presence "indicates L2 service to te mapping";
      description
        "Container to augment L2SM to TE parameters and mapping";
      uses tsm-types:te-mapping;
    }
  }

  //augment

  augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site"
        + "/l2vpn-svc:site-network-accesses"
        + "/l2vpn-svc:site-network-access" {
    description
      "This augment is only valid for TE mapping of L2SM network-access
       to TE endpoints";
    uses tsm-types:te-endpoint-ref;
  }

  //augment
}


<CODE ENDS>
```

7.2.3. ietf-l1csm-te-service-mapping

```
<CODE BEGINS> file "ietf-l1csm-te-service-mapping@2020-11-02.yang"
module ietf-l1csm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping";
  prefix l1-tsm;
```

```
   import ietf-te-service-mapping-types {
     prefix tsm-types;
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }
   import ietf-l1csm {
     prefix l1csm;
     reference
       "I-D.ietf-ccamp-l1csm-yang: A YANG Data Model for L1 Connectivity
        Service Model (L1CSM)";
   }

   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:   Young Lee
                <mailto:younglee.tx@gmail.com>
      Editor:   Dhruv Dhody
                <mailto:dhruv.ietf@gmail.com>
      Editor:   Qin Wu
                <mailto:bill.wu@huawei.com>";
   description
     "This module contains a YANG module for the mapping of
      Layer 1 Connectivity Service Module (L1CSM) to the TE and VN

      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
      (https://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX; see the
      RFC itself for full legal notices.

      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 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.";
```

```
   revision 2020-11-02 {
     description
       "Initial revision.";
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }

   /*
    * Augmentation to L1CSM
    */

   augment "/l1csm:l1-connectivity/l1csm:services/l1csm:service" {
     description
       "L1CSM augmented to include TE parameters and mapping";
     container te-service-mapping {
       presence "Indicates L1 service to TE mapping";
       description
         "Container to augment L1CSM to TE parameters and mapping";
       uses tsm-types:te-mapping;
     }
   }

   //augment

   augment "/l1csm:l1-connectivity/l1csm:access/l1csm:unis/"
         + "l1csm:uni" {
     description
       "This augment is only valid for TE mapping of L1CSM UNI to TE
        endpoints";
     uses tsm-types:te-endpoint-ref;
   }

   //augment
 }

 <CODE ENDS>
```

7.3. Network Models

7.3.1. ietf-l3nm-te-service-mapping

```
 <CODE BEGINS> file "ietf-l3nm-te-service-mapping@2020-11-02.yang"
 module ietf-l3nm-te-service-mapping {
   yang-version 1.1;
   namespace
     "urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping";
   prefix l3nm-tsm;
```

```
   import ietf-te-service-mapping-types {
     prefix tsm-types;
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }
   import ietf-l3vpn-ntw {
     prefix l3vpn-ntw;
     reference
       "I-D.ietf-opsawg-l3sm-l3nm: A Layer 3 VPN Network YANG Model";
   }

   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:   Young Lee
                <mailto:younglee.tx@gmail.com>
      Editor:   Dhruv Dhody
                <mailto:dhruv.ietf@gmail.com>
      Editor:   Qin Wu
                <mailto:bill.wu@huawei.com>";
   description
     "This module contains a YANG module for the mapping of Layer 3
      Network Model (L3NM) to the TE and VN.

      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
      (https://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX; see the
      RFC itself for full legal notices.

      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 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.";

   revision 2020-11-02 {
```

```
     description
       "Initial revision.";
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }

   /*
    * Augmentation to L3NM
    */

   augment "/l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services"
         + "/l3vpn-ntw:vpn-service" {
     description
       "L3SM augmented to include TE parameters and mapping";
     container te-service-mapping {
       presence "Indicates L3 network to TE mapping";
       description
         "Container to augment l3nm to TE parameters and mapping";
       uses tsm-types:te-mapping;
     }
   }

   //augment

   augment "/l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services"
         + "/l3vpn-ntw:vpn-service"
         + "/l3vpn-ntw:vpn-nodes/l3vpn-ntw:vpn-node"
         + "/l3vpn-ntw:vpn-network-accesses"
         + "/l3vpn-ntw:vpn-network-access" {
     description
       "This augment is only valid for TE mapping of L3NM network-access
        to TE endpoints";
     uses tsm-types:te-endpoint-ref;
   }

   //augment
 }

   <CODE ENDS>
```

7.3.2. ietf-l2nm-te-service-mapping

```
 <CODE BEGINS> file "ietf-l2nm-te-service-mapping@2020-11-02.yang"
 module ietf-l2nm-te-service-mapping {
   yang-version 1.1;
   namespace
     "urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping";
   prefix l2nm-tsm;
```

```
   import ietf-te-service-mapping-types {
     prefix tsm-types;
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }
   import ietf-l2vpn-ntw {
     prefix l2vpn-ntw;
     reference
       "I-D.ietf-l2nm: A Layer 2 VPN Network YANG Model";
   }

   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:   Young Lee
                <mailto:younglee.tx@gmail.com>
      Editor:   Dhruv Dhody
                <mailto:dhruv.ietf@gmail.com>
      Editor:   Qin Wu
                <mailto:bill.wu@huawei.com>";
   description
     "This module contains a YANG module for the mapping of Layer 2
      Network Model (L2NM) to the TE and VN.

      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
      (https://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX; see the
      RFC itself for full legal notices.

      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 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.";

   revision 2020-11-02 {
```

```
     description
       "Initial revision.";
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }

   /*
    * Augmentation to L2NM
    */

   augment "/l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services"
         + "/l2vpn-ntw:vpn-service" {
     description
       "L2SM augmented to include TE parameters and mapping";
     container te-service-mapping {
       presence "Indicates L2 network to TE mapping";
       description
         "Container to augment l2nm to TE parameters and mapping";
       uses tsm-types:te-mapping;
     }
   }

   //augment

   augment "/l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services"
         + "/l2vpn-ntw:vpn-service"
         + "/l2vpn-ntw:vpn-nodes/l2vpn-ntw:vpn-node"
         + "/l2vpn-ntw:vpn-network-accesses"
         + "/l2vpn-ntw:vpn-network-access" {
     description
       "This augment is only valid for TE mapping of L2NM network-access
        to TE endpoints";
     uses tsm-types:te-endpoint-ref;
   }

   //augment
 }
```

<CODE ENDS>

8. Security Considerations

The YANG modules defined in this document is designed to be accessed via network management protocol such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is 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 pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in the YANG moduleS which 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:

- /l3vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Service mapping.
- /l3vpn-svc/sites/site/site-network-accesses/site-network-access/te/ - configure TE Endpoint mapping.
- /l2vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Service mapping.
- /l2vpn-svc/sites/site/site-network-accesses/site-network-access/te/ - configure TE Endpoint mapping.
- /l1-connectivity/services/service/te-service-mapping/te-mapping/ - configure TE Service mapping.
- /l1-connectivity/access/unis/uni/te/ - configure TE Endpoint mapping.
- /l3vpn-ntw/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Network mapping.
- /l3vpn-ntw/vpn-services/vpn-service/vpn-nodes/vpn-node/vpn-network-accesses/vpn-network-access/te/ - configure TE Endpoint mapping.
- /l2vpn-ntw/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Network mapping.
- /l2vpn-ntw/vpn-services/vpn-service/vpn-nodes/vpn-node/vpn-network-accesses/vpn-network-access/te/ - configure TE Endpoint mapping.

Unauthorized access to above list can adversely affect the VPN service.

Some of the readable data nodes in the 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. The TE related parameters attached to the VPN service can leak sensitive information about the network. This is apploicable to all elements in the yang models defined in this document.

This document has no RPC defined.

9. IANA Considerations

This document request the IANA to register four URIs in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registrations are requested -

URI: urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

This document request the IANA to register four YANG modules in the "YANG Module Names" registry [RFC6020], as follows -

```
Name:      ietf-te-service-mapping-types
Namespace: urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types
Prefix:    tsm-types
Reference: [This.I-D]

Name:      ietf-l3sm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping
Prefix:    l3-tsm
Reference: [This.I-D]

Name:      ietf-l2sm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping
Prefix:    l2-tsm
Reference: [This.I-D]

Name:      ietf-l1csm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping
Prefix:    l1-tsm
Reference: [This.I-D]

Name:      ietf-l3nm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping
Prefix:    l3nm-tsm
Reference: [This.I-D]

Name:      ietf-l2nm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping
Prefix:    l2nm-tsm
Reference: [This.I-D]
```

10. Acknowledgements

We thank Diego Caviglia, and Igor Bryskin for useful discussions and motivation for this work.

11. References

11.1. Normative References

[I-D.ietf-ccamp-l1csm-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-l1csm-yang-12 (work in progress), September 2020.

[I-D.ietf-opsawg-l2nm]
barguil, s., Dios, O., Boucadair, M., Munoz, L., Jalil, L., and J. Ma, "A Layer 2 VPN Network YANG Model", draft-ietf-opsawg-l2nm-00 (work in progress), July 2020.

[I-D.ietf-opsawg-l3sm-l3nm]
barguil, s., Dios, O., Boucadair, M., Munoz, L., and A. Aguado, "A Layer 3 VPN Network YANG Model", draft-ietf-opsawg-l3sm-l3nm-05 (work in progress), October 2020.

[I-D.ietf-spring-sr-policy-yang]
Raza, K., Sawaya, R., Shunwan, Z., Voyer, D., Durrani, M., Matsushima, S., and V. Beeram, "YANG Data Model for Segment Routing Policy", draft-ietf-spring-sr-policy-yang-00 (work in progress), September 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-09 (work in progress), July 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.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[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>.

[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>.

[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>.

11.2. Informative References

[I-D.ietf-teas-actn-yang] Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B., Dios, O., Shin, J., and S. Belotti, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-yang-06 (work in progress), August 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>.

[RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <https://www.rfc-editor.org/info/rfc8199>.

[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>.

Appendix A. Contributor Addresses

Adrian Farrel
Old Dog Consulting

EMail: adrian@olddog.co.uk

Italo Busi
Huawei Technologies

EMail: Italo.Busi@huawei.com

Haomian Zheng
Huawei Technologies

EMail: zhenghaomian@huawei.com

Anton Snitser
Sedonasys

EMail: antons@sedonasys.com

SAMIER BARGUIL GIRALDO
Telefonica

EMail: samier.barguilgiraldo.ext@telefonica.com

Oscar Gonzalez de Dios
Telefonica

EMail: oscar.gonzalezdedios@telefonica.com

Carlo Perocchio
Ericsson

EMail: carlo.perocchio@ericsson.com

Kenichi Ogaki
KDDI
Email: ke-oogaki@kddi.com

Authors' Addresses

Young Lee (editor)
Samsung Electronics

Email: younglee.tx@gmail.com

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Giuseppe Fioccola
Huawei Technologies

Email: giuseppe.fioccola@huawei.com

Qin Wu (editor)
Huawei Technologies

Email: bill.wu@huawei.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Jeff Tantsura
Apstra

Email: jefftant.ietf@gmail.com

TEAS
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

Y. Lee
Samsung Electronics
X. Liu
Volta Networks
LM. Contreras
Telefonica

November 2, 2020

DC aware TE topology model
draft-llc-teas-dc-aware-topo-model-00

Abstract

This document proposes the extension of the TE topology model for including information related to data center resource capabilities.

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 May 6, 2021.

Copyright Notice

Table of Contents

1. Introduction

More and more service providers are deploying cloud computing facilities in order to host different kinds of services and applications. Such facilities can be generally referred as Datacenter Points of Presence (DC-PoPs). Those DCs will consist on a number of servers and networking elements for connecting all of them with the transport network. Depending on the number of servers in the data center, there will be distinct capabilities in terms of CPUs, memory and storage available for deploying and running the aforementioned services.

In such distributed and interconnected DC-PoPs, both computing and topological information are of interest for determining the optimal DC where to deploy a given service or application.

This document propose a DC-aware extension for the topology model.

2. Datacenter information

The relevant information for datacenter capabilities can be described in different ways. One potential manner is to describe resource capabilities such as CPU, memory, storage, etc. This can be done in terms of total, used and free capacity for each of the parameters of interest. Another form of populating the information is by describing those resource capabilities as a bundled, usually referred as quota or flavor. In this respect, reference bundles such as the ones proposed by the Common Network Function Virtualisation Infrastructure Telecom Taskforce (CNTT) [CNTT].

Additional information can refer to the management capabilities of the compute infrastructure, such as hypervisor details or virtualization technologies available.

Finally, all can be complemented with information related to the networking details for reaching the aforementioned compute capabilities (IP addressed, bandwidth, etc).

3. Model structure

```
   module: ietf-dcpop-dc
       +--rw dcpop
          +--rw dc* [id]
          |  +--rw hypervisor* [id]
          |  |  +--rw ram
          |  |  |  +--rw total?   uint32
          |  |  |  +--rw used?    uint32
          |  |  |  +--rw free?    uint32
          |  |  +--rw disk
          |  |  |  +--rw total?   uint32
          |  |  |  +--rw used?    uint32
          |  |  |  +--rw free?    uint32
          |  |  +--rw vcpu
          |  |  |  +--rw total?   uint16
          |  |  |  +--rw used?    uint16
          |  |  |  +--rw free?    uint16
          |  |  +--rw instance*   -> /dcpop/dc/instance/id
          |  |  +--rw id          string
          |  |  +--rw name?       string
          |  +--rw instance* [id]
          |  |  +--rw flavor
          |  |  |  +--rw disk?    uint32
          |  |  |  +--rw ram?     uint32
          |  |  |  +--rw vcpus?   uint16
          |  |  |  +--rw id?      string
          |  |  |  +--rw name?    string
          |  |  +--rw image
          |  |  |  +--rw checksum    string
          |  |  |  +--rw size        uint32
          |  |  |  +--rw format
          |  |  |  |  +--rw container?   enumeration
          |  |  |  |  +--rw disk?        enumeration
          |  |  |  +--rw id?         string
          |  |  |  +--rw name?       string
          |  |  +--rw hypervisor?   -> /dcpop/dc/hypervisor/id
          |  |  +--rw port*         -> /dcpop/dc/network/subnetwork/port/id
          |  |  +--rw project?      string
          |  |  +--rw status?       enumeration
          |  |  +--rw id            string
          |  |  +--rw name?         string
          |  +--rw image* [id]
          |  |  +--rw checksum    string
          |  |  +--rw size        uint32
          |  |  +--rw format
          |  |  |  +--rw container?   enumeration
          |  |  |  +--rw disk?        enumeration
```

```
          |   |  +--rw id          string
          |   |  +--rw name?       string
          |   +--rw flavor* [id]
          |   |  +--rw disk?    uint32
          |   |  +--rw ram?     uint32
          |   |  +--rw vcpus?   uint16
          |   |  +--rw id       string
          |   |  +--rw name?    string
          |   +--rw dc-monitoring-param* [name]
          |   |  +--rw name            string
          |   |  +--rw value-string?   string
          |   +--rw network* [id]
          |   |  +--rw subnetwork* [id]
          |   |  |  +--rw port* [id]
          |   |  |  |  +--rw ip-address?   inet:ip-address
          |   |  |  |  +--rw instance?     -> /dcpop/dc/instance/id
          |   |  |  |  +--rw project?      string
          |   |  |  |  +--rw status?       enumeration
          |   |  |  |  +--rw id            string
          |   |  |  |  +--rw name?         string
          |   |  |  +--rw project?   string
          |   |  |  +--rw status?    enumeration
          |   |  |  +--rw id         string
          |   |  |  +--rw name?      string
          |   |  +--rw dhcp-agent* [id]
          |   |  |  +--rw enabled?   boolean
          |   |  |  +--rw pools* [ip-address]
          |   |  |  |  +--rw ip-address    inet:ip-address
          |   |  |  +--rw project?   string
          |   |  |  +--rw status?    enumeration
          |   |  |  +--rw id         string
          |   |  |  +--rw name?      string
          |   |  +--rw project?      string
          |   |  +--rw status?       enumeration
          |   |  +--rw id            string
          |   |  +--rw name?         string
          |   |  +--rw dcpop-ref?      -> /dcpop/dcpop-id
          |   +--rw ap*                    -> /actn-vn:actn/ap
   /access-point-list/access-point-id
          |   +--rw dcpop-ref?               -> /dcpop/dcpop-id
          |   +--rw id                       string
          |   +--rw name?                    string
          +--rw dcpop-id?   string
```

4. Security Considerations

The data-model in this document does not have any security implications. The model is designed to be accessed via NETCONF [RFC6241], thus the security considerations for the NETCONF protocol are applicable here.

5. IANA Considerations

This draft does not include any IANA considerations

6. References

[CNTT] "Common NFVI for Telco Reference Model, Release 4.0", September 2020, <https://cntt-n.github.io/CNTT/doc/ref_model/>.

[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>.

Acknowledgments

The work of L.M. Contreras has been partly funded by the European Commission through the H2020 project 5GROWTH (Grant Agreement no. 856709).

Authors' Addresses

Young Lee
Samsung Electronics
Seoul
South Korea

Email: younglee.tx@gmail.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Luis M. Contreras
Telefonica
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://lmcontreras.com/

teas
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

R. Rokui
Nokia
S. Homma
NTT
K. Makhijani
Futurewei
LM. Contreras
Telefonica
J. Tantsura
Apstra, Inc.
November 2, 2020

Definition of IETF Network Slices
draft-nsdt-teas-ietf-network-slice-definition-01

Abstract

This document provides a definition of the term "IETF Network Slice" for use within the IETF and specifically as a reference for other IETF documents that describe or use aspects of network slices.

The document also describes the characteristics of an IETF network slice, related terms and their meanings, and explains how IETF network slices can be used in combination with end-to-end network slices or independent of them.

Table of Contents

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 wrt network resources use. In this document, as detailed

in the subsequent sections, we refer to this connectivity and resource commitment as an IETF network slice. Services that might benefit from the network slices include but not limited to:

- 5G services (e.g. eMBB, URLLC, mMTC)(See [TS.23.501-3GPP])
- Network wholesale services
- Network infrastructure sharing among operators
- NFV connectivity and Data Center Interconnect

The use cases are further described in [I-D.nsdt-teas-ns-framework].

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. 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.

1.1. Rationale

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 same network infrastructure.

An IETF Network Slice is a well-defined structure of connectivity requirements and associated network behaviors. IETF Network Slices are defined such that they are independent of the underlying infrastructure connectivity and technologies used. This is to allow an IETF Network Slice consumer 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 [TS.23.501-3GPP].

2. Terms and Abbreviations

The terms and abbreviations used in this document are listed below.

- NS: Network Slice
- NSC: Network Slice Controller
- NBI: NorthBound Interface
- SBI: SouthBound Interface
- SLI: Service Level Indicator
- SLO: Service Level Objective
- SLA: Service Level Agreement

The above terminology is defined in greater details in the remainder of this document.

3. 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 with a set of shared or dedicated network resources, that are used to satisfy specific Service Level Objectives (SLOs).

IETF Network Slice specification 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 is defined in terms of several quantifiable characteristics or service level objectives (SLOs). SLOs along with terms Service Level Indicator (SLI) and Service Level Agreement (SLA) are used to define the performance of a service at different levels.

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 network slice is expressed in terms of the set of SLOs that are to be delivered for the different connections between endpoints.

A Service Level Agreement (SLA) is an explicit or implicit contract between the consumer of an IETF Network Slice and the provider of the slice. The SLA is expressed in terms of a set of SLOs and may include commercial terms as well as the consequences of missing/ violating the SLOs they contain.

Additional descriptions of IETF network slice attributes is covered in [I-D.contreras-teas-slice-nbi].

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 network slice.

4.1.2. Minimal Set of SLOs

This document defines a minimal set of SLOs and later systems or standards could extend this set as per Section 4.1.3.

SLOs can be categorized in to 'Directly Measurable Objectives' or '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'. While 'Indirectly Measurable Objectives' include security, geographical restrictions, maximum occupancy level objectives. The later standard might define other SLOs as needed.

Editor's Note TODO: Minimal set describes most commonly used objectives to describe network behavior. Other directly or indirectly measurable objectives may be requested by that customer of an IETF network slice.

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], s 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.

Security

An IETF Network Slice consumer may request that the network applies encryption or other security techniques to traffic flowing between endpoints.

Note that the use of security or the violation of this SLO is not directly observable by the IETF Network Slice consumer and cannot be measured as a quantifiable metric.

Also note that the objective may include request for encryption (e.g., [RFC4303]) between the two endpoints explicitly to meet architecture recommendations as in [TS33.210] or for compliance with [HIPAA] and/or [PCI].

Editor's Note: Please see more discussion on security in Section 10.

4.1.3. Other Objectives

Additional SLOs may be defined to provide additional description of the IETF network slice that a consumer requests.

If the IETF Network Slice consumer 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).

Maximal occupancy for an IETF network slice should be provided. Since it carries traffic for multiple flows between the two endpoints, the objectives should also say if they are for the entire connection, group of flows or on per flow basis. Maximal occupancy should specify the scale of the flows (i.e. maximum number of flows to be admitted) and optionally a maximum number of countable resource units, e.g IP or MAC addresses a slice might consume.

4.2. IETF Network Slice Endpoints

As noted in Section 3, an IETF network slice describes connectivity between endpoints across the underlying network. This connectivity may be be point-to-point, point-to-multipoint (P2MP), multipoint-to-point, or multipoint-to-multipoint.

The characteristics of IETF network slice endpoints (NSEs) are as follows.

- They are conceptual points of connection of a customer network, network function, device, or application to the IETF network slice. This might include 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.

- They are identified in a request provided by the consumer of an IETF Network Slice when the IETF Network Slice is requested.

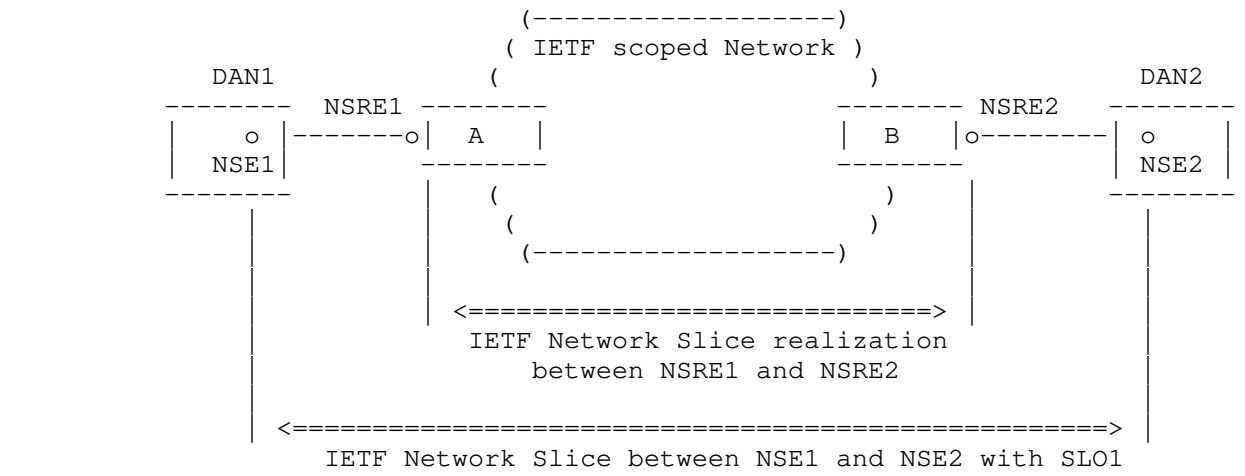
- An NSE is identified a unique identifier and/or a unique name and other data. A non-exhaustive list of other data includes IPv4 or IPv6 address, VLAN tag, port number, connectivity type (P2P, P2MP, MP2MP).

Note that the NSE is different from access points (AP) defined in [RFC8453] as an AP is a logical identifier to identify the shared link between the customer and the operator where as NSE is an identifier of an endpoint. Also NSE is different from TE Link Termination Point (LTP) defined in [I-D.ietf-teas-yang-te-topo] as it is a conceptual point of connection of a TE node to one of the TE links on a TE node.

The NSE is similar to the Termination Point (TP) defined in [RFC8345] and can contain more attributes. NSE could be modeled by augmenting the TP model.

There is another type of the endpoints called "IETF Network Slice Realization Endpoints (NSREs)". These endpoints are allocated and assigned by the network controller during the realization of an IETF Network Slice and are technology-specific, i.e. they depend on the network technology used during the IETF Network Slice realization. The identification of NSREs forms part of the realization of the IETF Network Slice and is implementation and deployment specific.

Figure 1 shows an example of an IETF Network Slice and its realization between multiple NSEs and NSREs.



Legend:
DAN: Device, application and/or network function

Figure 1: An IETF Network Slice between NSEs and its realization between NSREs

4.2.1. IETF Network Slice Connectivity Types

The IETF Network Slice connection types can be point to point (P2P), point to multipoint (P2MP), multi-point to point (MP2P), or multi-point to multi-point (MP2MP). They will requested by the higher level operation system.

4.3. IETF Network Slice Composition

Operationally, an IETF Network Slice maybe decomposed in two or more IETF Network Slices as specified below. Decomposed network slices are then 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. IETF Network Slice Structure

Editor's note: This content of this section merged with Relationship with E2E slice discussion.

An IETF Network Slice is a set of connections among various endpoints to form a logical network that meets the SLOs agreed upon.

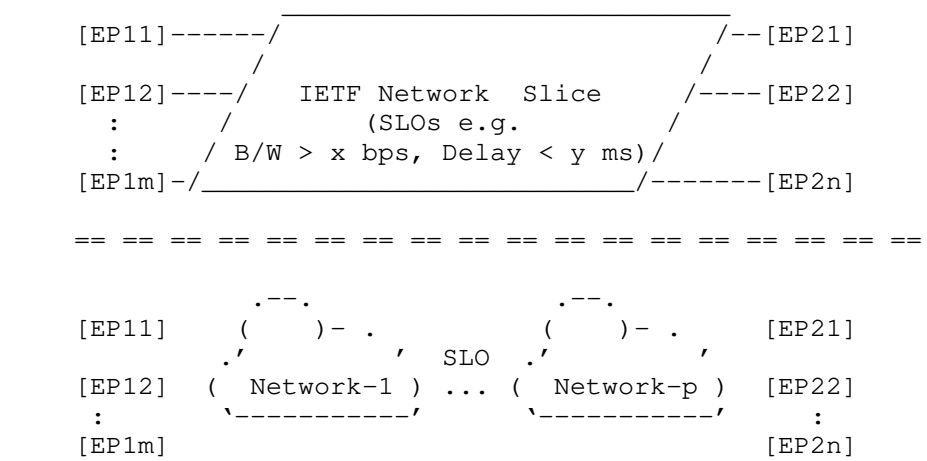


Figure 2: IETF Network slice

Figure 2 illustrates a case where an IETF Network Slice provides connectivity between a set of endpoints pairs with specific characteristics for each SLO (e.g. guaranteed minimum bandwidth of x bps and guaranteed delay of no more than y ms). The endpoints may be distributed in the underlay networks, and an IETF Network Slice can be deployed across multiple network domains. Also, the endpoints on 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 [TS.23.501-3GPP].

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. 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.

Consumer: A consumer is the requester of an IETF Network Slice. Consumers may request monitoring of SLOs. A consumer may manage the IETF Network Slice service directly by interfacing with the IETF Network Slice controller or indirectly through an orchestrator.

Orchestrator: An orchestrator is an entity that composes different services, resource and network requirements. It interfaces with the IETF Network Slice controllers.

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 Network Slice controllers 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 Network Slice Controller(s).

Network Controller: is a form of network infrastructure controller that offers network resources to 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.

7. 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 3).

NSC Northbound Interface (NBI): The NSC Northbound Interface is an interface between a consumer's higher level operation system (e.g., a network slice orchestrator) and the NSC. It is a technology agnostic interface. The consumer 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 consumer.

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.

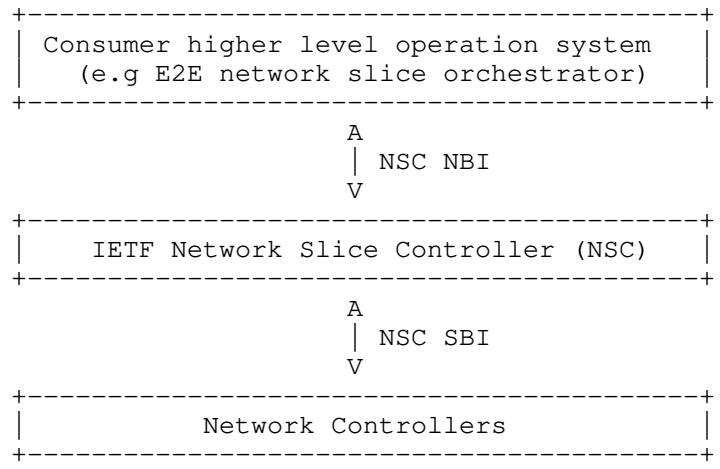


Figure 3: Interface of IETF Network Slice Controller

8. Realizing IETF Network Slice

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 through VPNs (see, for example, [I-D.ietf-teas-enhanced-vpn], a variety of tunneling technologies such as Segment Routing, MPLS, etc. Accordingly, endpoints may be realized as physical or logical service or network functions.

9. Isolation in IETF Network Slices

Editor's note: This content is a work in progress. The section on isolation is too descriptive.

An IETF Network Slice consumer may request, that the IETF Network Slice delivered to them is isolated from any other network slices of services delivered to any other customers. It is expected that the changes to the other network slices of services do not have any negative impact on the delivery of the IETF Network Slice. In a more general sense, isolation can be classified in the following ways:

Traffic Separation: Traffic of one network slice should not be subjected to policies and forwarding rules of other network slices.

Interference Avoidance: Changes in other network slices should not impact to the SLOs of the network slice. Here the changes in other network slice may include the changes in connectivity, traffic volume, traffic pattern, etc.

Service Assurance: In case service degradation is unacceptable due to unpredictable network situations producing service degradation (e.g., major congestion events, etc.), explicit reservation of resources in the network maybe requested for a reduces set IETF network slices.

9.1. Isolation as a Service Requirement

Isolation is an important requirement of IETF network slices for services like critical services, emergencies, etc. A consumer may express this request through the description of SLOs.

This requirement can be met by simple conformance with other SLOs. For example, traffic congestion (interference from other services) might impact on the latency experienced by an IETF network slice. Thus, in this example, conformance to a latency SLO would be the primary requirement for delivery of the IETF network slice service, and isolation from other services might be only a means to that end.

It should be noted that some aspects of isolation may be measurable by a customer who have the information about the traffic on a number of IETF network slices or other services.

9.2. Isolation in IETF Network Slice Realization

The isolation requirement can be achieved with existing, in-development, and potential new technologies in IETF.

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 or 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 network slices, etc.

9.3. Relationship with Isolation in 5G Network Slice

Editor’s note: This 5G subsection should not be added to terminology. it does not add value to the definitions.

In the context of 5G network slice, "isolation level" is listed as one of the attributes which can be used to characterize the type of network slice [GSMA Generic Network Slice Template]. For 5G network slice, different types of isolation are considered, including physical and logical isolation. Physical isolation refers to different physical network entities, and logical isolation is further classified into virtual resource isolation, network function isolation and tenant/service isolation.

10. Security Considerations

Editor’s Note: Need further improvement; work in progress.

This document specifies terminology and has no direct effect on the security of implementations or deployments.

As noted in Section 4.1.2, some aspects of security may be expressed in SLOs and so form part of the service delivered as an IETF network slice. As further mentioned in Section 8, there is an underlying asumption that traffic presented to an IETF network slice will not be misdelivered to an endpoint that is not part of that IETF network slice.

Furthermore, the nature of conformance to SLOs 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. This concern can be strengthened by the stipulation of "isolation" as an SLO.

Note, however, that 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.

11. IANA Considerations

This memo includes no request to IANA.

12. Acknowledgment

The entire TEAS NS design team and everyone participating in those discussion has contributed to this draft. Particularly, Eric Gray, Xufeng Liu, Jie Dong, Adrian Farrel, and Jari Arkko for a thorough review among other contributions.

13. 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.contreras-teas-slice-nbi] Contreras, L., Homma, S., and J. Ordonez-Lucena, "Considerations for defining a Transport Slice NBI", draft-contreras-teas-slice-nbi-01 (work in progress), March 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+) Services", draft-ietf-teas-enhanced-vpn-05 (work in progress), February 2020.

[I-D.ietf-teas-yang-te-topo] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-02 (work in progress), March 2020.

[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>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[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>.

[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>.

[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>.

[TS.23.501-3GPP] 3rd Generation Partnership Project (3GPP), "3GPP TS 23.501 (V16.2.0): System Architecture for the 5G System (5GS); Stage 2 (Release 16)", September 2019, <http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-g20.zip>.

[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>.

Authors' Addresses

Reza Rokui
Nokia
Canada

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
Apstra, Inc.

Email: jefftant.ietf@gmail.com

TEAS
Internet-Draft
Intended status: Standards Track
Expires: April 23, 2021

Shaofu. Peng
Ran. Chen
Gregory. Mirsky
ZTE Corporation
Fengwei. Qin
China Mobile

October 20, 2020

Packet Network Slicing using Segment Routing
draft-peng-teas-network-slicing-04

Abstract

This document presents a mechanism aimed at providing a solution for network slicing in the transport network for 5G services. The proposed mechanism uses a unified administrative instance identifier to distinguish different virtual network resources for both intra-domain and inter-domain network slicing scenarios. Combined with the segment routing technology, the mechanism could be used for both best-effort and traffic engineered services for tenants.

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 April 23, 2021.

Copyright Notice

Table of Contents

1. Introduction

According to 5G context, network slicing is the collection of a set of technologies to create specialized, dedicated logical networks as a service (NaaS) in support of network service differentiation and meeting the diversified requirements from vertical industries. Through the flexible and customized design of functions, isolation mechanisms, and operation and management (O&M) tools, network slicing is capable of providing dedicated virtual networks over a shared infrastructure. A Network Slice Instance (NSI) is the realization of network slicing concept. It is an E2E logical network, which comprises of a group of network functions, resources, and connection relationships. An NSI typically covers multiple technical domains, which include a terminal, access network (AN), transport network (TN) and a core network (CN), as well as a DC domain that hosts third-party applications from vertical industries. Different NSIs may have different network functions and resources. They may also share some of the network functions and resources.

For a transport network, network slicing requires the underlying network to support partitioning of the network resources to provide the client with dedicated (private) networking, computing, and storage resources drawn from a shared pool. The slices may be seen as virtual networks.

This document describes how to realize TN-slice in the underlay network, and analyze the necessity and usage of a new slice-based identifier to represent specific virtual network that is requested by the user of 5G service who have the specific resources and connection requirements. This new slice-based identifier is expected to be not only used for management system, but also for control and data plane in the underlay network.

2. Architecture of TN Slicing

Relationship with NS Design Team:

The current scope of NS design team will focus on the framework of the TN Slice.We would like to make some contributions of it, and we will sent this section to the NS Design Team for dicussion.

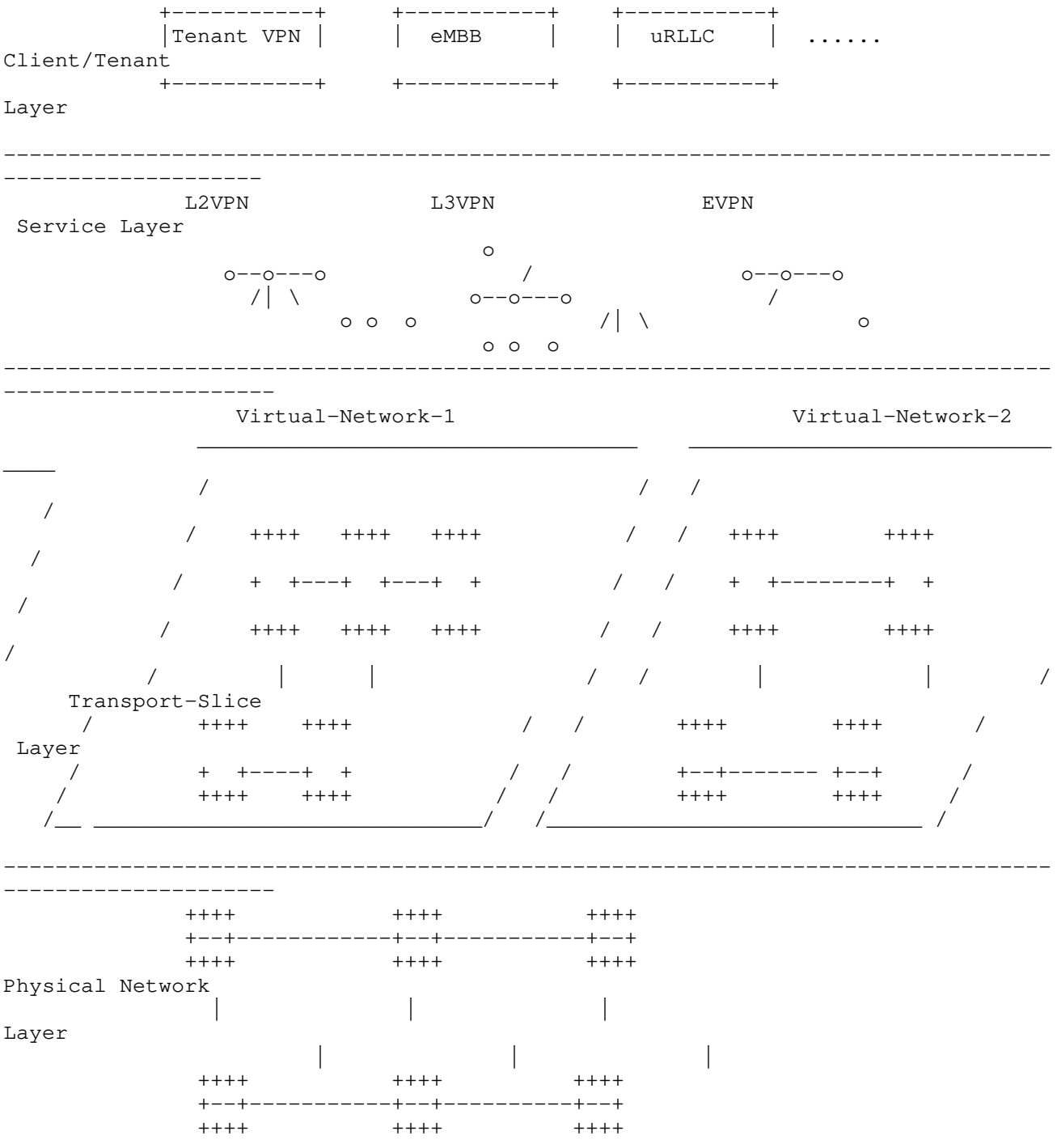


Figure 1 Architecture of TN Slicing

Based on the concept and architecture of Transport slice, the basic requirements and features of Transport slice are as following:

- On-Demand network reconstitution: The slice network can be reconstituted in network topology and node capability to meet service needs. Each slice network has its own specific bandwidth, latency and lifecycle. Different Transport Slice networks are isolated from each other, and have independent topology and network resources.

- Decoupling of Service Slice Layer and Physical Network Layer: The Service Slice Layer and the Physical Network Layer are decoupled, and unaware of the details of each other, which simplifies the deployment of services.

- Similarity of Transport Slice Network and Physical Network for Service Layer: A Transport Slice Network Layer provides network resources to the upper layer (Service Layer) which is the same as the resources provided directly by a physical network from the point view of the upper layer. Services such as VPN service etc. can be deployed directly on the Transport slice network just as they are deployed on the physical network. One Transport slice network can support the deployment of more than one services or VPNs.

- Data Plane Isolation of Transport Slice Network: The TN provides two types of traffic isolation between different TN slices: hard isolation and soft isolation. Hard isolation is implemented by providing independent circuit switched connections for the exclusive use of one slice, such as MTN (Metro Transport Network, see ITU-T G.mtn), and ODUk. Soft isolation is implemented by using a packet technology (e.g., Ethernet VLAN, MPLS tunnel, and VPN). Services of different slices are isolated from each other.

- Transport Slice Network: There may be multiple Sub-TN-slices in a Transport Slice Network, and those Sub-Transport slices may be nested. Different sub-TN-slices can be also combined together for an end-to-end TN slice service.

2.1. Key Technologies of Transport slice

For the transport network forwarding plane slicing, there are basically two kinds of isolation technology: soft isolation technology and hard isolation technology. The soft isolation is a Layer 2 or Layer 3 technology, such as SR/IP/MPLS based tunnel technology and VPN/VLAN based virtualization technology. The hard isolation is a Layer 1 or optical-layer slicing technology based on physically rigid pipelines, such as MTN, OTN and Wavelength Division

Multiplexing (WDM) technologies. In applications, the hybrid hard and soft isolation solution is always used. The hard isolation ensures service isolation, and the soft isolation supports service bandwidth reuse.

So, The Key Technologies of Transport slice should include: Layer-one Data Plane, Layer-Two Data Plane, and Layer-Three Data Plane.

3. Slicing Requirements

3.1. Dedicated Virtual Networks

An end-to-end virtual network with dedicated resources is the advantage of network slicing than traditional DiffServ QoS and VPN. For example, DiffServ QoS can distinguish VoIP traffic and other type of traffic (such as high-definition video, web browsing), but can not distinguish the same type of traffic from different tenants, nor isolation of these traffic at all.

Another example is the IoT traffic of health monitoring network which connected hospital and outpatient, it always has strict privacy and safety requirements, including where the data can be stored and who can access the data, all this can not be satisfied by DiffServ QoS as it has not any function of network computing and storage.

Dedicated VN is a distinct object purchased by a customer, and it provides specific function with predictable performance, guaranteed level of isolation and safety. It is not just as QoS.

3.2. End-to-End Slicing

Only an end-to-end slice and fine-grained network can match ultra delay and safety requirements of special service. End-to-end means that it is constructed with AN-slice, TN-slice, and CN-slice part.

Although 3GPP technical specifications mainly focus on the operation and management of AN-slice and CN-slice, which include some NF (network function) components, TN-slice is also created and destroyed according to the related NSI lifecycle. In fact, the 3GPP management system will request expected link requirements related to the network slice (e.g., topology, QOS parameters) with the help of the management system that handles the TN part related to the slice.

For TN part, the link requirements are independent of the existing domain partition of the network, i.e., any intra- or inter-domain link is the candidate resource for the slice. It is also independent of the existing underlay frame or routing technologies (IGP, BGP,

Segment Routing, Flex-E, etc.), i.e., any L2 or L3 link is the candidate resource.

From the end-to-end slicing requirement, the inter domain resource guarantee needs to be paid more attention to.

3.3. Unified NSI

An NSI is indentified by S-NSSAI (Single Network Slice Selection Assistance Information), which is allocated per PDU session and has semantic global within the AN and CN.

For the purpose of operation and management simplicity, it is also better to have a unified identifier with semantic global to distinguish different TN-slice within the whole TN. TN-slice identifier has a mapping relation with S-NSSAI, perhaps 1:1 or 1:n.

Instead, using different slice identifier across multi-domain of TN for the specific TN-slice will introduce much and unnecessary complexity, especially for case two devices belongs to different domain try to exchange slice-based information directly through the protocol mechanism in control plane, without the help of SDN controller to translate the unified TN-slice identifier to an individual domain-wide indentifier.

3.4. Traffic Engineering

5G system is expected to be able to provide optimized support for a variety of different communication services, different traffic loads, and different end-user communities. For example, the communication services using network slicing may include: vehicle-to-everything (V2X) services, 5G seamless enhanced Mobile BroadBand (eMBB) service with FMC (fixed-mobile convergence), massive IoT connections. Among these service types, high data rates, high traffic densities, low-latency, high-reliability are highlighted requirements.

Traffic engineering mechanism in TN must support the above requirements, bandwidth and delay are two primary TE constraints.

3.5. Summarized Requirements

In summary, the following requirements would be satisfied for the realization of TN-slice:

REQ1: Provide a distinct virtual network, including dedicated topology, computation, and storage resource, not only traditional QoS;

REQ2: Unified NSI for easy operation and maintenance;

REQ3: E2E network slicing, including both intra-domain and inter-domain case;

REQ4: Customization resource for QoS purpose, bandwidth and delay are basic constraints;

REQ5: Layer 2 as well as Layer 3 link resource partition;

4. Conventions Used in This Document

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.

5. Overview of Existing Identifiers

Currently there are multiple existing mature identifiers that could be used to identify all or part of the virtual network's resources in the transport network, such as:

- Administrative Group (AG) described in [RFC3630], [RFC5329], [RFC5305] and Extended Administrative Groups (EAGs) described in [RFC7308]

- Multi-Topology Routing (MTR) described in [RFC5120], [RFC4915], [RFC5340]

- SR policy color described in [I-D.ietf-spring-segment-routing-policy]

- FA-id described in [I-D.ietf-lsr-flex-algo]

However, all these identifiers are not sufficient to meet the above requirements of TN-slice. Note that all these identifiers have use case of their own, besides the network slicing use case. Next, we will discuss each of them to determine their matching of slicing requirements.

5.1. AG and EAG Bit

AG and EAG are limited to serve as a link color scheme used in TE path computation to meet the requirements of TE service for a tenant. It is difficult to use them for an NSI allocation mapping (assuming that each bit position of AG/EAG represents an NSI), and also difficult to use them to represent non-link resources. Hence, they do not meet REQ1, 2. Additionally, AG or EAG cannot be as an

identifier to organize specific forwarding tables or policys for different virtual networks.

AG and EAG can be used as the attribute of both L3 interface and L2-bundles member, to meet REQ5.

Although AG and EAG have semantic global, they don't fully meet REQ3. For example, for an E2E path, inter domain links can also be selected by their AG/EAG attributes, but they can't be adhered to intra domain links through AG/EAG to let a complete view of virtual network be presented.

5.2. Multi-Topology Identifier

MTR is limited to serve as an IGP logical topology scheme only used in the intra-domain scenario. Thus it is challenging to select inter-area link resources based on MT-ID when E2E inter-domain TE path needs to be created for a tenant. And, by definition, MTR is only used to represent topology resources and cannot be used for various other types of resources. That is, it does not meet REQ1, 3.

Different IGP domain within the same TN-slice may be configured with different MT-ID. Thus MT-ID does not meet REQ2.

MT-ID is only as L3 link attribute, not appropriate for L2-bundles member, so it does not meet REQ5.

5.3. SR Policy Color

The color of SR policy defines a TE purpose, which includes a set of constraints such as bandwidth, delay, TE metric, etc. Therefore color is an abstract target, and it is difficult to get a distinct virtual network according to a specific color value. In most cases, only the headend and some other border nodes need to maintain the color template, and a color-based virtual network is hard to present because of too few participants and lack of interaction scheme. That is, the color does not meet REQ1, 2.

We can continue to define TE affinity information in color-template, to select L3 link or L2-bundles member,to meet REQ5.

Note that the color has global semantic, so it meets REQ3.

5.4. Flex-algorithm Identifier

Indeed, FA-id is a short mapping of SR policy color, and it may inherit the matched-degree of the Policy Color. However, FA-id has its own characteristics. A specific FA-id can have more distributed

participants and define explicit link resource so that an explicit FA plane can be created. Unfortunately, different service of the same tenant-slice will define different constraints, resulting in the need to occupy more FA-id resources for single tenant-slice. The relationship between FA-id and slice is not clear. That is, FA-id does not meet REQ1.

On the other hand, FA-id, like MT-ID, is limited to serve as an IGP algorithm scheme used in the intra-domain scenario. It is challenging to select inter-area (especially inter-AS) link resources according to FA-id when the E2E inter-domain TE path needs to be created for the tenant. So, FA-id does not meet REQ3.

Different IGP domain within the same TN-slice may configure different FA-ids, so it does not meet REQ2.

What is more important, the path in FA plane identified by FA-id is MP2P LSP, so it is hard to define bandwidth reservation for service. So, FA-id does not meet REQ4. Unless each link is totally dedicated to a single FA plane, i.e., link resources are not shared among multiple FA plane.

It is possible to let the link include/exclude rules defined by FA-id be appropriate for both L3 link and L2-bundles member, to meet REQ5.

5.5. New Slice-based Identifier Introduced

Thus, there needs to introduce a new characteristic of NSI that meets the above-listed requirements to isolate underlay resources, and it is a slice-based identifier.

Firstly, it could serve as TE criteria for TE service, this aspect is like AG/EAG; and secondly, as an identifier to orgnize specific forwarding tables or policys for different virtual networks, this aspect is like MT-ID or FA-id.

This document introduces a new property of NSI called "Administrative Instance Identifier" (AII) and corresponding method of how to instantiate it in the underlay network to match the above-listed requirements.

6. Overview of AII-based Mechanism

SR policy [I-D.ietf-spring-segment-routing-policy], just like traditional RSVP-TE LSP, can be used to realize IETF network slice in underlay network. This document introduce AII-based mechanism to enhance SR policy to support tenant-slice as well as service-slice. It will signal the association of AII and shared resources required

to create and manage an NSI, and steer the packets to the path within the specific NSI according to SR policy color.

SR policy color has semantic global in order to be conveniently exchanged between two endpoints within TN-slice. These two endpoints can configure the same color template information for the same color value. AII, also with global semantic, can be contained in color template to enhance SR policy to create a TE path within global TN-slice identified by AII. Besides TE service served by explicit SR policy instance, best-effort service can be served by AII-specific forwarding tables or policys that are created by default once AII configured.

The following is how AII-based mechanism works.

6.1. Physical Network Partition by AII

Each node and link in the physical network can be colored dynamically, with shared or dedicated resources, to conform with network slicing requirements. As previously mentioned, AII can be used to color nodes and links to partition underlay resources. Also, we may continue to use AG or EAG to color links for traditional TE within a virtual network specified by an AII. A single or multiple AIIs could be configured on each intra-domain or inter-domain link regardless of IGP instance configuration. At the minimum, a link always belongs to default AII (the value is 0).

The extension of the existing IGP-TE mechanisms [RFC3630] and [RFC5305] to distribute AII information as a new TE parameter of a link in the underlay network will be defined in another document.

Using BGP-LS [RFC7752], an SDN controller,can collect topology information of each virtual network specified by AII with related resources, and have a distinct view of each virtual network. Extension of BGP-LS will also be defined in another document.

6.2. Path within AII specific Slice

Using the CSPF algorithm, a TE path for any best-effort (BE) or traffic-engineered (TE) service can be calculated within a virtual network specified by the AII. The computation criteria could be <AII, endogenous criteria> or <AII, traditional TE critieria> for the BE and TE respectively. Combined with segment routing, the TE path could be represented as:

- A single node-SID of the destination node, for the best-effort service intra-domain;

- Several node-SIDs of the border node and the destination node, adjacency-SID of inter-domain link, for the inter-domain best-effort service;

- An explicit adjacency-SID list or compressed with several loose node-SIDs, for traffic engineered service.

To distinguish forwarding behavior of different virtual networks, Prefix-SID and Adjacency-SID need to be allocated per AII with related resources, and advertised in the IGP domain.

6.2.1. SR-BE Path within AII specific Slice

Because packets of the best-effort service could be transported over an MP2P LSP without congestion control, SR-BE path for each virtual network specified by AII to forward best-effort packets may be created in the IGP domain.

This document will register some standard AII value (see section "IANA Considerations") to represent different type of slice. For example, a slice type "normal" represent any slices that have endogenous "min IGP metric" criteria to compute SPF path within the slice. Thus, for a virtual network with slice type "normal", CSPF computation with criteria <AII, min igp-metric> is distributed on each node within the virtual network.

Similarly, a slice type "uRLLC" represent any slices that have endogenous "min delay metric" (Min Unidirectional Link Delay as defined in [RFC7810]) criteria to compute SPF path within the slice. For a virtual network with slice type "uRLLC", CSPF computation with criteria <AII, min delay> is distributed on each node within the virtual network.

Similarly, a slice type "TE" represent any slices that have endogenous "TE metric" (TE default metric as defined in [RFC5305]) criteria to compute SPF path within the slice. For a virtual network with slice type "TE metric", CSPF computation with criteria <AII, min te-metric> is distributed on each node within the virtual network.

That is similar to the behavior in [I-D.ietf-lsr-flex-algo], but the distributed CSPF computation is triggered by AII, using determined criteria without negotiation among noeds.

Besides the best-effort service, SR-BE path for specific AII also provide an escape way for traffic engineering service within the same virtual network when the expected TE purpose can not be meet.

6.2.2. SR-TE Path within AII specific Slice

Other different constraints sets can also be defined to calculate TE paths for different overlay services within the same slice, regardless of the endogenous criteria of the slice. It is suggested that the set of constraints defined should contain the endogenous constraint of the slice. For some traffic engineering services that have strict requirements, especially such as the ulra-reliable low-latency communication service, it SHOULD not transfer over an MP2P LSP to avoid the risk of traffic congestion. The segment list should consist of pure adjacency-SIDs and other service funciton SIDs per AII, with guaranteed resources. The segment list could be computed by headend or SDN controller.

However, stack depth of the segment list MAY be optimized at a later time based on local policies.

6.3. Traffic Steering to SR policy within Slice

The traffic of overlay service can be steerred to the above SR-BE path or SR-TE tunnel/policy instance for the specific virtual network. The overlay service could specify a color for TE purpose.

For example, color 1000 means <AII=10, min igp-metric> to say "I need best-effort forwarding within AII 10 resource", color 1001 means <AII=10, delay=10ms, AG=0x1> to say "I need traffic engineering forwarding within AII 10 resource, and only use links with AG 0x1 to reach guarantee of not exceeding 10ms delay upper bound".

Service with color 1000 will be steered to an SR-BE entry of the table identified by AII, or an SR-TE tunnel/policy in case of inter-domain.

Service with color 1001 will be steered to an SR-TE tunnel/policy.

6.4. Simple Variant of AII-based Slicing Scheme

There is a simple variant of AII-based slicing scheme for initial slicing requirement of service, where the SDN controller in management partition the whole E2E network topology to multiple strictly isolated VNs identified by AII in local, but let the forwarding equipments of the underlay network be totally unware of that.

The overlay service is steered to the SR policy whose path is limited within specific VN using a pure adjacency-segment list.

This variant need not introduce any complex protocol mechanism of control plane in the underlay network, however only for limited scenes. There are no states per AII, except that QoS policy per AII need to be configured in the underlay network.

7. Resource Allocation per AII

7.1. L3 Link Resource AII Configuration

In IGP domain, each numbered or unnumbered L3 link could be configured with AII information and synchronized among IGP neighbors. The IGP link-state database will contain L3 links with AII information to support TE path computation taking account of AII criteria. For a numbered L3 link, it could be represented as a tuple <local node-id, remote node-id, local ip-address, remote ip-address> , for unnumbered it could be <local node-id, remote node-id, local interface-id, remote interface-id>. Each L3 link could be configured to belong to a single AII or multiple AIIs. Note that an L3 link always belongs to default AII(0).

For different <L3 link, AII> tuple it would allocate a different adjacency-SID, as well as advertising with different resource portion such as bandwidth occupied.

Note that AII is independent of IGP instance. An L3 link that is not part of the IGP domain, such as the special purpose for a static route, or an inter-domain link, can also be configured with AII information and allocate adjacency-SID per AII as the same as IGP links. BGP-LS could be used to collect link state data with AII information to the controller, BGP-LS has already provided a mechanism to collect link state data from many source protocols, such as IGP, Direct, Static configuration, etc., to cover network slicing requirements.

When an L3 link joins to to multiple VNs, as traditional ways that these VNs can share the total bandwidth resouce of the link with preemption mode based on packet priority, this is what we know as soft slicing. However, In some other scenarios, a hard slicing scheme can be used to establish a hardened pipe to meet the slicing business requirements, at this time each VN need dedicated bandwidth resouce reserved from the same link, and at each node QoS policy per AII (e.g, traffic shaper per slice) should be used to ensure that the traffic between different slices is isolated and does not affect each other.

7.2. L2 Link Resource AII Configuration

[RFC8668] described how to encode adjacency-SID for each L2 member link of an L3 parent link. In the network slicing scenario, it is beneficial to deploy LAG or another virtual aggregation interface between two nodes. If that, the dedicated link resources belong to different virtual networks could be added or removed on demand, they are treated as L2 member links of a single L3 virtual interface. It is the single L3 virtual interface which needs to occupy IP resource and join the IGP instance. Creating a new slice-specific link on demand or removing the old one is likely to affect little configurations.

For network slicing purpose, [RFC8668] need to be extended to advertise the AII attribute for each L2 member link. In practice, for hard isolation purpose, different L2 member link of the same L3 parent link is SUGGESTED to be configured to belong to different single AII, with different adjacency-SID. Note that in this case, the L3 parent link belongs to default AII(0) (i.e, for this L3 link it is not necessary to allocate adjacency-SID per AII any more), but each L2 member link belongs to the specific non-default AII as well as the shared default AII. An L2 member link maybe a Flex-E channel or ODUK tunnel created/destroyed on demand.

In practice, for hard isolation purpose, different L2 member link of the same L3 parent link SUGGESTED to be configured to belong to different AII, with different adjacency-SID. Note that in this case, the L3 parent link belongs to default AII(0), but each L2 member link belongs to the specific non-default AII. An L2 member link maybe a Flex-E channel or ODUK tunnel created/destroyed on demand.

In the control plane, routing protocol packets following the L3 parent link will select the L2 member link with the highest priority.

In the forwarding plane, data packets that belong to the specific virtual network will pass along the L2 member link with the specific AII value.

TE path computation based on link-state database need inspect the detailed L2 members of an L3 adjacency to select the expected L2 link resource.

7.3. Node Resource AII Configuration

For topology resource, each node needs to allocate node-SID per AII when it joins the related virtual network. All nodes in the IGP domain can run the CSPF algorithm with criteria <AII, endogenous criteria> to compute SR-BE path to any other destination nodes for an

AII-specific virtual network based on the link-state database that containing AII information, so that SR-BE FIB can be constructed for each AII. Static routes could also be added to the AII-specific FIB.

An intra-domain overlay best-effort service belongs to an AII specific virtual network could be directly over the SR-BE path within that VN.

An inter-domain overlay best-effort service belongs to an AII specific virtual network could be over a path containing a single destination node-SID or a segment list containing domain border node-SID and destination node-SID within that VN.

7.4. Service Function Resource AII Configuration

[I-D.ietf-spring-sr-service-programming] introduces the notion of service segments, and describes how to implement service segments and achieve stateless service programming in SR-MPLS and SRv6 networks. The ability of encoding the service segments along with the topological segment enables service providers to forward packets along a specific network path and through VNFs or physical service appliances available in the network. Typically, a Service Function may be any purposeful execution for the packet, such as DPI, firewall, NAT, etc.

The Service Function is independent of topology, it can also be instantiated per AII, each with different priority to be executed or scheduled. For example, a docker container including specific Service Funciton process can be generated or destroyed on demand according to the life-cycle of a particular slice. It will have a particular CPU scheduling priority.

At a node, multiple instance of the same type of Service Function for different slice will allocate different Service SID and advertise to other nodes.

8. E2E Slicing with Centralized Mode

[RFC7752] BGP-LS describes the methodology that using BGP protocol to transfer the Link-State information that maybe originated from IGP instance (for intra-domain topology information) or from local direct interface or static configuration(for inter-domain topology information). [I-D.ietf-idr-bgpls-inter-as-topology-ext] also describes a method to firstly put inter-domain interconnections to IGP instance, then always import data from IGP protocol source to BGP-LS. In any case BGP-LS need extend to transfer the Link-State data with AII information.

An E2E inner-AS SR-TE instance with particular color template could be initiated on PE1, PE1 is head-end and PE2 is destination node. BGP-LS could be used to inform the SDN controller about the underlay network topology information including AII attribute. Thus the controller could calculate E2E TE path within the particular virtual network. Especially an AII specific Adacency-SID of inter-domain link can be included in the E2E SID list.

9. E2E Slicing with Distributed Mode

In some deployments, especially the network evolution from seamless MPLS in practice, operators adopt BGP-LU to build inter-domain MPLS LSP, and overlay service will be directly over BGP-LU LSP.

In this case, the network is divided into some domains and each domain will run its own IGP process. These IGP process are isolated to each other to be simple. That means it is inconvenient to realize network slicing depending on IGP itself with inter-area route leak or redistribution.

For an E2E BGP-LU LSP, if overlay service has TE requirements that defined by a color, the BGP-LU LSP need also have a sense of color, i.e., BGP-LU label could be allocated per color.

At entry node of each domain, BGP-LU LSP generated for specific color will be over intra-domain SR-TE or SR-BE path generated for that color again. At exit node of each domain, BGP-LU LSP generated for specific color will select inter-domain forwarding resource per color. Especially, an ASBR will select slice-specfic inter-AS link according to AII information of color template.

[RFC7911] defined that multiple paths UPDATE message for the same destination prefix can be advertised in BGP, each UPDATE can contain the Color Extended Community ([I-D.ietf-idr-tunnel-encaps]) with different color value. That is a simple existing way to realize BGP-LU color function, with needless new BGP extensions.

10. Combined with SR Flex-algorithm for Stack Depth Optimization

[I-D.ietf-lsr-flex-algo] introduced a mechanism to do segment stack depth optimization for an SR policy in IGP domain part. As the color of SR policy defined a TE purpose, traditionally the headend or SDN controller will compute an expected TE path to meet that purpose.

It is necessary to map a color (32 bits) to an FA-id (8 bits) when SR flex-algorithm enabled for an SR policy. Besides that, it is necessary to enable the FA-id on each node that wants to join the

same FA plane manually. The FAD could copy the TE constraints (not including bandwidth case) contained in the color template.

We need to consider the cost of losing the flexibility of color when executing the flex-algo optimization, and also consider the gap between P2P TE requirements and MP2P SR FA LSP capability, to reach the right balance when deciding which SR policy need optimization.

10.1. Flex-algo Using AII Criteria

Because the first feature of AII is a TE criteria of link and node, it could be served as a parameter of Flex-algo Definition. [I-D.peng-lsr-flex-algo-opt-slicing] described how to extend IGP Flex-algo to compute constraint based paths over the AII specific network slice.

10.2. Best-effort Color Template Mapping to Flex-algo

As described above, for best-effort service within an AII specific virtual network, we have already constructed SR-BE FIB tables per AII, that is mostly like Flex-algo. Thus, it is not necessary to map to FA-id again for a color template which has defined a best-effort behavior within an AII specific virtual network. Of course, if someone forced to remap it, there is no downside for the operation, the overlay best-effort service (with a color which defined specific AII, best-effort requirement, and mapping FA-id) in IGP domain will try to recurse over <AII, prefix> or <FA-id, prefix> FIB entry.

10.3. Traffic Engineering Color Template Mapping to Flex-algo

An SR-TE tunnel/policy that served for traffic engineering service of an AII speficif virtual network was generated and computed according to the relevant color template, which contained specific AII and some other traditional TE constraints. If we config mapping FA-id under the color template, the SR-TE tunnel/policy instance will inherit forwarding information from corresponding SR Flex-Algo FIB entry. If we config merging FA-id under the color template, the SR-TE tunnel/ policy instance will have a TE path within Flex-algo plane.

11. Network Slicing Examples

In this section, we will further illustrate the point through some examples. All examples share the same figure below.

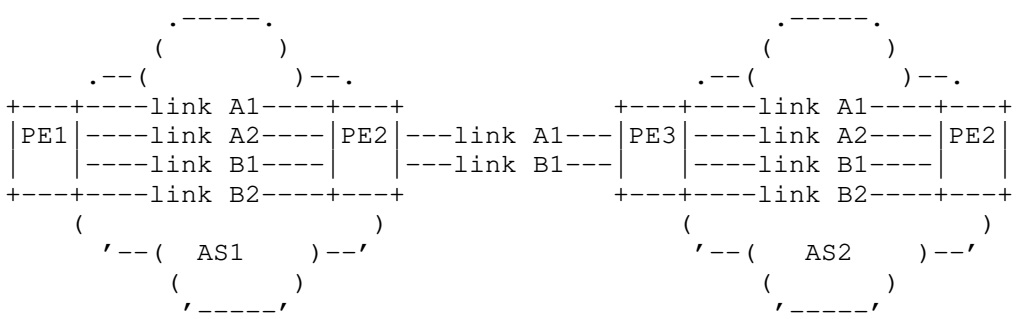


Figure 2 Network Slicing via AII

Suppose that each link belongs to separate virtual network, e.g., link Ax belongs to the virtual network colored by AII A, link Bx belongs to the virtual network colored by AII B. link x1 has an IGP metric smaller than link x2, but TE metric lager.

To simplify the use case, each AS just contained a single IGP area.

11.1. Intra-domain Network Slicing Example

11.1.1. Best-effort Service over Network Slice Example

From the perspective of node PE1 in AS1, it will calculate best-effort forwarding entry for each AII instance (including default AII) to destinations in the same IGP area. For example:

For <AII=0, destination=ASBR1> entry, forwarding information could be ECMP during link A1 and link B1, with destination node-SID 100 for <AII=0, destination=ASBR1>.

For <AII=A, destination=ASBR1> entry, forwarding information could be link A1, with destination node-SID 200 for <AII=A, destination=ASBR1>.

For <AII=B, destination=ASBR1> entry, forwarding information could be link B1, with destination node-SID 300 for <AII=B, destination=ASBR1>.

11.1.2. TE Service over Network Slice Example

It could also initiate an SR-TE instance (SR tunnel or SR policy) with the particular color template on PE1, PE1 is headend and ASBR1 is destination node. For example:

For SR-TE instance 1 with color template which defined criteria including {default AII, min TE metric}, forwarding information could be ECMP during two segment list {adjacency-SID 1002 for <AII=0, link A2> @PE1} and {adjacency-SID 1004 for <AII=0, link B2> @PE1}.

For SR-TE instance 2 with the color template which defined criteria including {AII=A, min TE metric}, forwarding information could be presented as the segment list {adjacency-SID 2002 for <AII=A, link A2> @PE1}.

For SR-TE instance 3 with the color template which defined criteria including {AII=B, min TE metric}, forwarding information could be presented as the segment list {adjacency-SID 3004 for <AII=B, link B2> @PE1}.

11.1.3. TE Service over Network Slice with Flex-algo Example

Furthermore, we can use SR Flex-algo to optimize the above SR-TE instance. For example, for SR-TE instance 1, we can define FA-ID 201 with FAD that contains the same information as the color template, in turn, FA-ID 202 for SR-TE instance 2, FA-ID 203 for SR-TE instance 3. Note that each FA-ID also needs to be enabled on ASBR1. So that the corresponding SR FA entry could be:

For <FA-ID=201, destination=ASBR1> entry, forwarding information could be ECMP during link A2 and link B2, with destination node-SID 600 for <FA-ID=201, destination=ASBR1>.

For <FA-ID=202, destination=ASBR1> entry, forwarding information could be link A2, with destination node-SID 700 for <FA-ID=202, destination=ASBR1>.

For <FA-ID=203, destination=ASBR1> entry, forwarding information could be link B2, with destination node-SID 800 for <FA-ID=203, destination=ASBR1>.

11.2. Inter-domain Network Slicing via BGP-LS Example

11.2.1. Best-effort Service Example

For SR-TE instance 4 with color template which defined criteria including {default AII, min IGP metric}, forwarding information could be segment list {node-SID 100 for <AII=0, destination=ASBR1> , adjacency-SID 1001 for <AII=0, link A1> @ASBR1, node-SID 400 for <AII=0, destination=PE2> }.

For SR-TE instance 5 with color template which defined criteria including {AII=A, min IGP metric}, forwarding information could be

segment list {node-SID 200 for <AII=A, destination=ASBR1> , adjacency-SID 1001 for <AII=A, link A1> @ASBR1, node-SID 500 for <AII=A, destination=PE2> }.

For SR-TE instance 6 with color template which defined criteria including {AII=B, min IGP metric}, forwarding information could be segment list {node-SID 300 for <AII=B, destination=ASBR1> , adjacency-SID 1003 for <AII=B, link B1> @ASBR1, node-SID 600 for <AII=B, destination=PE2> }.

11.2.2. TE Service Example

For SR-TE instance 7 with color template which defined criteria including {default AII, min TE metric}, forwarding information could be ECMP during two segment list {adjacency-SID 1002 for <AII=0, link A2> @PE1, adjacency-SID 1001 for <AII=0, link A1> @ASBR1, adjacency-SID 1002 for <AII=0, link A2> @ASBR2} and {adjacency-SID 1004 for <AII=0, link B2> @PE1, adjacency-SID 1003 for <AII=0, link B1> @ASBR1, adjacency-SID 1004 for <AII=0, link B2> @ASBR2}.

For SR-TE instance 8 with color template which defined criteria including {AII=A, min TE metric}, forwarding information could be segment list {adjacency-SID 2002 for <AII=A, link A2> @PE1, adjacency-SID 2001 for <AII=A, link A1> @ASBR1, adjacency-SID 2002 for <AII=A, link A2> @ASBR2}.

For SR-TE instance 9 with color template which defined criteria including {AII=B, min TE metric}, forwarding information could be segment list {adjacency-SID 3004 for <AII=B, link B2> @PE1, adjacency-SID 3003 for <AII=B, link B1> @ASBR1, adjacency-SID 3004 for <AII=B, link B2> @ASBR2}.

11.2.3. TE Service Using Flex-algo Example

For TE service, if we use SR Flex-algo to do optimizaztion, the above forwarding information of each TE instance could inherit the corresponding SR FA entry, it would look like this:

For SR-TE instance 7, forwarding information could be ECMP during two segment list {node-SID 600 for <FA-ID=201, destination=ASBR1> , adjacency-SID 1001 for <AII=0, link A1> @ASBR1, node-SID 600 for <FA-ID=201, destination=PE2> } and {adjacency-SID 1004 for <AII=0, link B2> @PE1, adjacency-SID 1003 for <AII=0, link B1> @ASBR1, adjacency-SID 1004 for <AII=0, link B2> @ASBR2}.

For SR-TE instance 8 with color template which defined criteria including {AII=A, min TE metric}, forwarding information could be segment list {node-SID 700 for <FA-ID=202, destination=ASBR1> ,

adjacency-SID 2001 for <AII=A, link A1> @ASBR1, node-SID 700 for <FA-ID=202, destination=PE2> }.

For SR-TE instance 9 with color template which defined criteria including {AII=B, min TE metric}, forwarding information could be segment list {node-SID 800 for <FA-ID=203, destination=ASBR1> , adjacency-SID 3003 for <AII=B, link B1> @ASBR1, node-SID 800 for <FA-ID=203, destination=PE2> }.

11.3. Inter-domain Network Slicing via BGP-LU Example

In figure 1, PE2 can allocate and advertise six labels for its loopback plus color 1, 2, 3, 4, 5, 6 respectively. Suppose color 1 defines {default AII, min IGP metric}, color 2 defines {AII=A, min IGP metric}, color 3 defines {AII=B, min IGP metric}, and color 4 defines {default AII, min TE metric}, color 5 defines {AII=A, min TE metric}, color 6 defines {AII=B, min TE metric}. PE2 will advertise these labels to ASBR2 and ASBR2 then continues to allocate six labels each for prefix PE2 plus different color. Other nodes will have the same operation. Ultimately PE1 will maintain six BGP-LU LSP.

For example, the BGP-LU LSP for color 1 will be over SR best-effort FIB entry node-SID 100 for <AII=0, destination=ASBR1> to pass through AS1, over adjacency-SID 1001 for <AII=0, link A1>@ASBR1 to pass inter-AS, over SR best-effort FIB entry node-SID 400 for <AII=0, destination=PE2> to pass through AS2.

For example, The BGP-LU LSP for color 4 will over SR-TE instance 1 (see section 10.1.2), or SR best-effort FIB entry node-SID 600 for <FA-id=201, destination=ASBR1> (see section 10.1.3) to pass through AS1, over adjacency-SID 1001 for <AII=0, link A1>@ASBR1 to pass inter-AS, over SR-TE instance 1' or corresponding SR FA entry to pass through AS2. Note that ASBR1 need also understand the meaning of a specific color and select forwarding resource between two AS.

12. Implementation Suggestions

The implementation cost is low by means of existing segment routing infrastructure.

12.1. SR-MPLS

As a node often contains control plane and forwarding plane, a suggestion is that only default AII specific FTN table, i.e, traditional FTN table, need be installed on forwarding plane, so that there are not any modification and upgrade requirement for hardware and existing MPLS forwarding mechanism. FTN entry for non-default AII instance will only be maintained on the control plane and be used

for overlay service iteration according to next-hop plus color (color will give AII information and mapping FA-id information). Note that ILM entry for all AII need be installed on forwarding plane, that does not bring any confusion because of prefix-SID allocation per AII.

SR NHLFE entry and other iteration entry such as <next-hop, color> can contain AII information for expected packet scheduling. It is recommended that QoS policy per AII should be maintained in the underlay network. The Slice Type value of AII can distinguish flows by coarse-grained classification, while the Instance value of AII can be used for more scheduling policy.

12.2. SRv6

For SRv6 case, IPv6 address resource is directly used to represent SID, so that different IPv6 block could be allocated to different slice. There are two possible ways to advertise slice specfic IPv6 block:

- Traditional prefix reachability, but only for default AII (0) specific IPv6 block.
- New SRv6 Locator advertisement, for nonzero AII specific IPv6 block.

Forwarding entries for the default AII specific locators advertised in prefix reachability MUST be installed in the forwarding plane of receiving routers.

Forwarding entries for the nonzero AII specific locators advertised in the SRv6 Locator MUST be also installed in the forwarding plane of receiving SRv6 capable routers when the associated AII is supported by the receiving node.

The entries of both the above two cases SHOULD be installed in the unified FIB table, i.e., a single FIB table for default AII, because different IPv6 block is allocated to different slice. Instead, more FIB tables created for each VN in dataplane will bring comlexity for overlay service iteration, that is why MTR has no practical deployment.

The forwarding information of FIB entry can contain AII information for expected packet scheduling.

13. IANA Considerations

This document requests IANA to create a new top-level registry called "Network Slicing Parameters". This registry is being defined to serve as a top-level registry for keeping all other Network Slicing sub-registries.

Additionally, a new sub-registry "AII (TN-slice Identifier) codepoint" is to be created under top-level "Network Slicing Parameters" registry. This sub-registry maintains 32-bit identifiers and has the following registrations:

Slice Type (High 8bits)	Instance (Low 24bits)	Description
0(Normal) endogenous: IGP-metric	0	Reserved for Default Slice: the original physical network.
	nonzero	Normal Slice, for user defined.
1(uRLLC) endogenous: delay	0	Resevered.
	nonzero	Slice suitable for the handling of ultra- reliable low latency communications, for user defined.
2(TE) endogenous: TE-metric	0	Resevered.
	nonzero	General TE Slice, for user defined.
3(eMBB) endogenous: TBD	0	Resevered.
	nonzero	Slice suitable for the handling of 5G enhanced Mobile Broadband, for user defined.
4(MIoT) endogenous: TBD	0	Resevered.
	nonzero	Slice suitable for the handling of massive IoT, for user defined.
5(V2X) endogenous: TBD	0	Resevered.
	nonzero	Slice suitable for the handling of V2X services, for user defined.
6-255	any	Unassigned.

Table 1. AII Codepoint

14. Security Considerations

TBD.

15. Acknowledgements

TBD.

16. Normative references

[I-D.ali-spring-network-slicing-building-blocks]
Ali, Z., Filsfils, C., Camarillo, P., and D. Voyer, "Building blocks for Slicing in Segment Routing Network", draft-ali-spring-network-slicing-building-blocks-02 (work in progress), November 2019.

[I-D.ietf-idr-bgpls-inter-as-topology-ext]
Wang, A., Chen, H., Talaulikar, K., and S. Zhuang, "BGP-LS Extension for Inter-AS Topology Retrieval", draft-ietf-idr-bgpls-inter-as-topology-ext-09 (work in progress), September 2020.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-19 (work in progress), September 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-12 (work in progress), October 2020.

[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-08 (work in progress), July 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, d., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-03 (work in progress), September 2020.

[I-D.nsdt-teas-transport-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "IETF Definition of Transport Slice", draft-nsdt-teas-transport-slice-definition-04 (work in progress), September 2020.

[I-D.peng-lsr-flex-algo-opt-slicing]
Peng, S., Chen, R., and G. Mirsky, "IGP Flexible Algorithm Optimazition for Netwrok Slicing", draft-peng-lsr-flex-algo-opt-slicing-02 (work in progress), September 2020.

[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>.

[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>.

[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>.

[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>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <https://www.rfc-editor.org/info/rfc7308>.

[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>.

[RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, DOI 10.17487/RFC7810, May 2016, <https://www.rfc-editor.org/info/rfc7810>.

[RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <https://www.rfc-editor.org/info/rfc7911>.

[RFC8668] Ginsberg, L., Ed., Bashandy, A., Filsfils, C., Nanduri, M., and E. Aries, "Advertising Layer 2 Bundle Member Link Attributes in IS-IS", RFC 8668, DOI 10.17487/RFC8668, December 2019, <https://www.rfc-editor.org/info/rfc8668>.

Authors' Addresses

Shaofu Peng
ZTE Corporation

Email: peng.shaofu@zte.com.cn

Ran Chen
ZTE Corporation

Email: chen.ran@zte.com.cn

Gregory Mirsky
ZTE Corporation

Email: gregimirsky@gmail.com

Fengwei Qin
China Mobile

Email: qinfengwei@chinamobile.com

Individual
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

R. Rokui
Nokia
S. Homma
NTT
X. de Foy
InterDigital Inc.
LM. Contreras
Telefonica
P. Eardley
BT
K. Makhijani
Futurewei Networks
H. Flinck
Nokia
R. Schatzmayr
Deutsche Telekom
A. Tizghadam
TELUS Communications Inc
C. Janz
H. Yu
Huawei Canada

November 2, 2020

IETF Network Slice for 5G and its characteristics
draft-rokui-5g-ietf-network-slice-00

Abstract

5G Network slicing is an approach to provide separate independent end-to-end logical network from User Equipment (UE) to various mobile applications where each network slice has its own Service Level Agreement (SLA) and Objectives (SLO) requirements. Each end-to-end network slice consists of a multitude of contexts across RAN, Core and transport domains each with its own controller. To provide automation, assurance and optimization of the 5G the network slices, a 5G E2E network slice orchestrator is needed which interacts with controllers in RAN, Core and Transport network domains. The interfaces between the 5G E2E network slice orchestrator and RAN and Core controllers are defined in various 3GPP technical specifications. However, 3GPP has not defined a similar interface for transport network.

The aim of this document is to describe E2E network slicing and its relation to "IETF network slice" for 5G use-case. It also provides an information model for control and mangement of IETF network slices for 5G.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Network slicing offers network operators a mechanism to allocate dedicated infrastructure, resources and services from a shared operator's network to a customer for specific use-case. As discussed in draft [I-D.nsdt-teas-ietf-network-slice-definition], there are a number of use-cases benefiting from network slicing including:

- 5G network slicing (See [TS.23.501-3GPP])
- Network wholesale services
- Network sharing among operators
- NFV connectivity and Data Center Interconnect

It is important to note that the concept of network slicing is not only limited to 5G but other use-cases can also benefit from it as shown above. However, the 5G use case is one of the important use cases for network slicing. This memo will discuss the 5G use-case in more details. In specific, 5G network slicing is a mechanism which a mobile network operator can use to allocate dedicated infrastructures, resources and services from a shared mobile and transport network to a 5G customer for specific 5G use-case.

A 5G network slice is inherently an E2E concept and is composed of multiple logical independent networks in a common operator's network from a user equipment to various 5G applications. In particular, 5G network slicing receives attention due to factors such as diversity of services and devices in 5G each with its own SLA requirements. Each 5G E2E network slice consists of multiple 5G RAN, 5G Core and transport network domains each with its own controller (See [TS.28.531-3GPP].

To enable automation, assurance and optimization of 5G network slices, an E2E network slice orchestrator is needed which interacts with 5G RAN, 5G Core and Transport network controllers. The interfaces between the E2E network slice orchestrator and RAN and Core slice controllers are defined in various 3GPP technical specifications. However, 3GPP has not defined the same interface towards the transport network. Draft [I-D.wd-teas-transport-slice-yang] addresses the object model of such interface for all network slice use-cases. However, for 5G network

slicing, the current model shall be augmented to address the specific characteristics of the 5G network slices. The aim of this document is to provide characteristics of 5G network slices and how it relares to "IETF network slice". It also provides the IETF network slice interface specifications and its information model to be used for automation, monitoring and optimization of IETF network slices for 5G. See [I-D.contreras-teas-slice-nbi].

1.1. Definition of Terms

Please refer to [I-D.nsdt-teas-ietf-network-slice-definition] and [I-D.homma-slice-provision-models] as well.

Tenant: Also known as Customer. A network slice tenant is a person or group that rents and occupies an instance of the network slice from network provider.

5G End-to-end Network Slice: A logical end-to-end network provided by a 5G network slice provider that has the functionality and performance to support a specific 5G service. It spans multiple network domains (e.g. radio, transport and core) and in some cases more than one administrative domain. It may well support dynamic modification or it might be long-lasting i.e. only change on commercial timescales.

5G IETF Network Slice: We will use the term "5G IETF network slice" throughout this draft. It simply refers to IETF network slice define in [I-D.nsdt-teas-ietf-network-slice-definition] applicable to 5G.

RAN Slice: Also known in 3GPP as RAN Sub-Slice or RAN Slice-Subnet. The context and personality created on RAN network functions to address the 5G radio portion of a 5G E2E network slice.

Core Slice: Also known in 3GPP as Core Sub-Slice or Core Slice-Subnet. The context and personality created on Core network functions to address the 5G Core portion of a 5G E2E network slice.

S-NSSAI: Single-Network Slice Selection Assistance Information, defined by 3GPP which is the identification of a 5G E2E Network Slice

gNB: The radio portion of a 5G E2E network slice and in a distributed radio deployment (called Cloud-RAN), it incorporates two major modules; Central Unit (CU) and Distribute Unit (DU)

DU: Distributed Unit: This logical unit includes a subset of gNB real-time functions. Its operation is controlled by the CU.

CU: Central Unit: It is a logical unit that includes the gNB non-realtime functions.

UE: User Equipment such as vehicle infotainment unit, cell phone, IoT sensor and etc.

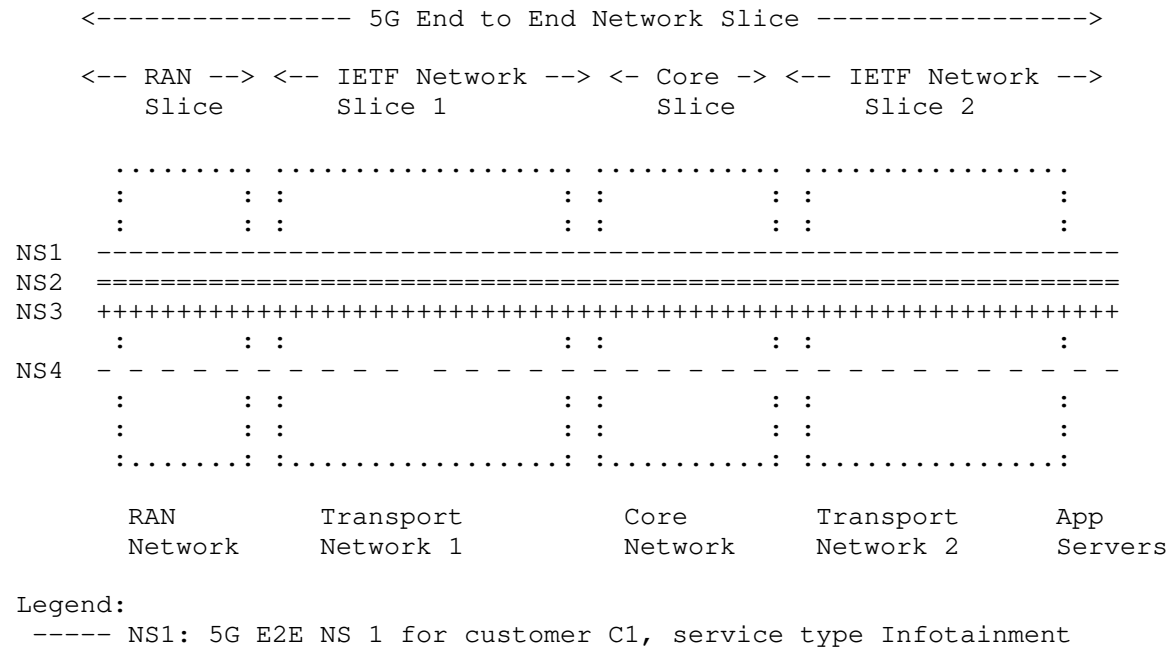
RAN: Radio Access Network is the part of a mobile system that connects individual devices to other parts of a network through radio connections. It provides connection between user equipment (UE) and mobile core network.

Transport Domain: Transport domain is a network domain implemented by the deployment of IETF network technologies.

2. Architecture of a 5G end-to-end network slice

To demonstrate the concept of 5G E2E network slice and the role of various controllers, consider a typical 5G network shown in Figure 1 where a mobile network operator Y has two customers C1 and C2. The boundaries of administrative domain of the operator includes RAN, transport, Core and mobile application domains. Customer C1 and C2 request to have one or more logical independent E2E networks from UEs (e.g. vehicle infotainment, mobile phone, IoT meters etc.) to the 5G application servers, each with its own distinct SLO.

Each of these independent networks is called a 5G E2E network slice. Each E2E network slice comprised of three componets RAN slice, IETF network slice, and Core slice, respectively representing RAN, transport and core domain portions of the slice.



Legend:
----- NS1: 5G E2E NS 1 for customer C1, service type Infotainment
===== NS2: 5G E2E NS 2 for customer C1, service type Autonomous Driving
+++++ NS3: 5G E2E NS 3 for customer C1, service type HD Map
- - - NS4: 5G E2E NS 4 for customer C2, service type CCTV

Figure 1: High level architecture of a 5G end-to-end network slice

In Figure 1 mobile network operator Y has created four 5G E2E network slices, NS1, NS2, NS3 and NS4, each with its own RAN, Core and IETF network slices. To create a RAN slice, the RAN network function s such as eNB and gNB should be programmed to have a context for each 5G E2E network slice. This context means that the RAN network functions should allocate certain resources for each 5G E2E network slice they belong to such as air interface, various schedulers, policies and profiles to guarantee the SLO requirement for that specific network slice. By the same token, the Core slices will be created which means that the mobile network operator will create the context for each 5G E2E network slice on Core network functions.

For each 5G E2E network slice NS1, NS2, NS3 and NS4, after creation of RAN and Core slices, they should be connected to each other and be connected to mobile application servers to form the 5G E2E network slice. As defined in [I-D.nsdt-teas-ietf-network-slice-definition], the set of connections are referred to "IETF Network Slices" and specifically for 5G they are referred to "5G IETF Network Slices".

Referring to Figure 1, for each 5G E2E network slice, the following 5G IETF network Slices are needed:

- 5G IETF Network Slice 1: To connect RAN slice to Core slice in Transport Network 1

- 5G IETF Network Slice 2: To connect Core slice to Mobile Application Servers in Transport Network 2. This might be needed if the mobile application servers are connected to core network functions through a transport network. For example, if the Core slice, which is realized on VNFs, and mobile application servers are in the same data center, the 5G IETF Network Slice 2 is not needed. In this case the transport network 2 does not exist.

Note that as we will see later in Section 4.1, Section 4.2 and Section 4.3, the number of "5G IETF network slices" might be more than two which depends on some factors such as RAN deployment:

After creation of RAN, Core and 5G IETF network slices, they will be associated together to form a working 5G E2E network slice identified by an ID referred as to S-NSSAI. Please refer to [TS.23.501-3GPP] for more info on S-NSSAI.

To support fully automated enablement and assurance of 5G E2E network slices, multiple controllers are needed to perform the life cycle of 5G E2E network slices in RAN, Core and Transport domains. As shown in Figure 2 each RAN, Core and Transport domain needs its own controller called RAN Slice Controller, Core Slice Controller and IETF Network Slice Controller. In addition, an E2E network slice orchestrator is needed to provide coordination and control of network slices from an E2E perspective.

In summary, a 5G E2E network slice will involve several domains, each with its own controller; 5G RAN, 5G Core and transport domains need to be coordinated in order to deliver an E2E mobile service. Note that in this context a service is not an IP/MPLS service but rather customer facing services such as CCTV service, eMBB service, Infotainment service and so on.

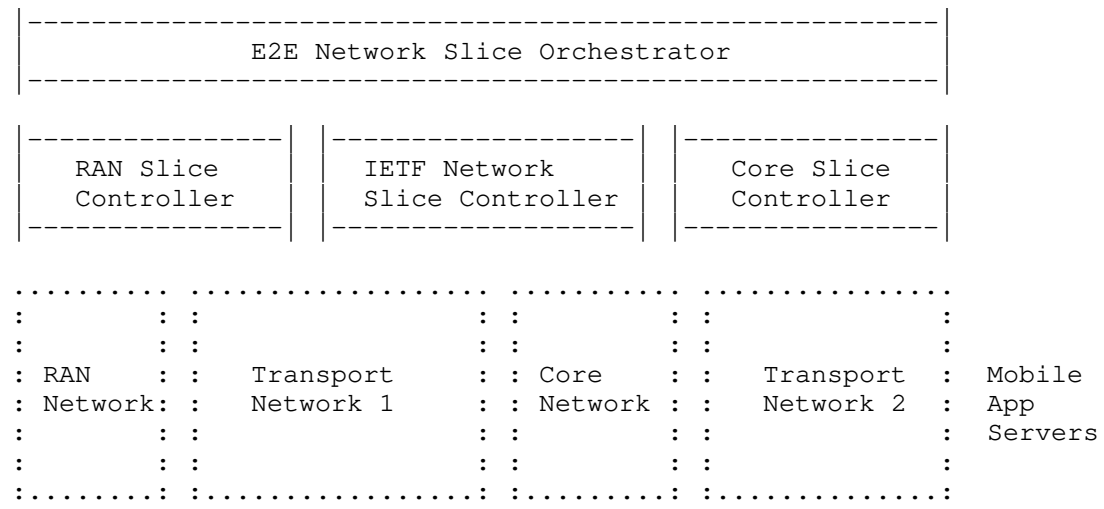


Figure 2: Various controllers for 5G end-to-end network slice

3. Typical flow for fulfillment of a 5G E2E network slice

Figure 3 provides a typical flow across various controllers, orchestrator, NFVO and RAN/Transport/Core networks to achieve the automatic creation of a 5G E2E network slices such as NS1, NS2, NS3 or NS4 shown in Figure 1. Below are typical steps from the time a customer sends its request for a 5G E2E network slice creation to the operators network until the network slice is created and ready to be used by the customer. It is important to note that in practice some of these steps can be combined or re-ordered.

1. The customer C1 requests, from operator Y, the creation of a 5G E2E network slice NS1 for Infotainment service type and SLO of 10 [Mbps]

2. The 5G E2E network slice orchestrator receives this request and using its pre-defined network slice blueprints (a.k.a. network slice templates), creates a network slice profile (a.k.a. network slice instance) containing all the network functions in RAN and core which should be part of this E2E network slice. It then goes through decomposition of this profile and triggers various actions towards RAN, Core and transport domains.

3. A request for creation of 5G RAN slice will be sent to RAN Slice Controller.

4. If new instances of virtual RAN network functions are needed, the RAN Slice Controller triggers the creation of new VNFs in RAN (using for example ETSI interface Os-Ma-nfvo)

5. NFVO manages the life cycle of virtual RAN network functions

6. Since both physical and virtual RAN network functions which are part of 5G E2E network functions are known to the RAN slice controller, it triggers the creation of a RAN slice by programming 5G RAN network functions

7. Similary to previous step (3), a request for creation of a Core slice will be sent to the Core Slice Controller.

8. If new instances of virtual Core network functions are needed, the Core Slice controller triggers the creation of new 5G core VNFs (using for example ETSI interface Os-Ma-nfvo) and NFVO manages the life cycle of virtual Core network functions

9. Since both physical and virtual 5G Core network functions as components of 5G E2E network functions are known to the Core slice controller, it triggers the creation of Core slice by programming 5G Core network functions

10. In this step, the creation of various 5G IETF network slices will be triggered. Each 5G IETF network slice contains one or more connections between RAN network functions, Core network functions and 5G mobile applications. For example, connectivity between 5G RAN and 5G Core slices, connectivity between 5G RAN network functions (such as DU to CU) or connectivity between 5G core slice and mobile applications. Note that this step can be triggered by E2E network slice orchestrator, RAN slice controller, Core slice controller, or a combination of them.

11. [Optional] If the realization of a 5G IETF network slice involves creation of new VNFs (e.g. Firewall, security gateway etc.), 5G IETF network slice controller triggers the creation of those VNFs (using for example ETSI interface Os-Ma-nfvo)

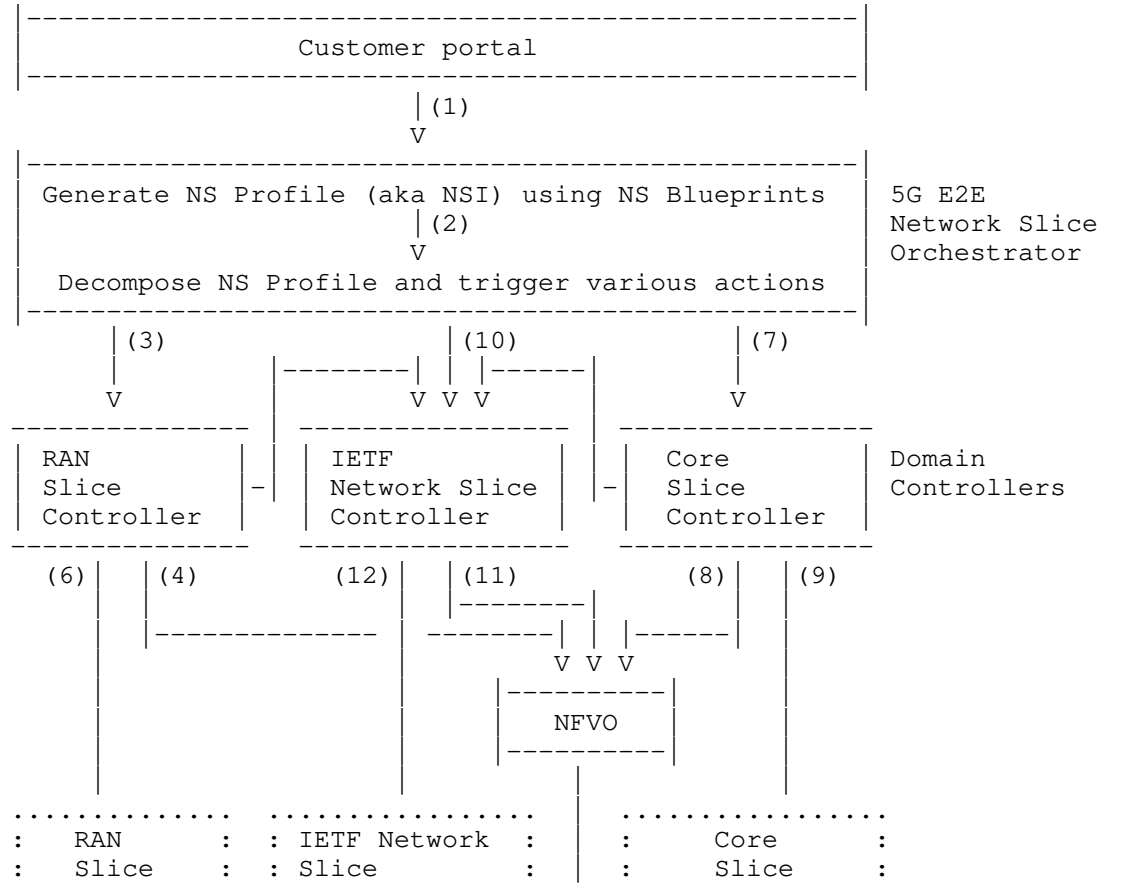
12. Various 5G IETF network slices will be realized in transport network. Note that interface (10) is technology-agnostic whereas interface (12) is technology-specific

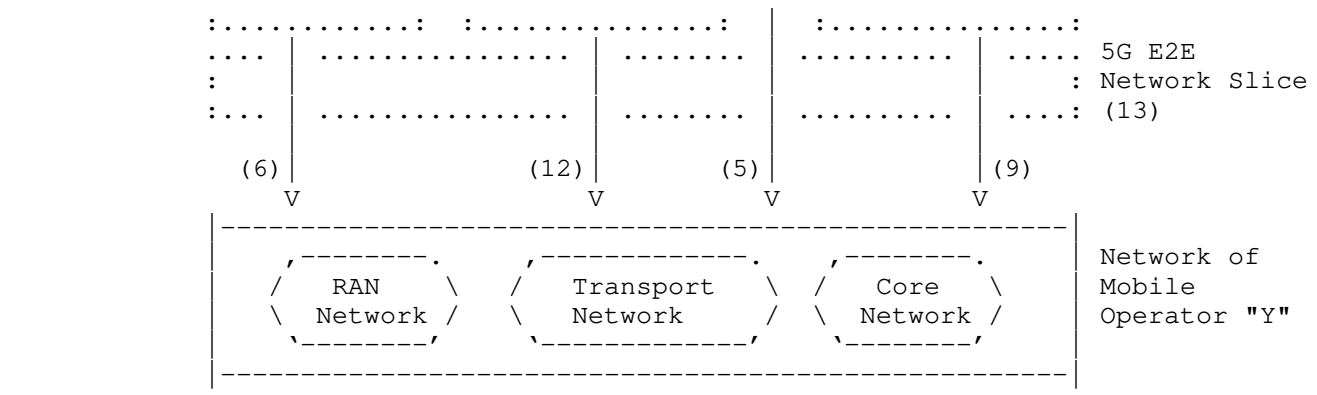
13. The E2E network slice orchestrator associates RAN slice, Core slice and 5G IETF network slices together to form a single 5G E2E network slice NS1

14. At the end, when the E2E network slice is created, the E2E network slice orchestrator will allocate a unique network slice id (called S-NSSAI) and eventually, during the UE network attach, UEs will be informed about the existence of this newly created E2E network slice. UEs can request it using the 3GPP 5G signaling procedures.

Note that the interfaces 3 and 7 between 5G E2E network slice orchestrator and RAN and Core slice controllers along with their information models are defined in various 3GPP technical specifications. However, 3GPP has not defined the same interface for transport network (i.e. interface 10).

The aim of this document is to define specific attributes related to 5G network slices which will be used later to augment [I-D.wd-teas-transport-slice-yang].





Legend:
NS: 5G E2E Network Slice
NSI: Network Slice Instance
NFVO: NFV Orchestrator

Figure 3: Typical flow for fulfillment of a 5G E2E network slice

4. Definition of 5G IETF Network Slice

Referring to [I-D.nsdt-teas-ietf-network-slice-definition], the IETF network slice is define as follows:

An IETF network slice is a logical network topology connecting a number of endpoints with a set of shared or dedicated network resources. These resources are used to satisfy specific Service Level Objectives (SLOs).

The 5G IETF Network slice specification is technology-agnostic. Using the above-mentioned definition, the 5G IETF network slice is define as follows:

5G IETF network slices are sets of connections among the following network functions and mobile applications:

- 5G RAN slice and 5G Core slice
- 5G Core slice and mobile application server
- Among 5G RAN network functions DU to CU
- Among 5G RAN network functions RU to DU

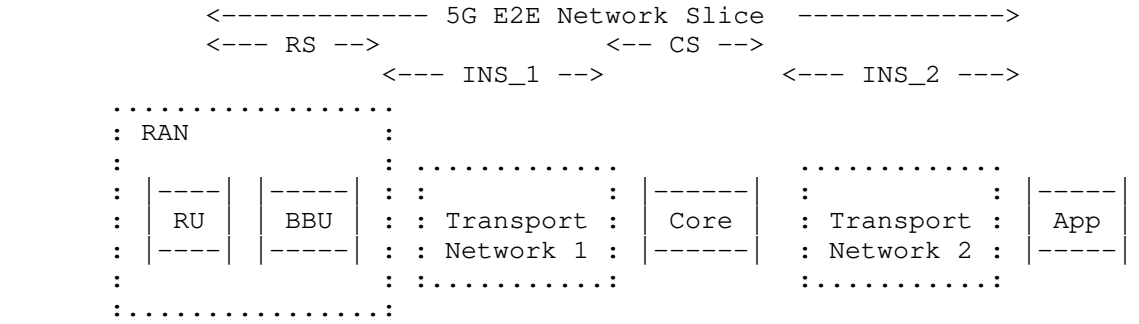
In Section 4.1, Section 4.2 and Section 4.3, the details of various 5G IETF network slices for following RAN deployment will be provided:

- Distributed RAN
- Centralized RAN
- Cloud RAN (C-RAN)

4.1. 5G IETF Network Slices in Distributed RAN deployment

Distributed RAN is the most common deployment of 4G and 5G RAN networks whereas shown in Figure 4, the RAN network is connected to Core network using the transport network 1. Optionally the mobile applications can be also connected to the Core network using another transport network 2.

In this case, a single 5G E2E network slice contains not only 5G RAN and 5G Core slices but two 5G IETF network slices INS_1 and INS_2. INS_1 connects the RAN slice to Core slice and INS_2 connects Core slice to mobile application servers (if needed).



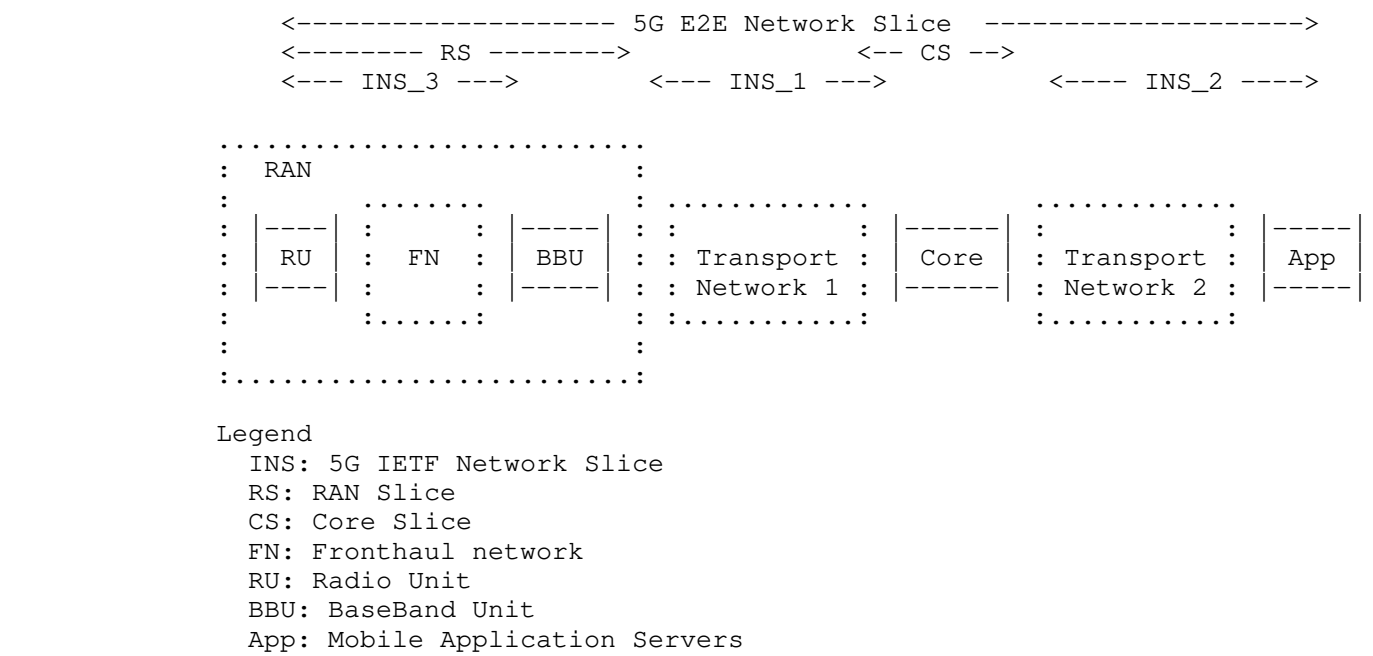
Legend
INS: 5G IETF Network Slice
RS: RAN Slice
CS: Core Slice
RU: Radio Unit
BBU: BaseBand Unit
App: Mobile Application Servers

Figure 4: 5G IETF network slices in distributed RAN deployment

4.2. 5G IETF Network Slices in Centralized RAN deployment

The RAN consists of two functional units: the baseband unit (BBU) and the radio unit (RU). The BBU processes the signal and is connected to the transport network. The RU transmits and receives the carrier signal that is transmitted over the air to the end user equipment (UE). In Centralized RAN as depicted in Figure 5, the RU and BBU are separated by a network called fronthaul network.

In this deployment a single 5G E2E network slice contains not only 5G RAN and 5G Core slices but three 5G IETF network slices INS_1, INS_2 and INS_3 where INS_1 and INS_2 are identical to their counterparts in distributed RAN deployment case (see Figure 4) and a new INS_3 connects the Radio Units (RU) to the BBUs.



Legend
- INS: 5G IETF Network Slice
- RS: RAN Slice
- CS: Core Slice
- FN: Fronthaul network
- RU: Radio Unit
- BBU: BaseBand Unit
- App: Mobile Application Servers

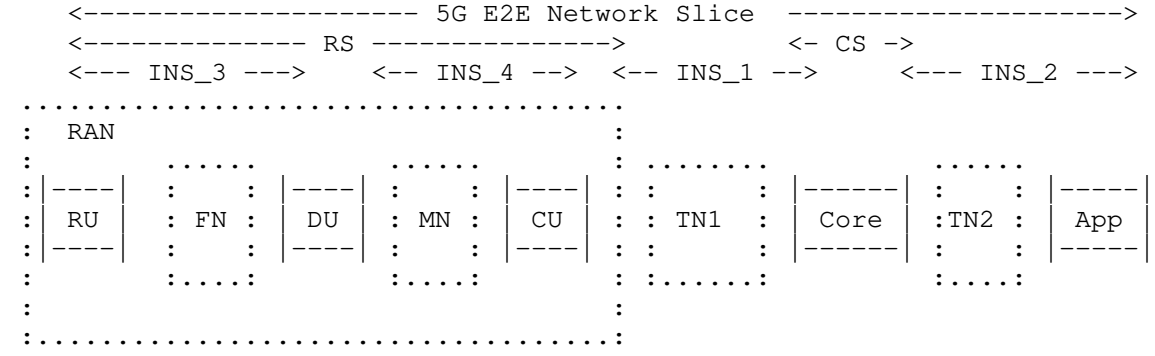
Figure 5: 5G IETF network slices in Centralized RAN deployment

4.3. 5G IETF Network Slices in Cloud RAN (C-RAN)

In Cloud-RAN deployment, the baseband unit (BBU) is further disaggregated into real-time and non-real-time components. The former is deployed close to antenna to manages the real-time air interface resources while the non-real-time control functions are

hosted centrally in the cloud. In 5G, BBU is split into two parts called CU (Central Unit) and DU (Distributed Unit) as shown in Figure 6 where these entities are connected by a new network called Midhaul network.

In this deployment a single 5G E2E network slice contains not only 5G RAN and 5G Core slices but four 5G IETF network slices INS_1, INS_2, INS_3 and INS_4 where INS_1, INS_2 and INS_3 are identical to their counterparts in centralized RAN deployment case (see Figure 5) and a new 5G IETF network slice INS_4 connects the DUs to CUs.



Legend
INS: 5G IETF Network Slice
RS: RAN Slice
CS: Core Slice
FN: Fronthaul network
MN: Midhaul network
TN: Transport network
DU: Distributed Unit
CU: Central Unit
RU: Radio Unit
App: Mobile Application Servers

Figure 6: 5G IETF network slices in Cloud RAN deployment

4.4. 5G IETF Network Slice as a set of Connection Groups

As discussed in [I-D.nsdt-teas-ietf-network-slice-definition], an IETF network slice contains one or more connections between various network functions, application and devices. These connections can be grouped into various Connection Groups where each group has its own SLA/SLO.

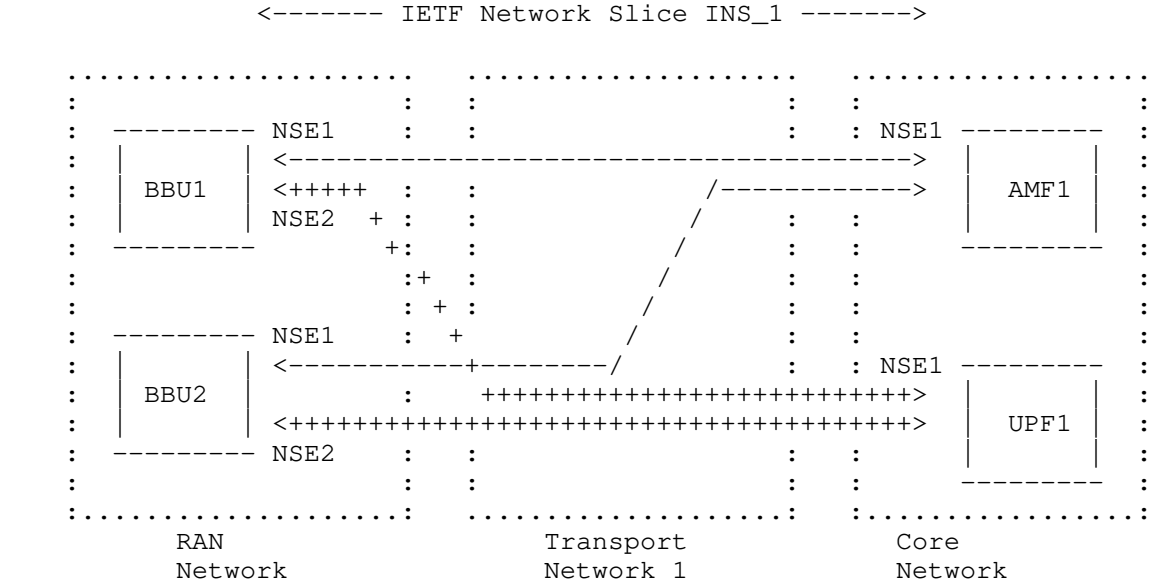
To further explore this concept in 5G E2E network slicing, consider Figure 7, where the details of 5G IETF network slice INS_1 introduced in Figure 4 is illustrated. The 5G IETF network slice INS_1 is between 5G RAN and Core slices and has multiple connections between 5G RAN network functions BBU1 and BBU2 and 5G Core network functions AMF1 and UPF1. In particular, it contains the following connection groups, each with its own SLO where SLO-C and SLO-U might be different (e.g. they might be control and user plans SLOs):

- "Connection group C" connects BBU1 and BBU2 to AMF1 with SLO-C

- "Connection group U" connects BBU1 and BBU2 to UPF1 with SLO-U

The combination of two connection groups will form the 5G IETF network slice INS_1. Note that the definition of 5G IETF network slice INS_1 does not specify how these connections should be realized in transport network 1. Although it is optionally possible, it is not necessarily mandatory for the definition of a 5G IETF network slice to state which technology (e.g. IP, MPLS, Optics, PON etc.) or tunnel type (e.g. RSVP-TE, SR-TE etc.) should be employed for realization. As discussed in [[I-D.nsdt-teas-ietf-network-slice-definition], any of these technologies may be used by the IETF Network Slice Controller (NSC) to realize an IETF network slice.

In summary, a 5G IETF network slice is a distinct set of technology-agnostic connection groups between various 5G network functions, 5G devices or 5G applications each with its own deterministic SLO which can be realized by any suitable technology, tunnel type and service type.



5G IETF Network Slice INS_1:
{"Connection group C" + "Connection group U"}
Connection Group C {from BBU1.NSE1, BBU2.NSE1 to AMF1.NSE1 with SLO-C}
Connection Group U {from BBU1.NSE2, BBU2.NSE2 to UPF1.NSE1 with SLO-U}

Legend
BBU: BaseBand Unit
AMF: Access and Mobility Management Function
UPF: User Plane Function
NSE: 5G IETF Network Slice Endpoint
---- Connection group C
++++ Connection group U

Figure 7: Details of 5G IETF Network Slice as a set of Connection Groups

5. IETF Network Slice Controller NBI for 5G

As discussed in [I-D.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework], to fulfill (i.e. create, modify, delete) any IETF network slice and perform monitoring on it, an entity called IETF Network Slice Controller (NSC) is required to take abstract requests for 5G IETF network slices and realize them using suitable underlying technologies. An IETF Network Slice Controller is the key building block for control and management of the transport

slice. It provides the creation/modification/deletion, monitoring and optimization of transport Slices in a multi-domain, a multi-technology and multi-vendor environment.

Figure 8 shows the NSC and its NBI interface for 5G. Draft [I-D.wd-teas-transport-slice-yang] addresses the base data model of the NSC NBI interface for all network slicing use-cases. However, for 5G network slicing, the current model shall be augmented to include the specific characteristics of the 5G network slices for this interface. The details of NSC NBI interface for 5G provided in Section 6.

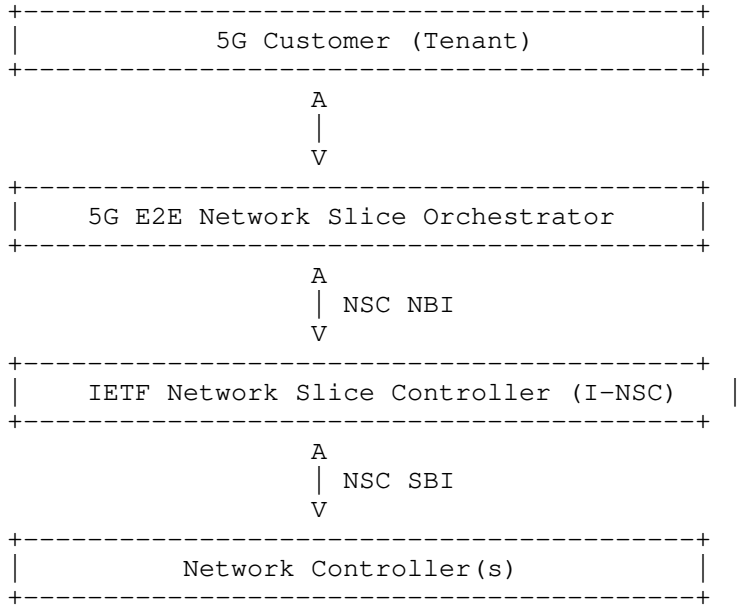


Figure 8: IETF Network Slice Controller NBI for 5G

5.1. Relationship between 5G IETF Network Slice NBI and various IETF data models

As discussed in [I-D.nsdt-teas-ns-framework], the main task of the IETF Network Slice Controller is to map abstract IETF network slice requirements from NBI to concrete technologies on SBI and establish the required connectivity, and ensure that required resources are allocated to IETF network slice. There are a number of different technologies that can be used on SBI including physical connections, MPLS, TSN, Flex-E, PON etc. If the undelay technology is IP/MPLS/

Optics, any IETF models can be used during the realization of IETF network slice.

There are no specific mapping requirements for 5G. The only difference is that in case of 5G, the NBI interface contains additional 5G specific attributes such as customer name, mobile service type, 5G E2E network slice ID (i.e. S-NSSAI) and so on (See Section 6). These 5G specific attributes can be employed by IETF Network Slice Controller during the realization of 5G IETF network slices on how to map NBI to SBI. They can also be used for assurance of 5G IETF network slices. Figure 9 shows the mapping between NBI to SBI for 5G IETF network slices.

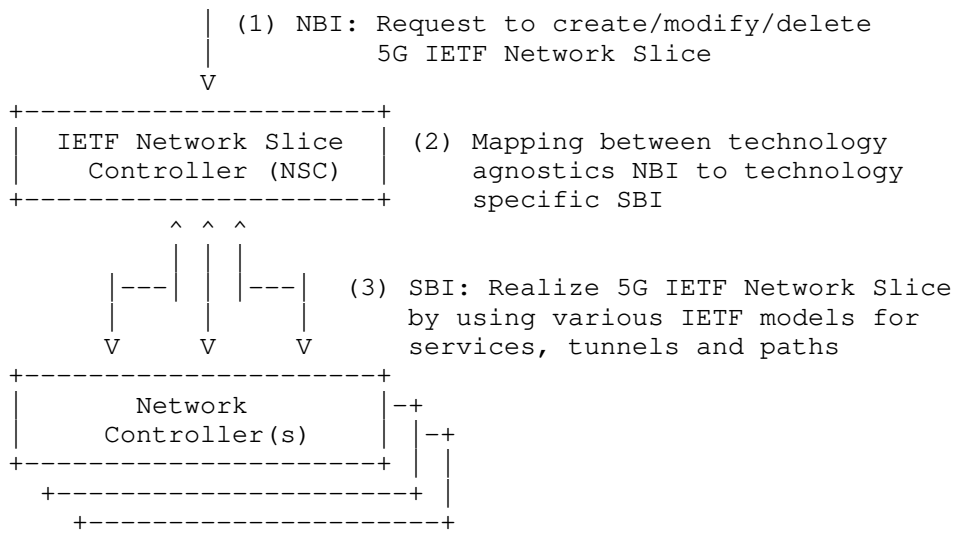


Figure 9: Relationship between transport slice interface and IETF Service/Tunnels/Path data models

6. 5G IETF Network Slice NBI Information Model

Based on the definition of 5G IETF Network slices (see Section 4), the high-level information model of northbound interface of IETF Network Slice Controller (NSC) for 5G IETF network slices should conform with Figure 10:

```
module: 5g-ietf-network-slices
  +--rw 5g-ietf-network-slice
      +--rw 5g-ietf-network-slice-info
```

```
        +--rw ins-id
        +--rw ins-name
        +--rw ins-plmn
        +--rw ins-hierarchical-tenant-id
     +--rw 5g-network-slice-info [s-nssai]
        +--rw s-nssai (i.e. 5G E2E network slice id)
        +--rw 5g-customer (i.e. 5G tenant)
        +--rw 5g-mobile-service-type (e.g. CCTV, infotainment etc)
     +--rw 5g-connection-group* [connection-group-id ]
        +--rw connection-group-id
        +--rw connection-group-name
        +--rw connection-group-type  (e.g., P2P, MP2MP, etc.)
        +--rw connection-group-status
           +-- admin-status
           +-- operational-status
        +--rw connection-group-member* [member-id]
           +--rw member-id
           +--rw member-name
           +--rw member //Ref. to 5G-ietf-connection-group-member
           +--ro member-slo-monitoring
             +--ro latency?
             +--ro jitter?
             +--ro loss?
        +--rw connection-group-slo-policy
           +--rw policy-id
           +--rw slo attributes
        +--rw connection-group-realization-policy //Optional
           +--rw policy-id
           +--rw realization-attributes //Optional
                                        //Technology-specific attributes
        +--rw connection-group-monitoring-policy //Optional
           +--rw policy-id
           +--rw monitoring-attributes //Optional. Such as if monitoring
                                       //is needed, frequency of
                                       //monitoring and how often send
                                       //them to NBI etc.
  +--rw 5g-ietf-network-slice-endpoint* [ep-id]
     +--rw ep-id
     +--rw ep-name
     +--rw domain-id
     +--rw node-id
     +--rw transport-port-id
     +--rw transport-vlan-id
     +--rw transport-id (e.g. IP address of the transport)
     +--rw transport-label (For future use)
     +--rw transport-bsid (For future use)
   +--rw 5G-ietf-connection-group-member* [member-id]
     +--rw member-id
```

```
   +--rw member-endpoint-a  //Ref. to 5g-ietf-network-slice-endpoint
   +--rw member-endpoint-b  //Ref. to 5g-ietf-network-slice-endpoint
```

Figure 10: Information model of NSC NBI interface for 5G IETF Network Slices

The proposed information model should include the following building blocks:

- 5g-ietf-network-slice-info: All attributes related to 5G IETF Network Slice. It contains information such as 5G IETF network slice name, 5G IETF network slice ID, PLMN and hierarchical tenant ID etc.

- 5g-network-slice-info: A list of all E2E network slices mapped to this 5G IETF network slice. As discussed in Section 3, a request for creation of a 5G IETF network slice is sent from 5G E2E network slice orchestrator to IETF Network Slice Controller (NSC) for a customer and certain service type (e.g. CCTV, Infotainment, URLLC, etc.). It is NSC's decision to either create a new transport slice or use one of the existing ones. As a result, the mapping between 5G E2E network slice and IETF network slice is many to one, i.e. one 5G IETF network slice can be used with multiple 5G E2E network slices. The attributes of each 5G E2E network slices are included here. The 5g-network-slice-info contains the list of E2E network slices which are mapped to a 5G IETF network slice with all relevant attributes such as S-NSSAI, customer name and service type.

- 5g-connection-group: A 5G IETF network slice contains one or more connection groups each with its own SLA/SLO. Each connection group contains:

  * connection-group-attributes: A list of attributes for each 5g-connection-group such as connection-group-id, connection-group-name and connection-group-status

  * connection-group-member: A list of members. Each member is a connection between two endpoints. A connection group can contain one or more members. For example, the connections BBU1-UPF1 and BBU2-UPF1 are 2 members of "connection group U" in Figure 7.

  * connection-group-slo-policy: This is a mandatory policy. The connection-group-slo-policy represents in a generic and technology-agnostics fashion the SLO requirement needed to realize members of a connection group. It contains SLOs such

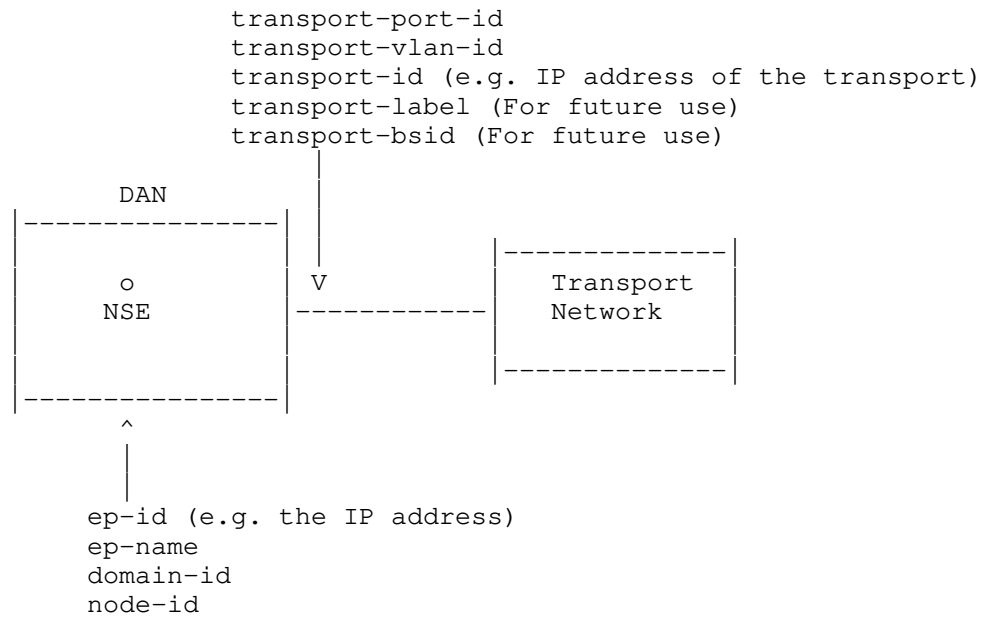
as bounded latency, bandwidth, reliability, security etc. Note that all members of a connection group must have the same SLO.

* connection-group-realization-policy: This is an optional policy. In some scenarios, the 5G E2E network slice orchestrator might be able to influence the IETF Network Slice Controller on how to realize a 5G IETF network slice by providing some technology-specific information.

* connection-group-monitoring-policy: This is an optional policy. The 5G E2E network slice orchestrator can influence the IETF Network Slice Controller on how to perform monitoring, analytics and optimization on 5G IETF Network Slices. It contains, the type of assurance needed, time interval, frequency of how often to inform the 5G E2E Network Slice Orchestrator etc.

- 5g-ietf-network-slice-endpoint: It contains the list of all endpoints along with their attributes which belong to a 5G IETF network slice. See Figure 11

- 5G-ietf-connection-group-member: It contains the list of all members of connectin groups along with their attributes which belong to a 5G IETF network slice.



Legend:
DAN: Device, application and/or network function
NSE: IETF Network Slice Endpoint

Figure 11: Details of the 5G IETF network slice endpoints

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

TBD

9. Acknowledgments

The authors would like to thank the following people for their contribution to this draft:

o Ryan Hoffman, Telus

10. Informative References

[I-D.boucadair-connectivity-provisioning-protocol]
Boucadair, M., Jacquenet, C., Zhang, D., and P. Georgatsos, "Connectivity Provisioning Negotiation Protocol (CPNP)", draft-boucadair-connectivity-provisioning-protocol-15 (work in progress), December 2017.

[I-D.contreras-teas-slice-nbi]
Contreras, L., Homma, S., and J. Ordonez-Lucena, "IETF Network Slice use cases and attributes for Northbound Interface of controller", draft-contreras-teas-slice-nbi-03 (work in progress), October 2020.

[I-D.homma-slice-provision-models]
Homma, S., Nishihara, H., Miyasaka, T., Galis, A., OV, V., Lopez, D., Contreras, L., Ordonez-Lucena, J., Martinez-Julia, P., Qiang, L., Rokui, R., Ciavaglia, L., and X. Foy, "Network Slice Provision Models", draft-homma-slice-provision-models-00 (work in progress), February 2019.

[I-D.ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-20 (work in progress), December 2017.

[I-D.ietf-teas-actn-vn-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., Yoon, B., Wu, Q., and P. Park, "A Yang Data Model for VN Operation", draft-ietf-teas-actn-vn-yang-04 (work in progress), February 2019.

[I-D.king-teas-applicability-actn-slicing]
King, D. and Y. Lee, "Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing", draft-king-teas-applicability-actn-slicing-04 (work in progress), October 2018.

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-00 (work in progress), October 2020.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

[I-D.qiang-coms-netslicing-information-model]
Qiang, L., Galis, A., Geng, L., kiran.makhijani@huawei.com, k., Martinez-Julia, P., Flinck, H., and X. Foy, "Technology Independent Information Model for Network Slicing", draft-qiang-coms-netslicing-information-model-02 (work in progress), January 2018.

[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.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <https://www.rfc-editor.org/info/rfc7297>.

[RFC8049] Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8049, DOI 10.17487/RFC8049, February 2017, <https://www.rfc-editor.org/info/rfc8049>.

[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>.

[TS.23.501-3GPP]
3rd Generation Partnership Project (3GPP), "3GPP TS 23.501 (V16.2.0): System Architecture for the 5G System (5GS); Stage 2 (Release 16)", September 2019, <http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-g20.zip>.

[TS.28.530-3GPP]
3rd Generation Partnership Project (3GPP), "3GPP TS 28.530 V15.1.0 Technical Specification Group Services and System Aspects; Management and orchestration; Concepts, use cases and requirements (Release 15)", December 2018, <http://ftp.3gpp.org//Specs/archive/28_series/28.530/28530-f10.zip>.

[TS.28.531-3GPP]
3rd Generation Partnership Project (3GPP), "3GPP TS 28.531 V16.2.0 Technical Specification Group Services and System Aspects; Management and orchestration; Provisioning; (Release 16)", June 2019, <http://ftp.3gpp.org//Specs/archive/28_series/28.531/28531-g20.zip>.

[TS.28.540-3GPP]
3rd Generation Partnership Project (3GPP), "3GPP TS 28.540 V16.0.0 Technical Specification Group Services and System Aspects; Management and orchestration; 5G Network Resource Model (NRM); Stage 1 (Release 16)", June 2019, <https://www.3gpp.org/ftp/Specs/archive/28_series/28.540/28540-g00.zip>.

[TS.28.541-3GPP]
3rd Generation Partnership Project (3GPP), "3GPP TS 28.541 V16.1.0 Technical Specification Group Services and System Aspects; Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 16)", June 2019, <http://www.3gpp.org/ftp//Specs/archive/28_series/28.541/28541-g10.zip>.

Authors' Addresses

Reza Rokui
Nokia
Canada

Email: reza.rokui@nokia.com

Shunsuke Homma
NTT
3-9-11, Midori-cho
Musashino-shi, Tokyo 180-8585
Japan

Email: shunsuke.homma.ietf@gmail.com

Xavier de Foy
InterDigital Inc.
Canada

Email: Xavier.Defoy@InterDigital.com

Luis M. Contreras
Telefonica
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Philip Eardley
BT
UK

Email: philip.eardley@bt.com

Kiran Makhijani
Futurewei Networks
US

Email: kiranm@futurewei.com

Hannu Flinck
Nokia
Finland

Email: hannu.flinck@nokia-bell-labs.com

Rainer Schatzmayr
Deutsche Telekom
Germany

Email: rainer.schatzmayr@telekom.de

Ali Tizghadam
TELUS Communications Inc
Canada

Email: ali.tizghadam@telus.com

Christopher Janz
Huawei Canada
Canada

Email: christopher.janz@huawei.com

Henry Yu
Huawei Canada
Canada

Email: henry.yu1@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2021

B. Wu
D. Dhody
Huawei Technologies
L. Han
China Mobile
R. Rokui
Nokia Canada
October 30, 2020

A Yang Data Model for IETF Network Slice NBI
draft-wd-teas-ietf-network-slice-nbi-yang-00

Abstract

This document provides a YANG data model for the IETF Network Slice NBI (Northbound Interface). The model can be used by a higher level system which is the IETF Network Slice consumer of an IETF Network Slice Controller (NSC) to request, configure, and manage the components of an IETF Network Slice.

The YANG modules in this document conforms to the Network Management Datastore Architecture (NMDA) defined in RFC 8342.

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 May 3, 2021.

Copyright Notice

Table of Contents

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.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework], which is used to operate IETF Network Slice during the IETF Network Slice instantiation, and the operations includes modification, deletion, and monitoring.

The YANG model discussed in this document describes the requirements of an IETF Network Slice that interconnects a set of IETF Network Slice Endpoints from the point of view of the consumer, which is classified as Customer Service Model in [RFC8309].

It will be up to the management system or NSC (IETF Network Slice controller) 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]. How the configuration of network elements is done 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:

- client
- configuration data
- state data

This document also makes use of the following terminology introduced in the YANG 1.1 Data Modeling Language [RFC7950]:

- augment
- data model
- data node

This document also makes use of the following terminology introduced in the IETF Network Slice definition draft [I-D.nsdt-teas-ietf-network-slice-definition]:

- IETF Network Slice (NS): An IETF Network Slice is a logical network topology connecting a number of endpoints and a set of shared or dedicated network resources, which are used to satisfy specific Service Level Objectives (SLO). The definition is from Section 3 of [I-D.nsdt-teas-ietf-network-slice-definition].

- IETF Network Slice Endpoint (NSE): An IETF Network Slice Endpoint is a logical identifier at DAN (Device,Application,Network Function) of the customer network to identify the logical access to which, a particular subset of traffic traversing the external interface, is mapped to a specific IETF Network Slice and it follows the definition of NSE (IETF Network Slice Endpoint) in Section 4.2 of [I-D.nsdt-teas-ietf-network-slice-definition].

- SLO: An SLO is a service level objective

- DAN: Device,Application,Network Function

- NSC: IETF Network Slice Controller

- NBI: Northbound Interface

In addition, this document defines the following terminology:

- IETF Network Slice Member (Network-Slice-Member): A IETF Network-Slice-Member is an abstract entity which represents the network resources mapped to a particular connection between a pair of NSEs belonging to an IETF Network Slice. Note that different SLO requirement per Network-Slice-Member could be applied.

- Network-Slice-Slo-Group: Indicates a group of Network-Slice-Members with same SLOs in one IETF Network Slice.

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 consumer, e.g. A higher level management system, to request and monitor IETF Network Slices. In particular, the model allows consumers to operate in an abstract, technology-agnostic manner, with implementation details hidden.

In the use case of 5G transport application, 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.

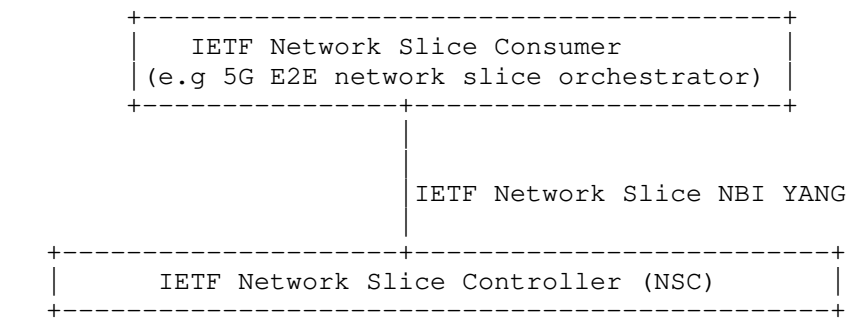
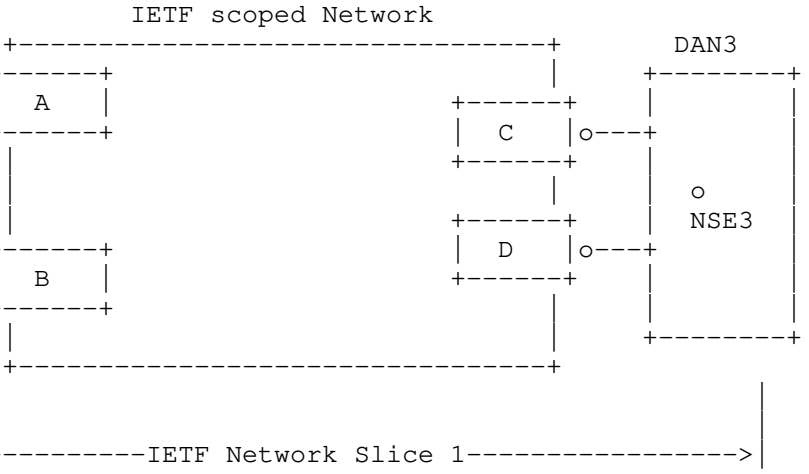


Figure 1 IETF Network Slice NBI Model Context

4. IETF Network Slice NBI Model Overview

From a consumer perspective, an example of an IETF Network Slice is shown in figure 2.



Legend:DAN (Device,Application,Network Function)

Network-Slice-SLO-Group Red
Network-Slice-Member 1 NSE1-NSE3
Network-Slice-Member 2 NSE2-NSE3

Network-Slice-SLO-Group Blue
Network-Slice-Member 3 NSE1-NSE2

Figure 2: An example of an IETF Network Slice

As shown in figure 2, an IETF Network Slice (NS) links together NSEs at the DANs, which are customer endpoints that request an IETF Network Slice. At each customer DAN, one or multiple NSEs could be connected to the IETF Network Slice.

A NS is a connectivity with specific SLO characteristics, including bandwidth, QoS metric, etc. The connectivity is a combination of logical connections, represented by Network-Slice-Members. When some parts of a slice have different SLO requirements, a group of Network-Slice-Members with the same SLO is described by Network-Slice-SLO-Group.

Based on this design, the IETF Network Slice YANG module consists of the main containers: "network-slice", "network-slice-endpoint", "network-slice-member",and "network-slice-slo-group".

The figure below describes the overall structure of the YANG module:

```
module: ietf-network-slice
  +--rw network-slices
     +--rw slice-templates
     |  +--rw slo-template* [id]
     |     +--rw id                        string
     |     +--rw template-description?   string
     +--rw network-slice* [network-slice-id]
        +--rw network-slice-id           uint32
        +--rw network-slice-name?        string
        +--rw network-slice-tag?         string
        +--rw network-slice-topology*    identityref
        +--rw network-slice-slo-group* [slo-group-name]
        |  +--rw slo-group-name                    string
        |  +--rw default-slo-group?                boolean
        |  +--rw (slo-template)?
        |  |  +--:(standard)
        |  |  |  +--rw template?                   leafref
        |  |  +--:(custom)
        |  |     +--rw network-slice-slo-policy
        |  |        +--rw latency
        |  |        |  +--rw one-way-latency?   uint32
        |  |        |  +--rw two-way-latency?   uint32
        |  |        +--rw jitter
        |  |        |  +--rw one-way-jitter?   uint32
        |  |        |  +--rw two-way-jitter?   uint32
        |  |        +--rw loss
        |  |        |  +--rw one-way-loss?   decimal64
        |  |        |  +--rw two-way-loss?   decimal64
        |  |        +--rw availability-type?             identityref
        |  |        +--rw isolation-type?                identityref
```

```
|  |        +--rw network-slice-metric-bounds
|  |           +--rw network-slice-metric-bound*
|  |                   [metric-type]
|  |              +--rw metric-type    identityref
|  |              +--rw upper-bound?   uint64
|  +--rw network-slice-member-group*
|  |       [network-slice-member-id]
|  |  +--rw network-slice-member-id    leafref
|  +--ro slo-group-monitoring
|     +--ro latency?   uint32
|     +--ro jitter?    uint32
|     +--ro loss?      decimal64
+--rw status
|  +--rw admin-enabled?   boolean
|  +--ro oper-status?     operational-type
+--rw network-slice-endpoint* [endpoint-id]
|  +--rw endpoint-id                    uint32
|  +--rw endpoint-name?                 string
|  +--rw endpoint-role*                 identityref
|  +--rw geolocation
|  |  +--rw altitude?    int64
|  |  +--rw latitude?    decimal64
|  |  +--rw longitude?   decimal64
|  +--rw node-id?                       string
|  +--rw port-id?                       string
|  +--rw network-slice-match-criteria
|  |  +--rw network-slice-match-criteria* [match-type]
|  |     +--rw match-type    identityref
|  |     +--rw value?        string
|  +--rw endpoint-ip?                   inet:host
|  +--rw bandwidth
|  |  +--rw incoming-bandwidth
|  |  |  +--rw guaranteed-bandwidth?   te-types:te-bandwidth
|  |  +--rw outgoing-bandwidth
|  |     +--rw guaranteed-bandwidth?   te-types:te-bandwidth
|  +--rw mtu                              uint16
|  +--rw routing
|  |  +--rw bgp
|  |  |  +--rw bgp-peer-ipv4*   inet:ipv4-prefix
|  |  |  +--rw bgp-peer-ipv6*   inet:ipv6-prefix
|  |  +--rw static
|  |     +--rw static-route-ipv4*   inet:ipv4-prefix
|  |     +--rw static-route-ipv6*   inet:ipv6-prefix
|  +--rw status
|  |  +--rw admin-enabled?   boolean
|  |  +--ro oper-status?     operational-type
|  +--ro endpoint-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 network-slice-member* [network-slice-member-id]
   +--rw network-slice-member-id            uint32
   +--rw src
   |  +--rw src-network-slice-endpoint-id?   leafref
   +--rw dest
   |  +--rw dest-network-slice-endpoint-id?   leafref
   +--rw monitoring-type?
   |       network-slice-monitoring-type
   +--ro network-slice-member-monitoring
      +--ro latency?   uint32
      +--ro jitter?    uint32
      +--ro loss?      decimal64
```

5. IETF Network Slice NBI Model Description

An IETF Network Slice consists of a group of interconnected NSEs, and the connections between NSEs may have different SLO requirements, including symmetrical or asymmetrical traffic throughput, different traffic delay, etc.

5.1. IETF Network Slice Connection Types

An IETF Network Slice can be point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), or multipoint-to-multipoint (MP2MP) based on the consumer's traffic pattern requirements.

Therefore, the "network-slice-topology" under the node "network-slice" is required for configuration. The model supports any-to-any, Hub and Spoke (where Hubs can exchange traffic), and the different combinations. New topologies could be added via augmentation. By default, the any-to-any topology is used.

In addition, "endpoint-role" under the node "network-slice-endpoint" also needs to be defined, which specifies the role of the NSE in a particular Network Slice topology. In the any-to-any topology, all NSEs MUST have the same role, which will be "any-to-any-role". In the Hub-and-Spoke topology, NSEs MUST have a Hub role or a Spoke role.

5.2. IETF Network Slice Endpoint (NSE)

An NSE belong to a single IETF Network Slice. An IETF Network Slice involves two or more NSEs.

A NSE is used to define the limit on the user traffic that can be injected to a network slice. For example, in some scenarios, the access traffic of a DAN is allowed only when it matches the logical Layer 2 connection identifier. In some scenarios, the access traffic of a DAN is allowed only when the traffic matches a source IP address. Sometimes, the traffic from a distinct physical connection of a DAN is allowed.

Therefore, to ensure that the NSE is uniquely identified, the model use the following parameters including "node-id", "port-id" and "network-slice-match-criteria". The "node-id" identifies a DAN node, the "port-id" identifies a port, and the "network-slice-filter-criteria" identifies a possible logical L2 ID or IP address or other possible traffic identifier in the user traffic.

Additionally, a number of slice interconnection parameters need to be agreed with a customer DAN and the IETF network, such as IP address (v4 or v6) etc.

5.3. IETF Network Slice SLO

As defined in [I-D.nsdt-teas-ietf-network-slice-definition], this model defines the minimum IETF Network Slice SLO attributes, and other SLO nodes can be augmented as needed. NS SLO assurance is implemented through the following mechanisms:

- Network Slice SLO list: Which defines the performance objectives of the NS. Performance objectives can be specified for various performance metrics,and different objectives are as follows:

  Latency: Indicates the maximum latency between two NSE. The unit is micro seconds. The latency could be round trip times or one-way metrics.

  Jitter: Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the one- way delay between sequential packets in a flow.

  Loss: Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

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.

Isolation: Whether the isolation needs to be explicitly requested is still in discussion.

- Bandwidth: Indicates the guaranteed minimum bandwidth between any two NSE. The unit is data rate per second. And the bandwidth is unidirectional. The bandwidth is specified at each NSE and can be applied to incoming NS traffic or outgoing NS traffic. When applied in the incoming direction, the Bandwidth 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.

Note: About the definition of SLO parameters, the author is discussing to reuse the TE-Types grouping definition as much as possible, to avoid duplication of definitions.

Consumers' Network Slices can be very different, e.g. some slices has the same SLO requirements of connections, some slices has the different SLO requirements for different parts of the slice. In some slices, the bandwidth of one endpoint is different from that of other endpoints, for example, one is central endpoint, the other endpoints are access endpoints.

The list "ns-slo-group" defines a group of different SLOs, which are used to describe that different parts of the slice have different SLOs. The specific SLO of the slice SLO group may use a standard SLO template, or may use different customized parameters. A group of "network-slice-member" is used to describe which connections of the slice use the SLO.

For some simplest IETF Network Slices, only one category SLO of "network-slice-slo-group" needs to be defined. For some complicated network slices, in addition to the configurations above, multiple "network-slice-slo-group" needs to be defined, and "network-slice-member-group" describes details of the per-connection SLO.

In addition to SLO performance objectives, there are also some other NS objectives, such as MTU and security which can be augmented when needed. MTU specifies the maximum packet length that the network slice guarantee to be able to carry across.

Note: In some use cases, the number of connections represented by "network-slice-member-group" may be huge, which may lead to configuration issues, for example, the scalability or error-prone.

6. IETF Network Slice Monitoring

This model also describes performance status of an IETF Network Slice. The statistics are described in the following granularity:

- Per NS SLO group: specified in 'network-slice-member-group-monitoring' under the "network-slice-slo-groupr"
- Per NS connection: specified in 'network-slice-member-monitoring' under the "network-slice-member"
- Per NS Endpoint: specified in 'ep-monitoring' under the "network-slice-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 'network-slice-member' or 'network-slice-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 client management system could initiate modification based on the IETF Network Slice running status.

7. IETF Network Slice NBI Model Usage Example

TBD

8. IETF Network Slice NBI Module

<CODE BEGINS> file "ietf-transport-slice@2020-07-12.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;
  }
  import ietf-te-types {
    prefix te-types;
  }
```

```
  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>";
  description
    "This module contains a YANG module for the IETF Network Slice.

     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 2020-10-27 {
    description
      "initial version.";
    reference
      "RFC XXXX: A Yang Data Model for IETF Network Slice Operation";
  }

  /* Features */
  /* Identities */

  identity network-slice-topology {
    description
      "Base identity for IETF Network Slice topology.";
  }

  identity any-to-any {
    base network-slice-topology;
    description
      "Identity for any-to-any IETF Network Slice topology.";
  }

  identity hub-spoke {
    base network-slice-topology;
    description
```

```
      "Identity for Hub-and-Spoke IETF Network Slice topology.";
  }

  identity endpoint-role {
    description
      "Network Slice Endpoint Role in an IETF Network Slice topology ";
  }

  identity any-to-any-role {
    base endpoint-role;
    description
      "Network Slice Endpoint as the any-to-any role in an any-to-any
       IETF Network Slice.";
  }

  identity hub {
    base endpoint-role;
    description
      "Network Slice Endpoint as the hub role in a Hub-and-Spoke
       IETF Network Slice.";
  }

  identity spoke {
    base endpoint-role;
    description
      "Network Slice Endpoint as the spoke role in a Hub-and-Spoke
       IETF Network Slice.";
  }

  identity isolation-type {
    description
      "Base identity from which specific isolation types are derived.";
  }

  identity physical-isolation {
    base isolation-type;
    description
      "physical isolation.";
  }

  identity logical-isolation {
    base isolation-type;
    description
      "logical-isolation.";
  }

  identity network-slice-slo-metric-type {
    description
```

```
      "Base identity for Network Slice SLO metric type";
  }

  identity network-slice-match-type {
    description
      "Base identity for Network Slice traffic match type";
  }

  identity network-slice-vlan-match {
    base network-slice-match-type;
    description
      "VLAN as Network Slice traffic match criteria.";
  }

  /*
   * 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 network-slice-monitoring-type {
    type enumeration {
      enum one-way {
        description
          "represents one-way monitoring type";
      }
      enum two-way {
        description
          "represents two-way monitoring type";
      }
    }
    description
      "enumerated type of monitoring on a network-slice-member ";
  }

  /* Groupings */
```

```
  grouping status-params {
    description
      "Grouping used to join operational and administrative status";
    container status {
      description
        "Container for status of administration and operational";
      leaf admin-enabled {
        type boolean;
        description
          "Administrative Status UP/DOWN";
      }
      leaf oper-status {
        type operational-type;
        config false;
        description
          "Operations status";
      }
    }
  }

  grouping network-slice-match-criteria {
    description
      "Grouping for Network Slice match definition.";
    container network-slice-match-criteria {
      description
        "Describes Network Slice match criteria.";
      list network-slice-match-criteria {
        key "match-type";
        description
          "List of Network Slice traffic criteria";
        leaf match-type {
          type identityref {
            base network-slice-match-type;
          }
          description
            "Identifies an entry in the list of match-type for
             the Network Slice.";
        }
        leaf value {
          type string;
          description
            "Describes Network Slice match criteria,e.g. IP address,
             VLAN, etc.";
        }
      }
    }
  }
```

```
  grouping network-slice-metric-bounds {
    description
      "Network Slice metric bounds grouping";
    container network-slice-metric-bounds {
      description
        "Network Slice metric bounds container";
      list network-slice-metric-bound {
        key "metric-type";
        description
          "List of Network Slice metric bounds";
        leaf metric-type {
          type identityref {
            base network-slice-slo-metric-type;
          }
          description
            "Identifies an entry in the list of metric-types
             bound for the Network Slice.";
        }
        leaf upper-bound {
          type uint64;
          default "0";
          description
            "Upper bound on network-slice-member metric. A zero indicate
             an unbounded upper limit for the specific metric-type";
        }
      }
    }
  }

  grouping routing-protocols {
    description
      "Grouping for endpoint protocols definition.";
    container routing {
      description
        "Describes protocol between Network Slice Endpoint and IETF
         scoped network edge device.";
      container bgp {
        description
          "BGP-specific configuration.";
        leaf-list bgp-peer-ipv4 {
          type inet:ipv4-prefix;
          description
            "BGP peer ipv4 address.";
        }
        leaf-list bgp-peer-ipv6 {
          type inet:ipv6-prefix;
          description
            "BGP peer ipv6 address.";
```

```
        }
      }
      container static {
        description
          "Only applies when protocol is static.";
        leaf-list static-route-ipv4 {
          type inet:ipv4-prefix;
          description
            "ipv4 static route";
        }
        leaf-list static-route-ipv6 {
          type inet:ipv6-prefix;
          description
            "ipv6 static route";
        }
      }
    }
  }

  grouping endpoint-monitoring-parameters {
    description
      "Grouping for endpoint-monitoring-parameters.";
    container endpoint-monitoring {
      config false;
      description
        "Container for endpoint-monitoring-parameters.";
      leaf incoming-utilized-bandwidth {
        type te-types:te-bandwidth;
        description
          "Bandwidth utilization that represents the actual
           utilization of the incoming 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
          "Bandwidth utilization that represents the actual
           utilization of the incoming 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 uint32;
      units "usec";
      description
        "The latency statistics per Network Slice member.";
    }
    leaf jitter {
      type uint32 {
        range "0..16777215";
      }
      description
        "The jitter statistics per Network Slice member.";
    }
    leaf loss {
      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 geolocation {
      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 endpoint {
    description
      "IETF Network Slice endpoint related information";
    leaf endpoint-id {
      type uint32;
      description
        "unique identifier for the referred IETF Network
         Slice endpoint";
    }
    leaf endpoint-name {
      type string;
      description
        "endpoint name";
    }
    leaf-list endpoint-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 port-id {
      type string;
      description
        "Reference to the Port-id of the customer node.";
    }
    uses network-slice-match-criteria;
    leaf endpoint-ip {
      type inet:host;
      description
        "The address of the TACACS+ server.";
    }
    container bandwidth {
      container incoming-bandwidth {
        leaf guaranteed-bandwidth {
          type te-types:te-bandwidth;
          description
            "If guaranteed-bandwidth is 0, it means best effort, no
             minimum throughput is guaranteed.";
        }
        description
          "Container for the incoming bandwidth policy";
      }
      container outgoing-bandwidth {
        leaf guaranteed-bandwidth {
          type te-types:te-bandwidth;
          description
            "If guaranteed-bandwidth is 0, it means best effort, no
             minimum throughput is guaranteed.";
        }
        description
          "Container for the bandwidth policy";
      }
      description
        "Container for the bandwidth policy";
    }
    leaf mtu {
```

```
      type uint16;
      units "bytes";
      mandatory true;
      description
        "MTU of Network Slice traffic. If the traffic type is IP,
         it refers to the IP MTU. If the traffic type is Ethertype,
         will refer to the Ethernet MTU. ";
    }
    uses routing-protocols;
    uses status-params;
    uses endpoint-monitoring-parameters;
  }

  //network-slice-endpoint

  grouping network-slice-member {
    description
      "network-slice-member is described by this container";
    leaf network-slice-member-id {
      type uint32;
      description
        "network-slice-member identifier";
    }
    container src {
      description
        "the source of Network Slice link";
      leaf src-network-slice-endpoint-id {
        type leafref {
          path "/network-slices/network-slice/"
             + "network-slice-endpoint/endpoint-id";
        }
        description
          "reference to source Network Slice endpoint";
      }
    }
    container dest {
      description
        "the destination of Network Slice link ";
      leaf dest-network-slice-endpoint-id {
        type leafref {
          path "/network-slices/network-slice"
             + "/network-slice-endpoint/endpoint-id";
        }
        description
          "reference to dest Network Slice endpoint";
      }
    }
    leaf monitoring-type {
```

```
      type network-slice-monitoring-type;
      description
        "One way or two way monitoring type.";
    }
    container network-slice-member-monitoring {
      config false;
      description
        "SLO status Per network-slice endpoint to endpoint ";
      uses common-monitoring-parameters;
    }
  }

  //network-slice-member

  grouping network-slice-slo-group {
    description
      "Grouping for SLO definition of Network Slice";
    list network-slice-slo-group {
      key "slo-group-name";
      description
        "List of Network Slice SLO groups, the SLO group is used to
         support different SLO objectives between different
         network-slice-members in the same slice.";
      leaf slo-group-name {
        type string;
        description
          "Identifies an entry in the list of SLO group for the
           Network Slice.";
      }
      leaf default-slo-group {
        type boolean;
        default "false";
        description
          "Is the SLO group is selected as the default-slo-group";
      }
      choice slo-template {
        description
          "Choice for SLO template.
           Can be standard template or customized template.";
        case standard {
          description
            "Standard SLO template.";
          leaf template {
            type leafref {
              path "/network-slices/slice-templates/slo-template/id";
            }
            description
              "QoS template to be used.";
```

```
          }
        }
        case custom {
          description
            "Customized SLO template.";
          container network-slice-slo-policy {
            container latency {
              leaf one-way-latency {
                type uint32 {
                  range "0..16777215";
                }
                units "usec";
                description
                  "Lowest latency in micro seconds.";
              }
              leaf two-way-latency {
                type uint32 {
                  range "0..16777215";
                }
                description
                  "Lowest-way delay or latency in micro seconds.";
              }
              description
                "Latency constraint on the traffic class.";
            }
            container jitter {
              leaf one-way-jitter {
                type uint32 {
                  range "0..16777215";
                }
                description
                  "lowest latency in micro seconds.";
              }
              leaf two-way-jitter {
                type uint32 {
                  range "0..16777215";
                }
                description
                  "lowest-way delay or latency in micro seconds.";
              }
              description
                "Jitter constraint on the traffic class.";
            }
            container loss {
              leaf one-way-loss {
                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";
              }
              leaf two-way-loss {
                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";
              }
              description
                "Loss constraint on the traffic class.";
            }
            leaf availability-type {
              type identityref {
                base availability-type;
              }
              description
                "Availability Requirement for the Network Slice";
            }
            leaf isolation-type {
              type identityref {
                base isolation-type;
              }
              default "logical-isolation";
              description
                "Network Slice isolation-level.";
            }
            uses network-slice-metric-bounds;
            description
              "container for customized policy constraint on the slice
               traffic.";
          }
        }
      }
      list network-slice-member-group {
        key "network-slice-member-id";
        description
```

```
          "List of included Network Slice Member groups for the SLO.";
        leaf network-slice-member-id {
          type leafref {
            path "/network-slices/network-slice/"
               + "network-slice-member/network-slice-member-id";
          }
          description
            "Identifies the included list of Network Slice member.";
        }
      }
      container slo-group-monitoring {
        config false;
        description
          "SLO status Per slo group ";
        uses common-monitoring-parameters;
      }
    }
  }

  grouping slice-template {
    description
      "Grouping for slice-templates.";
    container slice-templates {
      description
        "Container for slice-templates.";
      list slo-template {
        key "id";
        leaf id {
          type string;
          description
            "Identification of the SLO Template to be used.
             Local administration meaning.";
        }
        leaf template-description {
          type string;
          description
            "Description of the SLO template.";
        }
        description
          "List for SLO template identifiers.";
      }
    }
  }

  /* Configuration data nodes */

  container network-slices {
    description
```

```
      "network-slice configurations";
    uses slice-template;
    list network-slice {
      key "network-slice-id";
      description
        "a network-slice is identified by a network-slice-id";
      leaf network-slice-id {
        type uint32;
        description
          "a unique network-slice identifier";
      }
      leaf network-slice-name {
        type string;
        description
          "network-slice name";
      }
      leaf network-slice-tag {
        type string;
        description
          "Network Slice tag for operational management";
      }
      leaf-list network-slice-topology {
        type identityref {
          base network-slice-topology;
        }
        default "any-to-any";
        description
          "Network Slice topology.";
      }
      uses network-slice-slo-group;
      uses status-params;
      list network-slice-endpoint {
        key "endpoint-id";
        uses endpoint;
        description
          "list of endpoints in this slice";
      }
      list network-slice-member {
        key "network-slice-member-id";
        description
          "List of network-slice-member in a slice";
        uses network-slice-member;
      }
    }
    //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

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-00 (work in progress), October 2020.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

[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>.

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-02 (work in progress), July 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-09 (work in progress), July 2020.

[I-D.liu-teas-transport-network-slice-yang]
Liu, X., Tantsura, J., Bryskin, I., Contreras, L., WU, Q., Belotti, S., and R. Rokui, "Transport Network Slice YANG Data Model", draft-liu-teas-transport-network-slice-yang-01 (work in progress), July 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. Comparison with Other Possible Design choices for IETF Network Slice NBI

According to the 3.3.1. Northbound Inteface (NBI) [I-D.nsdt-teas-ns-framework], the IETF Network Slice NBI is a technology-agnostic interface, which is used for a consumer to express requirements for a particular IETF Network Slice. Consumers 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.

A.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 consumer 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. 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.

- 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. 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.

A.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 consumer 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:

- Network: a transport network list and an list of nodes contained in the network

- Link: "links" list and "termination points" list describe how nodes in a network are connected to each other

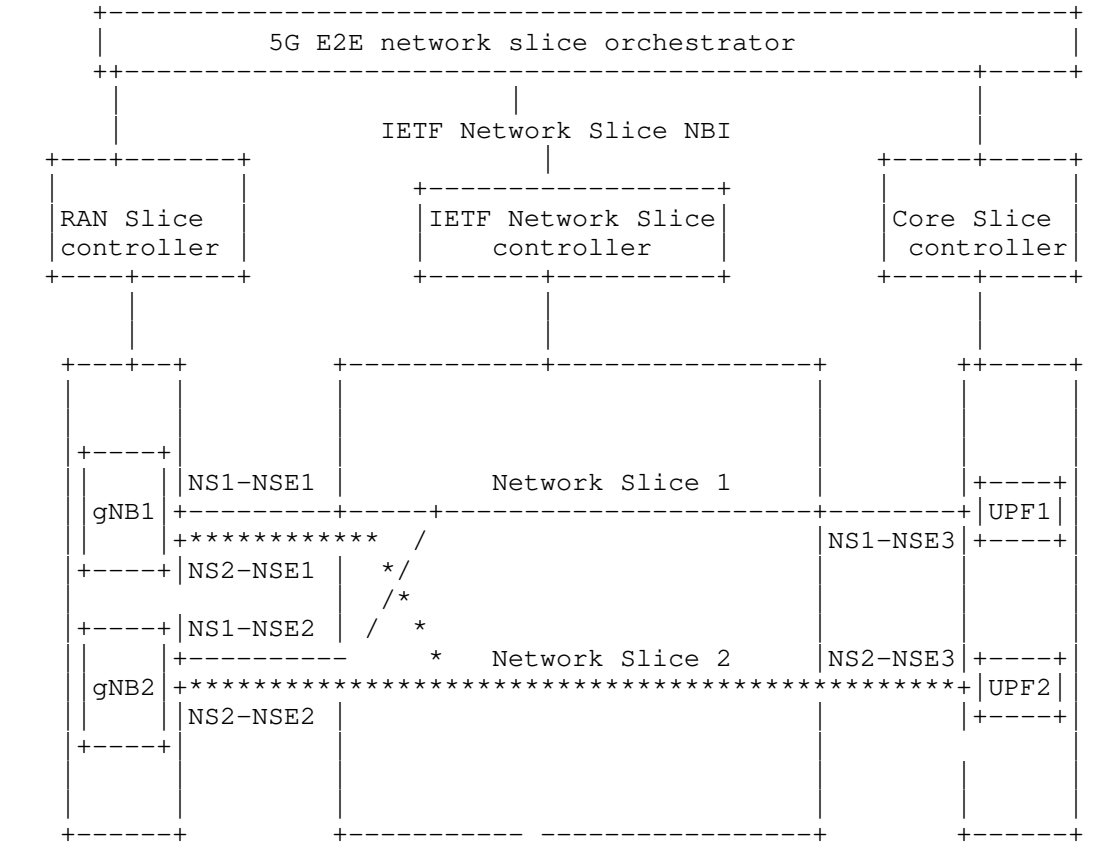
- Support network: vertical layering relationships between IETF Network Slice networks and underlay networks

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 B. Appendix B IETF Network Slice Filter 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 "port-id" to identify a Network Slice Endpont, 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

Liuyan Han
China Mobile

Email: hanliuyan@chinamobile.com

Reza Rokui
Nokia Canada

Email: reza.rokui@nokia.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 19, 2021

B. Wu
D. Dhody
Huawei Technologies
L. Han
China Mobile
R. Rokui
Nokia Canada
November 15, 2020

A Yang Data Model for IETF Network Slice NBI
draft-wd-teas-ietf-network-slice-nbi-yang-01

Abstract

This document provides a YANG data model for the IETF Network Slice NBI (Northbound Interface). The model can be used by a higher level system which is the IETF Network Slice consumer of an IETF Network Slice Controller (NSC) to request, configure, and manage the components of an IETF Network Slice.

The YANG modules in this document conforms to the Network Management Datastore Architecture (NMDA) defined in RFC 8342.

Table of Contents

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.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework], which is used to operate IETF Network Slice during the IETF Network Slice instantiation, and the operations includes modification, deletion, and monitoring.

The YANG model discussed in this document describes the requirements of an IETF Network Slice that interconnects a set of IETF Network Slice Endpoints from the point of view of the consumer, which is classified as Customer Service Model in [RFC8309].

It will be up to the management system or NSC (IETF Network Slice controller) 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]. How the configuration of network elements is done 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:

- client
- configuration data
- state data

This document also makes use of the following terminology introduced in the YANG 1.1 Data Modeling Language [RFC7950]:

- augment
- data model
- data node

This document also makes use of the following terminology introduced in the IETF Network Slice definition draft [I-D.nsdt-teas-ietf-network-slice-definition]:

- IETF Network Slice (NS): An IETF Network Slice is a logical network topology connecting a number of endpoints and a set of shared or dedicated network resources, which are used to satisfy specific Service Level Objectives (SLO). The definition is from Section 3 of [I-D.nsdt-teas-ietf-network-slice-definition].

- IETF Network Slice Endpoint (NSE): An IETF Network Slice Endpoint is a logical identifier at DAN (Device,Application,Network Function) of the customer network to identify the logical access to which, a particular subset of traffic traversing the external interface, is mapped to a specific IETF Network Slice and it follows the definition of NSE (IETF Network Slice Endpoint) in Section 4.2 of [I-D.nsdt-teas-ietf-network-slice-definition].

- SLO: An SLO is a service level objective

- DAN: Device,Application,Network Function

- NSC: IETF Network Slice Controller

- NBI: Northbound Interface

In addition, this document defines the following terminology:

- IETF Network Slice Member (Network-Slice-Member): A IETF Network-Slice-Member is an abstract entity which represents the network resources mapped to a particular connection between a pair of NSEs belonging to an IETF Network Slice. Note that different SLO requirement per Network-Slice-Member could be applied.

- Network Slice Connection Group: Represents a set of Network Slice Members with same SLO attributes in one IETF Network Slice.

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 consumer, e.g. A higher level management system, to request and monitor IETF Network Slices. In particular, the model allows consumers to operate in an abstract, technology-agnostic manner, with implementation details hidden.

In the use case of 5G transport application, 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.

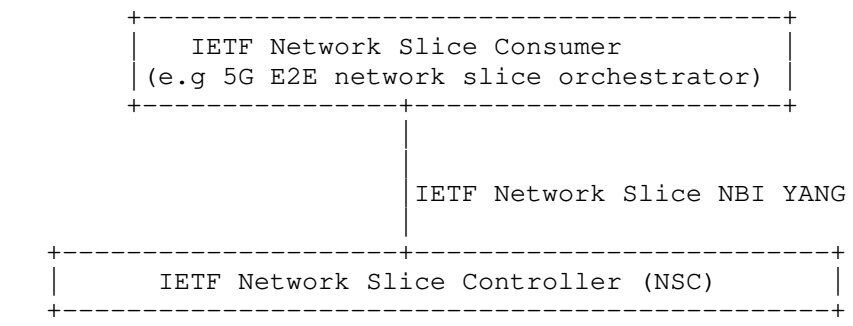
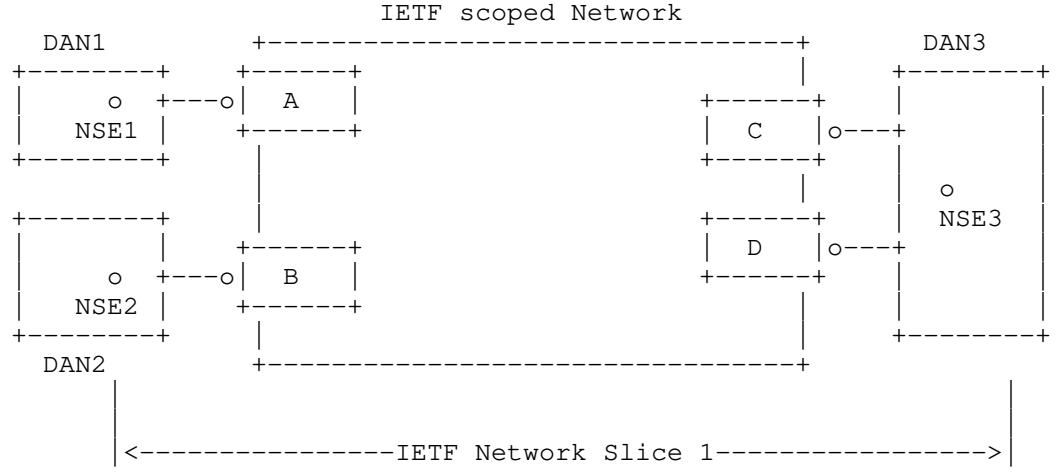


Figure 1 IETF Network Slice NBI Model Context

4. IETF Network Slice NBI Model Overview

From a consumer perspective, an example of an IETF Network Slice is shown in figure 2.



Legend:DAN (Device,Application,Network Function)

Network-Slice-Connection-Group Red Network-Slice-Connection-Group Blue
Network-Slice-Member 1 NSE1-NSE3 Network-Slice-Member 3 NSE1-NSE2
Network-Slice-Member 2 NSE2-NSE3

Figure 2: An example of an IETF Network Slice

As shown in figure 2, an IETF Network Slice (NS) links together NSEs at the DANs, which are customer endpoints that request an IETF Network Slice. At each customer DAN, one or multiple NSEs could be connected to the IETF Network Slice.

An IETF Network Slice is a connectivity with specific SLO characteristics, including bandwidth, QoS metric, etc. The connectivity is a combination of logical connections, represented by Network-Slice-Members. When some parts of a network slice have different SLO requirements, a set of Network-Slice-Members with the same SLO is described by Network Slice Connection Group.

Based on this design, the IETF Network Slice YANG module consists of the main containers: "network-slice", "network-slice-endpoint", "network-slice-member",and "network-slice-connection-group".

The figure below describes the overall structure of the YANG module:

```
module: ietf-network-slice
  +--rw ietf-network-slices
     +--rw slice-templates
     |  +--rw slo-template* [id]
     |     +--rw id                        string
     |     +--rw template-description?   string
     +--rw ietf-network-slice* [network-slice-id]
        +--rw network-slice-id                  uint32
        +--rw network-slice-name?               string
        +--rw network-slice-tag*                string
        +--rw network-slice-topology*           identityref
        +--rw network-slice-connection-group* [connection-group-name]
        |  +--rw connection-group-name             string
        |  +--rw default-connection-group?         boolean
        |  +--rw (slo-template)?
        |  |  +--:(standard)
        |  |  |  +--rw template?                   leafref
        |  |  +--:(custom)
        |  |     +--rw network-slice-slo-policy
        |  |        +--rw latency
        |  |        |  +--rw one-way-latency?   uint32
        |  |        |  +--rw two-way-latency?   uint32
        |  |        +--rw jitter
        |  |        |  +--rw one-way-jitter?   uint32
        |  |        |  +--rw two-way-jitter?   uint32
        |  |        +--rw loss
        |  |        |  +--rw one-way-loss?   decimal64
        |  |        |  +--rw two-way-loss?   decimal64
        |  |        +--rw availability-type?             identityref
        |  |        +--rw isolation-type?                identityref
```

```
|  |        +--rw network-slice-metric-bounds
|  |           +--rw network-slice-metric-bound*
|  |                   [metric-type]
|  |              +--rw metric-type    identityref
|  |              +--rw upper-bound?   uint64
|  +--rw network-slice-member-group*
|  |       [network-slice-member-id]
|  |  +--rw network-slice-member-id    leafref
|  +--ro connection-group-monitoring
|     +--ro latency?   uint32
|     +--ro jitter?    uint32
|     +--ro loss?      decimal64
+--rw status
|  +--rw admin-enabled?   boolean
|  +--ro oper-status?     operational-type
+--rw network-slice-endpoint* [endpoint-id]
|  +--rw endpoint-id                    uint32
|  +--rw endpoint-name?                 string
|  +--rw endpoint-role*                 identityref
|  +--rw geolocation
|  |  +--rw altitude?    int64
|  |  +--rw latitude?    decimal64
|  |  +--rw longitude?   decimal64
|  +--rw node-id?                       string
|  +--rw port-id?                       string
|  +--rw network-slice-match-criteria
|  |  +--rw network-slice-match-criteria* [match-type]
|  |     +--rw match-type    identityref
|  |     +--rw value?        string
|  +--rw endpoint-ip?                   inet:host
|  +--rw bandwidth
|  |  +--rw incoming-bandwidth
|  |  |  +--rw guaranteed-bandwidth?   te-types:te-bandwidth
|  |  +--rw outgoing-bandwidth
|  |     +--rw guaranteed-bandwidth?   te-types:te-bandwidth
|  +--rw mtu                              uint16
|  +--rw routing
|  |  +--rw bgp
|  |  |  +--rw bgp-peer-ipv4*   inet:ipv4-prefix
|  |  |  +--rw bgp-peer-ipv6*   inet:ipv6-prefix
|  |  +--rw static
|  |     +--rw static-route-ipv4*   inet:ipv4-prefix
|  |     +--rw static-route-ipv6*   inet:ipv6-prefix
|  +--rw status
|  |  +--rw admin-enabled?   boolean
|  |  +--ro oper-status?     operational-type
|  +--ro endpoint-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 network-slice-member* [network-slice-member-id]
   +--rw network-slice-member-id            uint32
   +--rw src
   |  +--rw src-network-slice-endpoint-id?   leafref
   +--rw dest
   |  +--rw dest-network-slice-endpoint-id?   leafref
   +--rw monitoring-type?
   |       network-slice-monitoring-type
   +--ro network-slice-member-monitoring
      +--ro latency?   uint32
      +--ro jitter?    uint32
      +--ro loss?      decimal64
```

5. IETF Network Slice NBI Model Description

An IETF Network Slice consists of a group of interconnected NSEs, and the connections between NSEs may have different SLO requirements, including symmetrical or asymmetrical traffic throughput, different traffic delay, etc.

5.1. IETF Network Slice Connection Types

An IETF Network Slice can be point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), or multipoint-to-multipoint (MP2MP) based on the consumer's traffic pattern requirements.

Therefore, the "network-slice-topology" under the node "network-slice" is required for configuration. The model supports any-to-any, Hub and Spoke (where Hubs can exchange traffic), and the different combinations. New topologies could be added via augmentation. By default, the any-to-any topology is used.

In addition, "endpoint-role" under the node "network-slice-endpoint" also needs to be defined, which specifies the role of the NSE in a particular Network Slice topology. In the any-to-any topology, all NSEs MUST have the same role, which will be "any-to-any-role". In the Hub-and-Spoke topology, NSEs MUST have a Hub role or a Spoke role.

5.2. IETF Network Slice Endpoint (NSE)

An NSE belong to a single IETF Network Slice. An IETF Network Slice involves two or more NSEs.

A NSE is used to define the limit on the user traffic that can be injected to a network slice. For example, in some scenarios, the access traffic of a DAN is allowed only when it matches the logical Layer 2 connection identifier. In some scenarios, the access traffic of a DAN is allowed only when the traffic matches a source IP address. Sometimes, the traffic from a distinct physical connection of a DAN is allowed.

Therefore, to ensure that the NSE is uniquely identified, the model use the following parameters including "node-id", "port-id" and "network-slice-match-criteria". The "node-id" identifies a DAN node, the "port-id" identifies a port, and the "network-slice-match-criteria" identifies a possible logical L2 ID or IP address or other possible traffic identifier in the user traffic.

Additionally, a number of slice interconnection parameters need to be agreed with a customer DAN and the IETF network, such as IP address (v4 or v6) etc.

5.3. IETF Network Slice SLO

As defined in [I-D.nsdt-teas-ietf-network-slice-definition], this model defines the minimum IETF Network Slice SLO attributes, and other SLO nodes can be augmented as needed. NS SLO assurance is implemented through the following mechanisms:

- Network Slice SLO list: Which defines the performance objectives of the NS. Performance objectives can be specified for various performance metrics,and different objectives are as follows:

  Latency: Indicates the maximum latency between two NSE. The unit is micro seconds. The latency could be round trip times or one-way metrics.

  Jitter: Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the one- way delay between sequential packets in a flow.

  Loss: Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

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.

Isolation: Whether the isolation needs to be explicitly requested is still in discussion.

- Bandwidth: Indicates the guaranteed minimum bandwidth between any two NSE. The unit is data rate per second. And the bandwidth is unidirectional. The bandwidth is specified at each NSE and can be applied to incoming NS traffic or outgoing NS traffic. When applied in the incoming direction, the Bandwidth 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.

Note: About the definition of SLO parameters, the author is discussing to reuse the TE-Types grouping definition as much as possible, to avoid duplication of definitions.

Consumers' Network Slices can be very different, e.g. some slices has the same SLO requirements of connections, some slices has the different SLO requirements for different parts of the slice. In some slices, the bandwidth of one endpoint is different from that of other endpoints, for example, one is central endpoint, the other endpoints are access endpoints.

The list "network-slice-connection-group" defines a set of "network-slice-membe" with a particular SLO attributes, which are used to describe that different parts of a network slice have different SLOs. The specific SLOs of the slice connection group may use a standard SLO template, or may use different customized parameters. A group of "network-slice-member" is used to describe which connections of the slice use the SLOs.

For some simplest IETF Network Slices, only one category SLO of "network-slice-connection-group" needs to be defined. For some complicated network slices, in addition to the configurations above, multiple "network-slice-connection-group" needs to be defined, and "network-slice-member-group" describes details of the per-connection SLO.

In addition to SLO performance objectives, there are also some other Network Slice objectives, such as MTU and security which can be augmented when needed. MTU specifies the maximum packet length that the network slice guarantee to be able to carry across.

Note: In some use cases, the number of connections represented by "network-slice-member-group" may be huge, which may lead to configuration issues, for example, the scalability or error-prone.

6. IETF Network Slice Monitoring

This model also describes performance status of an IETF Network Slice. The statistics are described in the following granularity:

- Per NS Connection group: specified in 'connection-group-monitoring' under the "network-slice-connection-group"
- Per NS connection: specified in 'network-slice-member-monitoring' under the "network-slice-member"
- Per NS Endpoint: specified in 'endpoint-monitoring' under the "network-slice-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 'network-slice-member' or 'network-slice-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 client management system could initiate modification based on the IETF Network Slice running status.

7. IETF Network Slice NBI Module

<CODE BEGINS> file "ietf-network-slice@2020-11-13.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;
  }
  import ietf-te-types {
    prefix te-types;
  }

  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>";
  description
    "This module contains a YANG module for the IETF Network Slice.

     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 2020-11-13 {
    description
      "initial version.";
    reference
      "RFC XXXX: A Yang Data Model for IETF Network Slice Operation";
  }

  /* Features */
  /* Identities */

  identity network-slice-topology {
    description
      "Base identity for IETF Network Slice topology.";
  }

  identity any-to-any {
    base network-slice-topology;
    description
      "Identity for any-to-any IETF Network Slice topology.";
  }

  identity hub-spoke {
    base network-slice-topology;
    description
      "Identity for Hub-and-Spoke IETF Network Slice topology.";
  }
```

```
  identity endpoint-role {
    description
      "Network Slice Endpoint Role in an IETF Network Slice topology ";
  }

  identity any-to-any-role {
    base endpoint-role;
    description
      "Network Slice Endpoint as the any-to-any role in an any-to-any
       IETF Network Slice.";
  }

  identity hub {
    base endpoint-role;
    description
      "Network Slice Endpoint as the hub role in a Hub-and-Spoke
       IETF Network Slice.";
  }

  identity spoke {
    base endpoint-role;
    description
      "Network Slice Endpoint as the spoke role in a Hub-and-Spoke
       IETF Network Slice.";
  }

  identity isolation-type {
    description
      "Base identity from which specific isolation types are derived.";
  }

  identity physical-isolation {
    base isolation-type;
    description
      "physical isolation.";
  }

  identity logical-isolation {
    base isolation-type;
    description
      "logical-isolation.";
  }

  identity network-slice-slo-metric-type {
    description
      "Base identity for Network Slice SLO metric type";
  }
```

```
  identity network-slice-match-type {
    description
      "Base identity for Network Slice traffic match type";
  }

  identity network-slice-vlan-match {
    base network-slice-match-type;
    description
      "VLAN as Network Slice traffic match criteria.";
  }

  /*
   * 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 network-slice-monitoring-type {
    type enumeration {
      enum one-way {
        description
          "represents one-way monitoring type";
      }
      enum two-way {
        description
          "represents two-way monitoring type";
      }
    }
    description
      "enumerated type of monitoring on a network-slice-member ";
  }

  /* Groupings */

  grouping status-params {
    description
      "Grouping used to join operational and administrative status";
```

```
    container status {
      description
        "Container for status of administration and operational";
      leaf admin-enabled {
        type boolean;
        description
          "Administrative Status UP/DOWN";
      }
      leaf oper-status {
        type operational-type;
        config false;
        description
          "Operations status";
      }
    }
  }

  grouping network-slice-match-criteria {
    description
      "Grouping for Network Slice match definition.";
    container network-slice-match-criteria {
      description
        "Describes Network Slice match criteria.";
      list network-slice-match-criteria {
        key "match-type";
        description
          "List of Network Slice traffic criteria";
        leaf match-type {
          type identityref {
            base network-slice-match-type;
          }
          description
            "Identifies an entry in the list of match-type for
             the Network Slice.";
        }
        leaf value {
          type string;
          description
            "Describes Network Slice match criteria,e.g. IP address,
             VLAN, etc.";
        }
      }
    }
  }

  grouping network-slice-metric-bounds {
    description
      "Network Slice metric bounds grouping";
```

```
    container network-slice-metric-bounds {
      description
        "Network Slice metric bounds container";
      list network-slice-metric-bound {
        key "metric-type";
        description
          "List of Network Slice metric bounds";
        leaf metric-type {
          type identityref {
            base network-slice-slo-metric-type;
          }
          description
            "Identifies an entry in the list of metric-types
             bound for the Network Slice.";
        }
        leaf upper-bound {
          type uint64;
          default "0";
          description
            "Upper bound on network-slice-member metric. A zero indicate
             an unbounded upper limit for the specific metric-type";
        }
      }
    }
  }

  grouping routing-protocols {
    description
      "Grouping for endpoint protocols definition.";
    container routing {
      description
        "Describes protocol between Network Slice Endpoint and IETF
         scoped network edge device.";
      container bgp {
        description
          "BGP-specific configuration.";
        leaf-list bgp-peer-ipv4 {
          type inet:ipv4-prefix;
          description
            "BGP peer ipv4 address.";
        }
        leaf-list bgp-peer-ipv6 {
          type inet:ipv6-prefix;
          description
            "BGP peer ipv6 address.";
        }
      }
      container static {
```

```
        description
          "Only applies when protocol is static.";
        leaf-list static-route-ipv4 {
          type inet:ipv4-prefix;
          description
            "ipv4 static route";
        }
        leaf-list static-route-ipv6 {
          type inet:ipv6-prefix;
          description
            "ipv6 static route";
        }
      }
    }
  }

  grouping endpoint-monitoring-parameters {
    description
      "Grouping for endpoint-monitoring-parameters.";
    container endpoint-monitoring {
      config false;
      description
        "Container for endpoint-monitoring-parameters.";
      leaf incoming-utilized-bandwidth {
        type te-types:te-bandwidth;
        description
          "Bandwidth utilization that represents the actual
           utilization of the incoming 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
          "Bandwidth utilization that represents the actual
           utilization of the incoming 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 uint32;
      units "usec";
      description
        "The latency statistics per Network Slice member.";
    }
    leaf jitter {
      type uint32 {
        range "0..16777215";
      }
      description
        "The jitter statistics per Network Slice member.";
    }
    leaf loss {
      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 geolocation {
      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 endpoint {
    description
      "IETF Network Slice endpoint related information";
    leaf endpoint-id {
      type uint32;
      description
        "unique identifier for the referred IETF Network
         Slice endpoint";
    }
    leaf endpoint-name {
      type string;
      description
        "endpoint name";
    }
    leaf-list endpoint-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 port-id {
      type string;
      description
        "Reference to the Port-id of the customer node.";
    }
    uses network-slice-match-criteria;
    leaf endpoint-ip {
      type inet:host;
      description
        "The address of the TACACS+ server.";
    }
    container bandwidth {
      container incoming-bandwidth {
        leaf guaranteed-bandwidth {
          type te-types:te-bandwidth;
          description
            "If guaranteed-bandwidth is 0, it means best effort, no
             minimum throughput is guaranteed.";
        }
        description
          "Container for the incoming bandwidth policy";
      }
      container outgoing-bandwidth {
        leaf guaranteed-bandwidth {
          type te-types:te-bandwidth;
          description
            "If guaranteed-bandwidth is 0, it means best effort, no
             minimum throughput is guaranteed.";
        }
        description
          "Container for the bandwidth policy";
      }
      description
        "Container for the bandwidth policy";
    }
    leaf mtu {
      type uint16;
      units "bytes";
      mandatory true;
```

```
      description
        "MTU of Network Slice traffic. If the traffic type is IP,
         it refers to the IP MTU. If the traffic type is Ethertype,
         will refer to the Ethernet MTU. ";
    }
    uses routing-protocols;
    uses status-params;
    uses endpoint-monitoring-parameters;
  }

  //network-slice-endpoint

  grouping network-slice-member {
    description
      "network-slice-member is described by this container";
    leaf network-slice-member-id {
      type uint32;
      description
        "network-slice-member identifier";
    }
    container src {
      description
        "the source of Network Slice link";
      leaf src-network-slice-endpoint-id {
        type leafref {
          path "/ietf-network-slices/ietf-network-slice/"
             + "network-slice-endpoint/endpoint-id";
        }
        description
          "reference to source Network Slice endpoint";
      }
    }
    container dest {
      description
        "the destination of Network Slice link ";
      leaf dest-network-slice-endpoint-id {
        type leafref {
          path "/ietf-network-slices/ietf-network-slice"
             + "/network-slice-endpoint/endpoint-id";
        }
        description
          "reference to dest Network Slice endpoint";
      }
    }
    leaf monitoring-type {
      type network-slice-monitoring-type;
      description
        "One way or two way monitoring type.";
```

```
    }
    container network-slice-member-monitoring {
      config false;
      description
        "SLO status Per network-slice endpoint to endpoint ";
      uses common-monitoring-parameters;
    }
  }

  //network-slice-member

  grouping network-slice-connection-group {
    description
      "Grouping for SLO definition of Network Slice";
    list network-slice-connection-group {
      key "connection-group-name";
      description
        "List of Network Slice connection groups, the connection group
         is used to support different SLO objectives between different
         network-slice-members in a same IETF Network slice.";
      leaf connection-group-name {
        type string;
        description
          "Identifies an entry in the list of connection group for the
           Network Slice.";
      }
      leaf default-connection-group {
        type boolean;
        default "false";
        description
          "Is the connection group is selected as the default connection
           group of a particular SLO";
      }
      choice slo-template {
        description
          "Choice for SLO template.
           Can be standard template or customized template.";
        case standard {
          description
            "Standard SLO template.";
          leaf template {
            type leafref {
              path "/ietf-network-slices"
                 + "/slice-templates/slo-template/id";
            }
            description
              "QoS template to be used.";
          }
```

```
        }
        case custom {
          description
            "Customized SLO template.";
          container network-slice-slo-policy {
            container latency {
              leaf one-way-latency {
                type uint32 {
                  range "0..16777215";
                }
                units "usec";
                description
                  "Lowest latency in micro seconds.";
              }
              leaf two-way-latency {
                type uint32 {
                  range "0..16777215";
                }
                description
                  "Lowest-way delay or latency in micro seconds.";
              }
              description
                "Latency constraint on the traffic class.";
            }
            container jitter {
              leaf one-way-jitter {
                type uint32 {
                  range "0..16777215";
                }
                description
                  "lowest latency in micro seconds.";
              }
              leaf two-way-jitter {
                type uint32 {
                  range "0..16777215";
                }
                description
                  "lowest-way delay or latency in micro seconds.";
              }
              description
                "Jitter constraint on the traffic class.";
            }
            container loss {
              leaf one-way-loss {
                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";
              }
              leaf two-way-loss {
                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";
              }
              description
                "Loss constraint on the traffic class.";
            }
            leaf availability-type {
              type identityref {
                base availability-type;
              }
              description
                "Availability Requirement for the Network Slice";
            }
            leaf isolation-type {
              type identityref {
                base isolation-type;
              }
              default "logical-isolation";
              description
                "Network Slice isolation-level.";
            }
            uses network-slice-metric-bounds;
            description
              "container for customized policy constraint on the slice
               traffic.";
          }
        }
      }
      list network-slice-member-group {
        key "network-slice-member-id";
        description
          "List of included Network Slice Member groups for the SLO.";
```

```
        leaf network-slice-member-id {
          type leafref {
            path "/ietf-network-slices/ietf-network-slice/"
               + "network-slice-member/network-slice-member-id";
          }
          description
            "Identifies the included list of Network Slice member.";
        }
      }
      container connection-group-monitoring {
        config false;
        description
          "SLO status per connection group ";
        uses common-monitoring-parameters;
      }
    }
  }

  grouping slice-template {
    description
      "Grouping for slice-templates.";
    container slice-templates {
      description
        "Container for slice-templates.";
      list slo-template {
        key "id";
        leaf id {
          type string;
          description
            "Identification of the SLO Template to be used.
             Local administration meaning.";
        }
        leaf template-description {
          type string;
          description
            "Description of the SLO template.";
        }
        description
          "List for SLO template identifiers.";
      }
    }
  }

  /* Configuration data nodes */

  container ietf-network-slices {
    description
      "IETF network-slice configurations";
```

```
    uses slice-template;
    list ietf-network-slice {
      key "network-slice-id";
      description
        "a network-slice is identified by a network-slice-id";
      leaf network-slice-id {
        type uint32;
        description
          "a unique network-slice identifier";
      }
      leaf network-slice-name {
        type string;
        description
          "network-slice name";
      }
      leaf-list network-slice-tag {
        type string;
        description
          "Network Slice tag for operational management";
      }
      leaf-list network-slice-topology {
        type identityref {
          base network-slice-topology;
        }
        default "any-to-any";
        description
          "Network Slice topology.";
      }
      uses network-slice-connection-group;
      uses status-params;
      list network-slice-endpoint {
        key "endpoint-id";
        uses endpoint;
        description
          "list of endpoints in this slice";
      }
      list network-slice-member {
        key "network-slice-member-id";
        description
          "List of network-slice-member in a slice";
        uses network-slice-member;
      }
    }
    //ietf-network-slice list
  }
}
```

<CODE ENDS>

8. 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/ietf-network-slices/ietf-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.

9. 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
```

10. 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.

11. References

11.1. Normative References

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-01 (work in progress), November 2020.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

[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>.

11.2. 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-02 (work in progress), July 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.

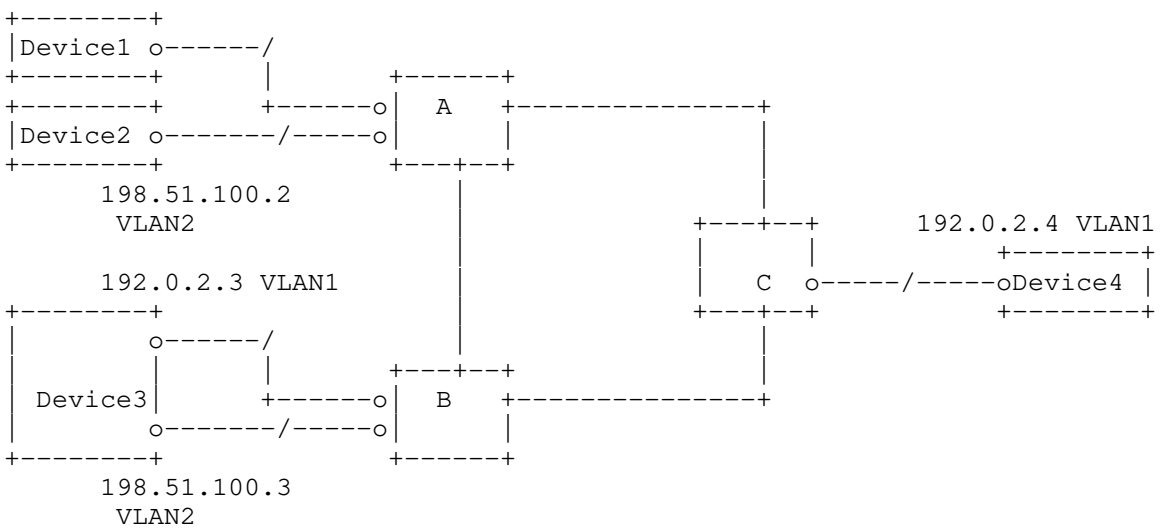
[I-D.liu-teas-transport-network-slice-yang]
Liu, X., Tantsura, J., Bryskin, I., Contreras, L., WU, Q., Belotti, S., and R. Rokui, "IETF Network Slice YANG Data Model", draft-liu-teas-transport-network-slice-yang-02 (work in progress), November 2020.

[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:

- IETF Network Slice 1 on Device1, Device3, and Device4, with any-to-any connection type
- IETF Network Slice 2 on Device2, Device3, with any-to-any connection type




```
POST: /restconf/data/ietf-network-slice:ietf-network-slices
Host: example.com
```

```
      Content-Type: application/yang-data+json

   {
     "ietf-network-slices": {
       "ietf-network-slice": [
         {
           "network-slice-id": 1,
           "network-slice-name": "slice1",
           "network-slice-topology": "any-to-any",
           "network-slice-endpoint": [
             {
              "endpoint-id": 11,
              "endpoint-name": "device1-ep1",
              "endpoint-role": "any-to-any-role",
              "network-slice-match-criteria": [
               {
                 "match-type": "network-slice-vlan-match",
                 "value": "1"
               }
              ]
             },
             {
              "endpoint-id": 12,
              "endpoint-name": "device3-ep1",
              "endpoint-role": "any-to-any-role",
              "network-slice-match-criteria": [
                {
                 "match-type": "network-slice-vlan-match",
                 "value": "1"
                }
              ]
             },
             {
               "endpoint-id": 13,
               "endpoint-name": "device4-ep1",
               "endpoint-role": "any-to-any-role",
               "network-slice-match-criteria": [
                 {
                   "match-type": "network-slice-vlan-match",
                   "value": "1"
                 }
               ]
             }
           ]
         },
         {
           "network-slice-id": 2,
           "network-slice-name": "slice2",
```

```
          "network-slice-topology": "any-to-any",
          "network-slice-endpoint": [
            {
              "endpoint-id": 21,
              "endpoint-name": "device2-ep1",
              "endpoint-role": "any-to-any-role",
              "network-slice-match-criteria": [
                {
                  "match-type": "network-slice-vlan-match",
                  "value": "2"
                }
              ]
            },
           {
             "endpoint-id": 22,
             "endpoint-name": "device3-ep2",
             "endpoint-role": "any-to-any-role",
             "network-slice-match-criteria": [
               {
                 "match-type": "network-slice-vlan-match",
                 "value": "2"
               }
             ]
           }
         ]
        }
      ]
     }
   }
```

Appendix B. Comparison with Other Possible Design choices for IETF Network Slice NBI

According to the 3.3.1. Northbound Inteface (NBI) [I-D.nsdt-teas-ns-framework], the IETF Network Slice NBI is a technology-agnostic interface, which is used for a consumer to express requirements for a particular IETF Network Slice. Consumers 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 consumer 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. 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.

- 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. 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 consumer 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:

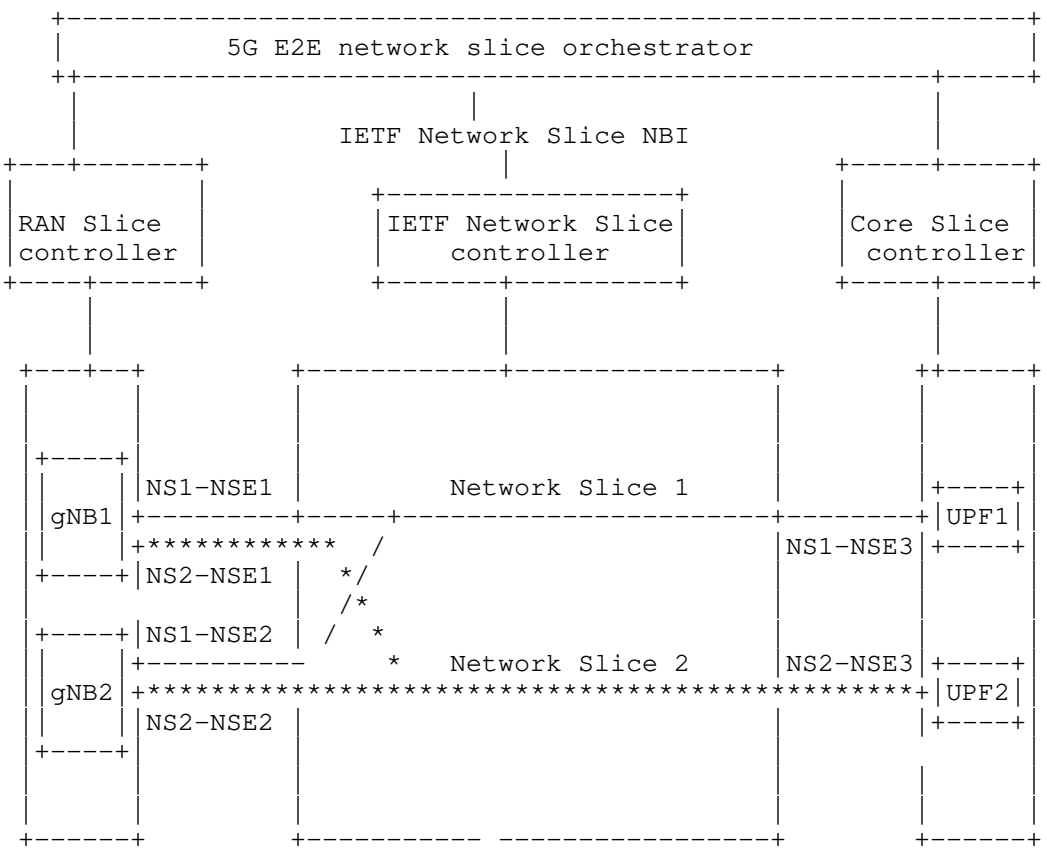
- Network: a transport network list and an list of nodes contained in the network
- Link: "links" list and "termination points" list describe how nodes in a network are connected to each other
- Support network: vertical layering relationships between IETF Network Slice networks and underlay networks

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 "port-id" to identify a Network Slice Endpont, 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

Liuyan Han
China Mobile

Email: hanliuyan@chinamobile.com

Reza Rokui
Nokia Canada

Email: reza.rokui@nokia.com