Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 19, 2019

I. Bryskin
Huawei Technologies
V. Beeram
Juniper Networks
T. Saad
Cisco Systems Inc
X. Liu
Volta Networks
Y. Lee
Huawei Technologies

October 16, 2018

Basic YANG Model for Steering Client Services To Server Tunnels
draft-bryskin-teas-service-tunnel-steering-model-00

Abstract

This document describes a YANG data model for managing pools of transport tunnels and steering client services on them.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Client layer services/signals are normally mapped onto carrying them across the network transport tunnels via client/server layer adaptation relationships. Such relationships are usually modeled as multi-layer topologies, whereas tunnels set up in underlay (server) topologies support links in respective overlay (client) topologies. In this respect having a link in a client topology means that the client layer traffic could be forwarded between link termination points (LTPs) terminating the link on opposite sides by the supporting tunnel(s) configured in the server layer topology.

This said there are numerous use cases in which describing the client service to server tunnel bindings via the topology formalism is impractical. Below are some examples of such use cases:

- o Mapping client services onto tunnels within the same network layer, for example, mapping L3 VPNs or MPLS-SR services onto IP MPLS tunnels;

- o Mapping client services onto tunnels provisioned in the highest layer topology supported by the network. For example, mapping L2VPNs or E(V)PL services onto IP MPLS tunnels provisioned in IP network;

- Mapping client services to tunnels configured in separate network layers at the network's access points. Consider, for example, an OTN/ODUk network that is used to carry client signals of, say, 20 different types (e.g. Ethernet, SDH, FKON, etc.) entering and exiting the network over client facing interfaces. Although it is possible to describe such a network as a 21-layer TE topology with the OTN/ODUk topology serving each of the 20 client layer topologies, such a description would be verbose, cumbersome, difficult to expand to accommodate additional client signals and unnecessary, because the client layer topologies would have zero switching flexibility inside the network (i.e. contain only unrelated links connecting access points across respective layer networks), and all what is required to know from the point of view of a management application is what ODUk tunnels are established or required, which client signals the tunnels could carry and at which network border nodes and how the client signals could be bound (adopted) to the tunnels.

It is worth noting that such non-topological client-service-to-server-tunnel mapping almost always happens on network border nodes. However, there are also important use cases where such a mapping is required in the middle of the network. One such use case is controlling on IP/MPLS FRR PLRs which LSPs are mapped onto which backup tunnels.

Service2tunnel mappings could be very complex: large number of instances of services of the same or different types (possibly governed by different models) could be mapped on the same set of tunnels, which could be set in different network layers and could be either TE or non-TE, P2P or P2MP or MP2MP. Furthermore, the mappings could be hierarchical: tunnels carrying services could be clients of other tunnels.

Despite of the differences of transport tunnels and of services they carry the srvice2tunnel mappings could be described in a simple uniform way. Access to a data store of such mappings could be beneficial to network management applications. It would be possible, for example, to discover which services depend on which tunnels, which services will be affected if a given tunnel goes out of service, how many more services could be placed onto a given TE tunnel without the latter violating its TE commitments (e.g. bandwidth, delay). It would be also possible to demand in a single request moving numerous (ranges of) service instances from one set of tunnels to another.

This document defines a YANG data model for such mappings.

1.1. Terminology

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 BCP 14, [RFC2119].

The following terms are defined in [RFC7950] and are not redefined here:

- augment
- data model
- data node

1.2. Tree Diagrams

A simplified graphical representation of the data model is presented in this document, by using the tree format defined in [RFC8340].

1.3. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
te-types	ietf-te-types	[I-D.ietf-teas-yang-te]

Table 1: Prefixes and Corresponding YANG Modules

2. Explicit vs. Implicit Service2tunnel Mapping. Steering Services to Transport Tunnel Pools

There are use cases in which client services require hard separation of the transport carrying them from the transport carrying other services. However, the environment in which the services may share the same transport tunnels is far more common. For this reason the model defined in this document suggests replacing (or at least augmenting) the explicit service2tunnel mapping configuration (in which the tunnels are referred to by their IDs/names) with implicit mapping. Specifically, the model introduces the notion of tunnel pool. A tunnel pool could be referred to by its network unique color and requires service2tunnel mapping configuration to specify tunnel pool color(s) instead of tunnel IDs/names. The model governs tunnel pool data store independently from the services steered on them. It is presumed (although not required) that the tunnels - components of a tunnel pool - are of the same type, provisioned using a common template and could be dynamically added to/removed from the pool without necessitating service2tunnel mapping re-configuration. Such a service to tunnel pool steering approach has the following advantages:

- Scalability and efficiency: pool component bandwidth utilization could be monitored, tunnels could be added to/removed from the pool if/when it is detected that current component bandwidth utilization has crossed certain thresholds. This allows for a very efficient network resource utilization and obviates the network management application from a very difficult task of service to tunnel mapping planning;

- Automation and elasticity: pool component attributes could be modified - bandwidth auto-adjusted, protection added, delay constrained, etc.. The tunnels could be completely or partially replaced with tunnels of different types (e.g. TE vs. non-TE, P2P vs. P2MP, etc.) or even provisioned in different network layers (OTN/ODUk tunnels replacing IP TE tunnels). All such modifications do not require service2tunell mapping re-configurations as long as the modified or new tunnels remain within the same tunnel pool(s);

- Transparency: new service sites supported by additional PEs could be added without service2tunnel mapping re-configuration.

3. The purpose of the model

To facilitate for network management applications, such as service orchestrators, managing pools of transport tunnels and steering on them client services independently of network technology/layer specifics of the services and the tunnels. The model could be applied to/implemented on physical devices, such as IP routers, as well as on abstract topology nodes. Furthermore, the model could be supported by a network (domain) controller, such as ACTN PNC, to act as a proxy server on behalf of any network element/node (physical or abstract) under its control.

4. Model Design

The data store described/governed by the model is comprised of a single top level list - TunnelPools. A TunnelPool, list element, is a container describing a set of transport tunnels (presumably with similar characteristics) identified by a network unique ID (color).

The TunnelPool container has the following fields:

- Color [uint32 list key];

- Tunnels list;

- Services list.

The Tunnels list describes the pool constituents - active transport tunnels. The list members - Tunnel containers - include the following infoemation:

- tunnel type [e.g. P2P-TE, P2MP-TE, SR-TE, SR P2P, LDP P2P, LDP MP2MP, GRE, PBB, etc]

- tunnel type specific tunnel ID [provided that a data store of the tunnel type, e.g. TE tunnels, is supported, the tunnelID allows for the management application to look up the tunnel in question to obtain detailed information about the tunnel];

- topology ID [identifies the topology over which the tunnel’s connection paths are defined];

- tunnel source topology node ID;

- tunnel destination topology node ID;

- tunnel layer ID;

- maximal and available/resolvable bandwidth;

- e2e cost;

- e2e one way and round trip delay metrics;

- tunnel protection/restoration capabilities;

- tunnel encapsulation [e.g. MPLS label stack, Ethernet STAGs, GRE header, PBB header, etc].

The Services list describes services currently steered on the tunnel pool. The list members - Service containers - have the following attributes:

- service type [e.g. fixed/transparent, L3VPN, L2VPN, EVPN, ELINE, EPL, EVPL, L1VPN, ACTN VN, etc.];

- service type specific service ID [provided that a data store of the service type, e.g. L2VPN, is supported, the service ID allows for the management application to look up the service in question to obtain detailed information about the service];

- client ports (source/destination node LTPs over which the service enters/exits the node/network, relevant only for fixed/transparent services);

- service layer ID;

- minimal bandwidth expectations;

- maximal delay expectations;

- minimal protection requirements;

- service encapsulation [e.g. MPLS label stack, Ethernet CTAGs, etc.] - for service multiplexing/de-multiplexing on/from a statistically shared tunnel].

5. Tree Structure

```
   module: ietf-tunnel-steering
     +--rw tunnel-pools
        +--rw tunnel-pool* [color]
           +--rw color          uint32
           +--rw description?   string
           +--rw service* [service-type id]
           |  +--rw service-type                  identityref
           |  +--rw id                            string
           |  +--rw access-point* [node-address link-termination-point]
           |  |  +--rw node-address              inet:ip-address
           |  |  +--rw link-termination-point    string
           |  |  +--rw direction?                enumeration
           |  +--rw switching-type?               identityref
           |  +--rw protection-type?              identityref
           |  +--rw encapsulation?                identityref
           |  +--rw performance-metric-one-way
           |  |  +--rw one-way-delay?                 uint32
           |  |  +--rw one-way-min-delay?             uint32
           |  |  +--rw one-way-max-delay?             uint32
           |  |  +--rw one-way-delay-variation?       uint32
           |  |  +--rw one-way-packet-loss?           decimal64
           |  |  +--rw one-way-residual-bandwidth?
           |  |  |       rt-types:bandwidth-ieee-float32
           |  |  +--rw one-way-available-bandwidth?
           |  |  |       rt-types:bandwidth-ieee-float32
           |  |  +--rw one-way-utilized-bandwidth?
           |  |          rt-types:bandwidth-ieee-float32
           |  +--rw performance-metric-two-way
           |     +--rw two-way-delay?             uint32
           |     +--rw two-way-min-delay?         uint32
           |     +--rw two-way-max-delay?         uint32
           |     +--rw two-way-delay-variation?   uint32
           |     +--rw two-way-packet-loss?       decimal64
           +--rw tunnel*
                   [provider-id client-id topology-id source destination
    tunnel-id]
              +--rw provider-id                   te-types:te-global-id
              +--rw client-id                     te-types:te-global-id
              +--rw topology-id
              |       te-types:te-topology-id
              +--rw tunnel-type?                  identityref
              +--rw source                        inet:ip-address
              +--rw destination                   inet:ip-address
              +--rw tunnel-id                     binary
              +--rw switching-type?               identityref
              +--rw protection-type?              identityref
              +--rw encapsulation?                identityref
              +--rw performance-metric-one-way
              |  +--rw one-way-delay?                 uint32
              |  +--rw one-way-min-delay?             uint32
              |  +--rw one-way-max-delay?             uint32
              |  +--rw one-way-delay-variation?       uint32
              |  +--rw one-way-packet-loss?           decimal64
              |  +--rw one-way-residual-bandwidth?
              |  |       rt-types:bandwidth-ieee-float32
              |  +--rw one-way-available-bandwidth?
              |  |       rt-types:bandwidth-ieee-float32
              |  +--rw one-way-utilized-bandwidth?
              |          rt-types:bandwidth-ieee-float32
              +--rw performance-metric-two-way
                 +--rw two-way-delay?             uint32
                 +--rw two-way-min-delay?         uint32
                 +--rw two-way-max-delay?         uint32
                 +--rw two-way-delay-variation?   uint32
                 +--rw two-way-packet-loss?       decimal64
```

6. YANG Modules

```
   <CODE BEGINS> file "ietf-tunnel-steering@2018-10-15.yang"
   module ietf-tunnel-steering {
     yang-version 1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-tunnel-steering";

     prefix "tnl-steer";

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

        Editors:  Igor Bryskin
                  <mailto:Igor.Bryskin@huawei.com>

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

        Editor:   Tarek Saad
                  <mailto:tsaad@cisco.com>

        Editor:   Xufeng Liu
                  <mailto:xufeng.liu.ietf@gmail.com>";

     description
       "This data model is for steering client service to server
        tunnels.

        Copyright (c) 2018 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).";

     revision 2018-10-15 {
       description "Initial revision";
       reference "TBD";
     }

     /*
      * Typedefs
      */

     /*
      * Identities
      */
     identity service-type {
       description "Base identity for client service type.";
     }
     identity service-type-l3vpn {
       base service-type;
       description
         "L3VPN service.";
     }
     identity service-type-l2vpn {
       base service-type;
       description
         "L2VPN service.";
     }
     identity service-type-evpn {
       base service-type;
       description
         "EVPN service.";
     }
     identity service-type-eline {
       base service-type;
       description
         "ELINE service.";
     }
     identity service-type-epl {
       base service-type;
       description
         "EPL service.";
     }
     identity service-type-evpl {
       base service-type;
       description
         "EVPL service.";
     }
     identity service-type-l1vpn {
       base service-type;
       description
         "L1VPN service.";
     }
     identity service-type-actn-vn {
       base service-type;
       description
         "ACTN VN service.";
     }
     identity service-type-transparent {
       base service-type;
       description
         "Transparent LAN service.";
     }

     identity encapsulation-type {
       description "Base identity for tunnel encapsulation.";
     }
     identity encapsulation-type-mpls-label {
       base encapsulation-type;
       description
         "Encapsulated by MPLS label stack.";
     }
     identity encapsulation-type-ethernet-s-tag {
       base encapsulation-type;
       description
         "Encapsulated by Ethernet S-TAG.";
     }
     identity encapsulation-type-pbb {
       base encapsulation-type;
       description
         "Encapsulated by PBB header.";
     }
     identity encapsulation-type-gre {
       base encapsulation-type;
       description
         "Encapsulated by GRE header.";
     }

     /*
      * Groupings
      */

     /*
      * Configuration data and operational state data nodes
      */
     container tunnel-pools {
       description
         "A list of mappings that steer client services to transport
          tunnel pools. The tunnel pools are managed independently from
          the services steered on them.";

       list tunnel-pool {
         key "color";
         description
           "A set of transport tunnels (presumably with similar
            characteristics) identified by a network unique ID, named
            'color'.";
         leaf color {
           type uint32;
           description
             "Unique ID of a tunnel pool.";
         }
         leaf description {
           type string;
           description
             "Client provided description of the tunnel pool.";
         }
         list service {
           key "service-type id";
           description
             "A list of client services that are steered on this tunnel
              pool.";
           leaf service-type {
             type identityref {
               base service-type;
             }
             description
               "Service type required by the client.";
           }
           leaf id {
             type string;
             description
               "Unique ID of a client service for the specified
                service type.";
           }
           list access-point {
             key "node-address link-termination-point";
             description
               "A list of client ports (Link Termination Points) for the
                service to enter or exist.";
             leaf node-address {
               type inet:ip-address;
               description
```

```
                 "Node over which the service enters or exists.";
             }
             leaf link-termination-point {
               type string;
               description
                 "Client port (Link Termination Point) over which the
                  service enters or exits.";
             }
             leaf direction {
               type enumeration {
                 enum "in" {
                   description "The service enters to the network.";
                 }
                 enum "out" {
                   description "The service exists from the network.";
                 }
                 enum "in-out" {
                   description
                     "The service enters to and exists from the
                      network.";
                 }
               }
               description
                 "Whether the service enters to or exits from the
                  network.";
             }
           }
           leaf switching-type {
             type identityref {
               base te-types:switching-capabilities;
             }
             description
               "Tunnel switching type required by the client.";
             reference "RFC3945";
           }
           leaf protection-type {
             type identityref {
               base te-types:lsp-protection-type;
             }
             description
               "The protection type required by the client.";
           }
           leaf encapsulation {
             type identityref {
               base encapsulation-type;
             }
             description
               "The encapsulation type used by the tunnel.";
```

```
          }
          uses te-types:performance-metric-container;
        }
        list tunnel {
          key "provider-id client-id topology-id source destination "
            + "tunnel-id";
          description
            "A list of tunnels in the tunnel pool.";

          leaf provider-id {
            type te-types:te-global-id;
            description
              "An identifier to uniquely identify a provider.";
          }
          leaf client-id {
            type te-types:te-global-id;
            description
              "An identifier to uniquely identify a client.";
          }
          leaf topology-id {
            type te-types:te-topology-id;
            description
              "It is presumed that a datastore will contain many
               topologies. To distinguish between topologies it is
               vital to have UNIQUE topology identifiers.";
          }
          leaf tunnel-type {
            type identityref {
              base te-types:te-tunnel-type;
            }
            description
              "Tunnel type based on constructing technologies and
               multipoint types, including P2P-TE, P2MP-TE, SR-TE,
               SR P2P, LDP P2P, LDP MP2MP, GRE, PBB, etc";
          }
          leaf source {
            type inet:ip-address;
            description
              "For a p2p or p2mp tunnel, this is the source address;
               for a mp2mp tunnel, this is the root address.";
            reference "RFC3209, RFC4875, RFC6388, RFC7582.";
          }
          leaf destination {
            type inet:ip-address;
            description
              "For a p2p tunnel, this is the tunnel endpoint address
               extracted from SESSION object;
               for a p2mp tunnel, this identifies the destination
```

```
                group, or p2mp-id;
                for a mp2mp tunnel identified by root and opaque-value,
                this value is set to '0'.";
             reference "RFC3209, RFC4875, RFC6388, RFC7582.";
           }
           leaf tunnel-id {
             type binary;
             description
               "For a p2p or p2mp tunnel, this is the tunnel identifier
                used in the SESSION that remains constant over the life
                of the tunnel;
                for a mp2mp tunnel, this is the opaque-value in the
                FEC element.";
             reference "RFC3209, RFC4875, RFC6388, RFC7582.";
           }
           leaf switching-type {
             type identityref {
               base te-types:switching-capabilities;
             }
             description "Tunnel switching type";
             reference "RFC3945";
           }
           leaf protection-type {
             type identityref {
               base te-types:lsp-protection-type;
             }
             description
               "The protection type provided by this tunnel.";
           }
           leaf encapsulation {
             type identityref {
               base encapsulation-type;
             }
             description
               "The encapsulation type used by the tunnel.";
           }
           uses te-types:performance-metric-container;
         }
       }
     }
   }
   <CODE ENDS>
```

7. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

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

```
--------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-tunnel-steering
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-tunnel-steering
namespace:    urn:ietf:params:xml:ns:yang:ietf-tunnel-steering
prefix:       tnl-steer
reference:    RFC XXXX
--------------------------------------------------------------------
```

8. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The data-model by itself does not create any security implications. The security considerations for the NETCONF protocol are applicable. The NETCONF protocol used for sending the data supports authentication and encryption.

9. References

9.1. Normative References

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

[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>.

[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>.

[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-18 (work in progress), June 2018.

[I-D.ietf-teas-yang-te] Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-16 (work in progress), July 2018.

9.2. 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>.

[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>.

Authors' Addresses

Igor Bryskin
Huawei Technologies

EMail: Igor.Bryskin@huawei.com

Vishnu Pavan Beeram
Juniper Networks

EMail: vbeeram@juniper.net

Tarek Saad
Cisco Systems Inc

EMail: tsaad@cisco.com

Xufeng Liu
Volta Networks

EMail: xufeng.liu.ietf@gmail.com

Young Lee
Huawei Technologies

EMail: leeyoung@huawei.com

TEAS Working Group
Internet-Draft
Intended status: Informational
Expires: April 21, 2019

J. Dong
S. Bryant
Huawei
Z. Li
China Mobile
T. Miyasaka
KDDI Corporation

October 18, 2018

A Framework for Enhanced Virtual Private Networks (VPN+)
draft-dong-teas-enhanced-vpn-02

Abstract

This document specifies a framework for using existing, modified and potential new networking technologies as components to provide an enhanced VPN (VPN+) service. The purpose is to enable virtual private networks (VPNs) to support the needs of new applications, particularly applications that are associated with 5G services. A network enhanced with these properties can form the underpinning of network slicing, but will also be of use in its own right.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Virtual private networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated access to a common network. The common or base network that is used to provide the VPNs is often referred to as the underlay, and the VPN is often called an overlay.

Customers of a network operator may request enhanced VPN services with additional characteristics such as complete isolation from other VPNs so that changes in network load have no effect on the throughput or latency of the service provided to the customer.

Driven largely by needs surfacing from 5G, the concept of network slicing has gained traction.[NGMN-NS-Concept] [TS23501] [TS28530] [BBF-SD406]. Network slicing requires the transport network to support partitioning 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 (and operated as) virtual networks.

Thus, there is a need to create virtual networks with enhanced characteristics. The tenant of such a virtual network can require a degree of isolation and performance that previously could only be satisfied by dedicated networks. Additionally the tenant may ask for some level of control to their virtual network e.g. to customize the service paths in the network slice.

These properties cannot be met with pure overlay networks, as they require tighter coordination and integration between the underlay and the overlay network. This document introduces a new network service called enhanced VPN (VPN+). VPN+ refers to a virtual network which has dedicated network resources allocated from the underlay network. Unlike a traditional VPN, an enhanced VPN can achieve greater isolation and guaranteed performance. These new properties, which have general applicability, may also be of interest as part of a network slicing solution.

This document specifies a framework for using existing, modified and potential new networking technologies as components to provide an enhanced VPN (VPN+) service. Specifically we are concerned with:

- The design of the enhanced data plane

- The necessary protocols in both underlay and the overlay of enhanced VPN

- The mechanisms to achieve integration between overlay and underlay

- The necessary Operation, Administration and Management (OAM) methods to instrument an enhanced VPN to make sure that the required Service Level Agreement (SLA) are met, and to take any required action to avoid SLA violation, such as switching to an alternate path

The required network layered structure to achieve this is shown in Section 3.1.

One use for enhanced VPNs is to create network slices with different isolation requirements. Such network slices may be used to provide different tenants of vertical industrial markets with their own virtual network with the explicit characteristics required. These network slices may be "hard" slices providing a high degree of confidence that the VPN+ characteristics will be maintained over the slice life cycle, or they may be "soft" slices in which case some interaction may be experienced.

2. Overview of the Requirements

In this section we provide an overview of the requirements of an enhanced VPN.

2.1. Isolation between Virtual Networks

Isolation is a feature of the services offered by a network operator where the traffic from one service instance is isolated from the traffic of other services. There are different grades of isolation that range from simple separation of traffic on delivery (ensuring that traffic is not delivered to the wrong customer) all the way to complete separation within the underlay so that the traffic from different services use distinct network resources.

We introduce the terms hard and soft isolation to cover some of these cases. A VPN has soft isolation if the traffic of one VPN cannot be inspected by the traffic of another. Both IP and MPLS VPNs are examples of soft isolated VPNs because the network delivers the traffic only to the required VPN endpoints. However, the traffic from one or more VPNs and regular network traffic may congest the network resulting in packet loss and delay for other VPNs operating normally. The ability for a VPN to be sheltered from this effect is called hard isolation, and this property is required by some critical applications. Although discussion of these isolation requirements is triggered by the needs of 5G networks, they have general utility.

The requirement is for an operator to provide both hard and soft isolation between the tenants/applications using one enhanced VPN and the tenants/applications using another enhanced VPN. Hard isolation is needed so that applications with exacting requirements can function correctly, despite other demands (perhaps a burst on another VPN) competing for the underlying resources. In practice isolation may be offered as a spectrum between soft and hard.

An example of hard isolation is a network supporting both emergency services and public broadband multi-media services. During a major incident the VPNs supporting these services would both be expected to experience high data volumes, and it is important that both make progress in the transmission of their data. In these circumstances the VPNs would require an appropriate degree of isolation to be able to continue to operate acceptably.

In order to provide the required isolation, resources may have to be reserved in the data plane of the underlay network and dedicated to traffic from a specific VPN. This may introduce scalability concerns, thus some trade-off needs to be considered to provide the required isolation between network slices while still allowing reasonable sharing inside each network slice.

An optical layer can offer a high degree of isolation, at the cost of allocating resources on a long term and end-to-end basis. Such an arrangement means that the full cost of the resources must be borne by the service that is allocated with the resources. On the other hand, where adequate isolation can be achieved at the packet layer, this permits the resources to be shared amongst many services and only dedicated to a service on a temporary basis. This in turn, allows greater statistical multiplexing of network resources and thus amortizes the cost over many services, leading to better economy. However, the degree of isolation required by network slicing cannot be entirely met with existing mechanisms such as Traffic Engineered Label Switched Paths (TE-LSPs). This is because most implementations enforce the bandwidth in the data-plane only at the PEs, but at the P routers the bandwidth is only reserved in the control plane, thus bursts of data can accidentally occur at a P router with higher than committed data rate.

There are several new technologies that provide some assistance with these data plane issues. Firstly there is the IEEE project on Time Sensitive Networking [TSN] which introduces the concept of packet scheduling of delay and loss sensitive packets. Then there is [FLEXE] which provides the ability to multiplex multiple channels over one or more Ethernet links in a way that provides hard isolation. Finally there are advanced queueing approaches which allow the construction of virtual sub-interfaces, each of which is provided with dedicated resource in a shared physical interface. These approaches are described in more detail later in this document.

In the remainder of this section we explore how isolation may be achieved in packet networks.

2.1.1. A Pragmatic Approach to Isolation

A key question is whether it is possible to achieve hard isolation in packet networks, which were never designed to support hard isolation. On the contrary, they were designed to provide statistical multiplexing, a significant economic advantage when compared to a dedicated, or a Time Division Multiplexing (TDM) network. However there is no need to provide any harder isolation than is required by the application. Pseudowires [RFC3985] emulate services that would have had hard isolation in their native form. An approximation to this requirement is sufficient in most cases.

Thus, for example, using FlexE or a channelized sub-interface together with packet scheduling as interface slicing, optionally along with the slicing of node resources, a type of hard isolation can be provided that is adequate for many VPN+ applications. Other applications may be either satisfied with a classical VPN with or without reserved bandwidth, or may need dedicated point to point fiber. The needs of each application must be quantified in order to provide an economic solution that satisfies those needs without over-engineering.

This spectrum of isolation is shown below:

```
        O==================================================O
        |          \---------------v---------------/
    Statistical                Pragmatic             Absolute
    Multiplexing               Isolation            Isolation
   (Traditional VPNs)        (Enhanced VPN)     (Dedicated Network)
```

At one end of the above figure, we have traditional statistical multiplexing technologies that support VPNs. This is a service type that has served the industry well and will continue to do so. At the opposite end of the spectrum we have the absolute isolation provided by traditional transport networks. The goal of enhanced VPN is pragmatic isolation. This is isolation that is better than is obtainable from pure statistical multiplexing, more cost effective and flexible than a dedicated network, but which is a practical solution that is good enough for the majority of applications.

2.2. Performance Guarantee

There are several kinds of performance guarantees, including guaranteed maximum packet loss, guaranteed maximum delay and guaranteed delay variation.

Guaranteed maximum packet loss is a common parameter, and is usually addressed by setting the packet priorities, queue size and discard policy. However this becomes more difficult when the requirement is combined with the latency requirement. The limiting case is zero congestion loss, and that is the goal of the Deterministic Networking work that the IETF [DETNET] and IEEE [TSN] are pursuing. In modern optical networks, loss due to transmission errors is already approaches zero, but there are the possibilities of failure of the interface or the fiber itself. This can only be addressed by some form of signal duplication and transmission over diverse paths.

Guaranteed maximum latency is required in a number of applications particularly real-time control applications and some types of virtual reality applications. The work of the IETF Deterministic Networking (DetNet) Working Group [DETNET] is relevant; however the scope needs to be extended to methods of enhancing the underlay to better support the delay guarantee, and to integrate these enhancements with the overall service provision.

Guaranteed maximum delay variation is a service that may also be needed. [I-D.ietf-detnet-use-cases] calls up a number of cases where this is needed, for example electrical utilities have an operational need for this. Time transfer is one example of a service that needs this, although it is in the nature of time that the service might be delivered by the underlay as a shared service and not provided through different virtual networks. Alternatively a dedicated virtual network may be used to provide this as a shared service.

This suggests that a spectrum of service guarantee be considered when deploying an enhanced VPN. As a guide to understanding the design requirements we can consider four types:

- Best effort

- Assured bandwidth

- Guaranteed latency

- Enhanced delivery

Best effort service is the basic service that current VPNs can provide.

An assured bandwidth service is one in which the bandwidth over some period of time is assured, this can be achieved either simply based on best effort with over-capacity provisioning, or it can be based on TE-LSPs with bandwidth reservation. The instantaneous bandwidth is however, not necessarily assured, depending on the technique used. Providing assured bandwidth to VPNs, for example by using TE-LSPs, is not widely deployed at least partially due to scalability concerns. Guaranteed latency and enhanced delivery are not yet integrated with VPNs.

A guaranteed latency service has a latency upper bound provided by the network. Assuring the upper bound is more important than achieving the minimum latency.

In Section 2.1 we considered the work of the IEEE Time Sensitive Networking (TSN) project [TSN] and the work of the IETF DetNet Working group [DETNET] in the context of isolation. The TSN and DetNet work is of greater relevance in assuring end-to-end packet latency. It is also of importance in considering enhanced delivery.

An enhanced delivery service is one in which the underlay network (at layer 3) attempts to deliver the packet through multiple paths in the hope of eliminating packet loss due to equipment or media failures.

It is these last two characteristics that an enhanced VPN adds to a VPN service.

Flex Ethernet [FLEXE] is a useful underlay to provide these guarantees. This is a method of providing time-slot based channelization over an Ethernet bearer. Such channels are fully isolated from other channels running over the same Ethernet bearer. As noted elsewhere this produces hard isolation but makes the reclamation of unused bandwidth more difficult.

These approaches can be used in tandem. It is possible to use FlexE to provide tenant isolation, and then to use the TSN/Detnet approach to provide a performance guarantee inside the a slice or tenant VPN.

2.3. Integration

A solution to the enhanced VPN problem has to provide close integration of both overlay VPN and the underlay network resource. This needs be done in a flexible and scalable way so that it can be widely deployed in operator networks to support a reasonable number of enhanced VPN customers.

Taking mobile networks and in particular 5G into consideration, the integration of network and the service functions is a likely requirement. The work in IETF SFC working group [SFC] provides a foundation for this integration.

2.4. Dynamic Configuration

Enhanced VPNs need to be created, modified, and removed from the network according to service demand. An enhanced VPN that requires hard isolation must not be disrupted by the instantiation or modification of another enhanced VPN. Determining whether modification of an enhanced VPN can be disruptive to that VPN, and in particular the traffic in flight will be disrupted can be a difficult problem.

The data plane aspects of this problem are discussed further in Section 4.

The control plane aspects of this problem are discussed further in Section 4.3.

The management plane aspects of this problem are discussed further in Section 4.4

Dynamic changes both to the VPN and to the underlay transport network need to be managed to avoid disruption to sensitive services.

In addition to non-disruptively managing the network as a result of gross change such as the inclusion of a new VPN endpoint or a change to a link, VPN traffic might need to be moved as a result of traffic volume changes.

2.5. Customized Control

In some cases it is desirable that an enhanced VPN has a customized control plane, so that the tenant of the enhanced VPN can have some control to the resources and functions allocated to this enhanced VPN. For example, the tenant may be able to specify the service paths in his own enhanced VPN. Depending on the requirement, an enhanced VPN may have its own dedicated controller, or it may be provided with an interface to a control system which is shared with a set of other tenants, or it may be provided with an interface to the control system provided by the network operator.

Further detail on this requirement will be provided in a future version of the draft.

2.6. Applicability

The technologies described in this document should be applicable to a number types of VPN services such as:

- Layer 2 point to point services such as pseudowires [RFC3985]

- Layer 2 VPNs [RFC4664]

- Ethernet VPNs [RFC7209]

- Layer 3 VPNs [RFC4364], [RFC2764]

Where such VPN types need enhanced isolation and delivery characteristics the technology described here can be used to provide an underlay with the required enhanced performance.

3. Architecture of Enhanced VPN

A number of enhanced VPN services will typically be provided by a common network infrastructure. Each enhanced VPN consists of both the overlay and a specific set of dedicated network resources and functions allocated in the underlay to satisfy the needs of the VPN tenant. The integration between overlay and various underlay resources ensures the isolation between different enhanced VPNs, and achieves the guaranteed performance for different services.

An enhanced VPN needs to be designed with consideration given to:

- A enhanced data plane

- A control plane to create enhanced VPN, making use of the data plane isolation and guarantee techniques

- A management plane for enhanced VPN service life-cycle management

These required characteristics are expanded below:

- Enhanced data plane

  * Provides the required resource isolation capability, e.g. bandwidth guarantee.

  * Provides the required packet latency and jitter characteristics

  * Provides the required packet loss characteristics

  - Provides the mechanism to identify network slice and the associated resources

- Control plane

  - Collect the underlying network topology and resources available and export this to other nodes and/or the centralized controller as required.

  - Create the required virtual networks with the resource and properties needed by the enhanced VPN services that are assigned to it.

  - Determine the risk of SLA violation and take appropriate avoiding action

  - Determine the right balance of per-packet and per-node state according to the needs of enhanced VPN service to scale to the required size

- Management plane

  - Provides the life-cycle management (creation, modification, decommissioning) of enhanced VPN

  - Provide a interface between the enhanced VPN provider and the enhanced VPN clients such that some of the operation requests can be met without interfering other enhanced VPN clients.

This document will focus on the data plane and control plane of the enhanced VPN. The details of the management plane is outside the scope of this document.

3.1. Layered Architecture

The layered architecture of enhanced VPN is shown in Figure 1.

```
                   +-------------------+              }
                   | Network Controller|              } Centralized
                   +-------------------+              }   Control
                   .    .    .     .  .
                  .     .    .     .  .
                 .      N----N----N  .                }
                .      /         /    .               }
               N-----N-----N----N-----N               }
                       N----N                         }
                      /    /  \                       }  Virtual
               N-----N----N----N-----N                } Networks
                             N----N                   }
                            /    /                    }
               N-----N-----N----N-----N               }


       +----+ ===== +----+ =====  +----+ ===== +----+  }
       +----+ ===== +----+ =====  +----+ ===== +----+  } Physical
       +----+ ===== +----+ =====  +----+ ===== +----+  } Network
       +----+       +----+        +----+       +----+  }
         N      L     N      L      N      L      N

       N = Partitioned node
       L = Partitioned link

       +----+ = Partition within a node
       +----+

       ====== = Partition within a link
```

Figure 1: The Layered Architecture

Underpinning everything is the physical infrastructure layer consisting of partitioned links and nodes which provide the underlying resources used to provision the separated virtual networks. Various components and techniques as discussed in Section 4 can be used to provide the resource partition, such as FlexE, Time Sensitive Networking, Deterministic Networking, etc. These partitions may be physical, or virtual so long as the SLA required by the higher layers is met.

These techniques can be used to provision the virtual networks with dedicated resources that they need. To get the required functionality there needs to be integration between these overlays and the underlay providing the physical resources.

The centralized controller is used to create the virtual networks, to allocate the resources to each virtual network and to provision the enhanced VPN services within the virtual networks. A distributed control plane may also be used for the distribution of the topology and attribute information of the virtual networks.

The creation and allocation process needs to take a holistic view of the needs of all of its tenants, and to partition the resources accordingly. However within a virtual network these resources can if required be managed via a dynamic control plane. This provides the required scalability and isolation.

3.2. Multi-Point to Multi-Point

At the VPN service level, the connectivity are usually mesh or partial-mesh. To support such kind of VPN service, the corresponding underlay is also an abstract MP2MP medium. However when service guarantees are provided, the point-to-point path through the underlay of the enhanced VPN needs to be specifically engineered to meet the required performance guarantees.

3.3. Application Specific Network Types

Although a lot of the traffic that will be carried over the enhanced VPN will likely be IPv4 or IPv6, the design has to be capable of carrying other traffic types, in particular Ethernet traffic. This is easily accomplished through the various pseudowire (PW) techniques [RFC3985]. Where the underlay is MPLS, Ethernet can be carried over the enhanced VPN encapsulated according to the method specified in [RFC4448]. Where the underlay is IP, Layer Two Tunneling Protocol - Version 3 (L2TPv3) [RFC3931] can be used with Ethernet traffic carried according to [RFC4719]. Encapsulations have been defined for most of the common layer two type for both PW over MPLS and for L2TPv3.

4. Candidate Technologies

A VPN is a network created by applying a multiplexing technique to the underlying network (the underlay) in order to distinguish the traffic of one VPN from that of another. A VPN path that travels by other than the shortest path through the underlay normally requires state in the underlay to specify that path. State is normally applied to the underlay through the use of the RSVP Signaling protocol, or directly through the use of an SDN controller, although other techniques may emerge as this problem is studied. This state gets harder to manage as the number of VPN paths increases. Furthermore, as we increase the coupling between the underlay and the overlay to support the enhanced VPN service, this state will increase further.

In an enhanced VPN different subsets of the underlay resources are dedicated to different enhanced VPNs. Any enhanced VPN solution thus needs tighter coupling with underlay than is the case with existing VPNs. We cannot for example share the tunnel between enhanced VPNs which require hard isolation.

A number of candidate underlay data plane solutions which can be used provide the required isolation and guarantee are described in following sections.

- FlexE

- Time Sensitive Networking

- Dedicated Queues

We then consider the problem of slice differentiation and resource representation in the network layer. The candidate technologies are:

- MPLS

- MPLS-SR

- Segment Routing over IPv6 (SRv6)

- Deterministic Networking

The considerations about the control plane is also described.

4.1. Underlay Data Plane

4.1.1. FlexE

FlexE [FLEXE] is a method of creating a point-to-point Ethernet with a specific fixed bandwidth. FlexE provides the ability to multiplex multiple channels over an Ethernet link in a way that provides hard isolation. FlexE also supports the bonding of multiple links, which can be used to create larger links out of multiple slower links in a more efficient way that traditional link aggregation. FlexE also supports the sub-rating of links, which allows an operator to only use a portion of a link. However it is a only a link level technology. When packets are received by the downstream node, they need to be processed in a way that preserves that isolation in the downstream node. This in turn requires a queuing and forwarding implementation that preserves the end-to-end isolation.

If different FlexE channels are used for different services, then no sharing is possible between the FlexE channels. This in turn means that it may be difficult to dynamically redistribute unused bandwidth to lower priority services. This may increase the cost of providing services on the network. On the other hand, FlexE can be used to provide hard isolation between different tenants on a shared interface. The tenant can then use other methods to manage the relative priority of their own traffic in each FlexE channel.

Methods of dynamically re-sizing FlexE channels and the implication for enhanced VPN is for further study.

4.1.2. Dedicated Queues

In order to provide multiple isolated virtual networks for enhanced VPN, the conventional Diff-Serv based queuing system [RFC2475] [RFC4594] is insufficient, due to the limited number of queues which cannot differentiate between traffic of different enhanced VPNs, and the range of service classes that each need to provide to their tenants. This problem is particularly acute with an MPLS underlay due to the small number of traffic class services available. In order to address this problem and reduce the interference between enhanced VPNs, it is necessary to steer traffic of VPNs to dedicated input and output queues. Routers usually have large amount of queues and sophisticated queuing systems, which could be used or enhanced to provide the levels of isolation required by the applications of enhanced VPN. For example, on one physical interface, the queuing system can provide a set of virtual sub-interfaces, each allocated with dedicated queueing and buffer resources. Sophisticated queuing systems of this type may be used to provide end-to-end virtual isolation between traffic of different enhanced VPNs.

4.1.3. Time Sensitive Networking

Time Sensitive Networking (TSN) [TSN] is an IEEE project that is designing a method of carrying time sensitive information over Ethernet. It introduces the concept of packet scheduling where a high priority packet stream may be given a scheduled time slot thereby guaranteeing that it experiences no queuing delay and hence a reduced latency. However, when no scheduled packet arrives, its reserved time-slot is handed over to best effort traffic, thereby improving the economics of the network. The mechanisms defined in TSN can be used to meet the requirements of time sensitive services of an enhanced VPN.

Ethernet can be emulated over a Layer 3 network using a pseudowire. However the TSN payload would be opaque to the underlay and thus not treated specifically as time sensitive data. The preferred method of carrying TSN over a layer 3 network is through the use of deterministic networking as explained in the following section of this document.

4.2. Network Layer

4.2.1. Deterministic Networking

Deterministic Networking (DetNet) [I-D.ietf-detnet-architecture] is a technique being developed in the IETF to enhance the ability of layer 3 networks to deliver packets more reliably and with greater control over the delay. The design cannot use re-transmission techniques such as TCP since that can exceed the delay tolerated by the applications. Even the delay improvements that are achieved with Stream Control Transmission Protocol Partial Reliability Extenstion (SCTP-PR) [RFC3758] do not meet the bounds set by application demands. DetNet pre-emptively sends copies of the packet over various paths to minimize the chance of all packets being lost, and trims duplicate packets to prevent excessive flooding of the network and to prevent multiple packets being delivered to the destination. It also seeks to set an upper bound on latency. The goal is not to minimize latency; the optimum upper bound paths may not be the minimum latency paths.

DetNet is based on flows. It currently does not specify the use of underlay topology other than the base topology. To be of use for enhanced VPN, DetNet needs to be integrated with different virtual topologies of enhanced VPNs.

The detailed design that allows the use DetNet in a multi-tenant network, and how to improve the scalability of DetNet in a multi-tenant network are topics for further study.

4.2.2. MPLS Traffic Engineering (MPLS-TE)

MPLS-TE introduces the concept of reserving end-to-end bandwidth for a TE-LSP, which can be used as the underlay of VPNs. It also introduces the concept of non-shortest path routing through the use of the Explicit Route Object [RFC3209]. VPN traffic can be run over dedicated TE-LSPs to provide reserved bandwidth for each specific connection in a VPN. Some network operators have concerns about the scalability and management overhead of RSVP-TE system, and this has lead them to consider other solutions for their networks.

4.2.3. Segment Routing

Segment Routing [RFC8402] is a method that prepends instructions to packets at the head-end node and optionally at various points as it passes though the network. These instructions allow the packets to be routed on paths other than the shortest path for various traffic engineering reasons. These paths can be strict or loose paths, depending on the compactness required of the instruction list and the degree of autonomy granted to the network, for example to support Equal Cost Multipath load-balancing (ECMP) [RFC2992].

With SR, a path needs to be dynamically created through a set of segments by simply specifying the Segment Identifiers (SIDs), i.e. instructions rooted at a particular point in the network. Thus if a path is to be provisioned from some ingress point A to some egress point B in the underlay, A is provided with a SID list from A to B and instructions on how to identify the packets to which the SID list is to be prepended.

By encoding the state in the packet, as is done in Segment Routing, per-path state is transitioned out of the network.

However, there are a number of limitations in current SR, which limit its applicability to enhanced VPNs:

- Segments are shared between different VPNs paths

- There is no reservation of bandwidth

- There is limited differentiation in the data plane.

Thus some extensions to SR are needed to provide isolation between different enhanced VPNs. This can be achieved by including a finer granularity of state in the network in anticipation of its future use by authorized services. We therefore need to evaluate the balance between this additional state and the performance delivered by the network.

With current segment routing, the instructions are used to specify the nodes and links to be traversed. However, in order to achieve the required isolation between different services, new instructions can be created which can be prepended to a packet to steer it through specific network resources and functions.

Traditionally an SR traffic engineered path operates with a granularity of a link with hints about priority provided through the use of the traffic class (TC) field in the header. However to achieve the latency and isolation characteristics that are sought by the enhanced VPN users, steering packets through specific queues and resources will likely be required. The extent to which these needs can be satisfied through existing QoS mechanisms is to be determined. What is clear is that a fine control of which services wait for which, with a fine granularity of queue management policy is needed.

Note that the concept of a queue is a useful abstraction for many types of underlay mechanism that may be used to provide enhanced isolation and latency support.

From the perspective of the control plane, and from the perspective of the segment routing, the method of steering a packet to a queue that provides the required properties is an abstraction that hides the details of the underlying implementation. How the queue satisfies the requirement is implementation specific and is transparent to the control plane and data plane mechanisms used. Thus, for example, a FlexE channel, or a time sensitive networking packet scheduling slot are abstracted to the same concept and bound to the data plane in a common manner.

We can also introduce such fine grained packet steering by specifying the queues through an SR instruction list. Thus new SR instructions may be created to specify not only which resources are traversed, but in some cases how they are traversed. For example, it may be possible to specify not only the queue to be used but the policy to be applied when enqueuing and dequeuing.

This concept could be further generalized, since as well as queuing to the output port of a router, it is possible to consider queuing data to any resource, for example:

- A network processor unit (NPU)

- A central processing unit (CPU) Core

- A Look-up engine

Both SR-MPLS and SRv6 are candidate network layer technologies for enhanced VPN. In some cases they can be supported by DetNet to meet the packet loss, delay and jitter requirement of particular service. However, currently the "pure" IP variant of DetNet [I-D.ietf-detnet-dp-sol-ip] does not support the Packet Replication, Elimination, and Re-ordering (PREOF) [I-D.ietf-detnet-architecture] functions. How to provide the DetNet enhanced delivery in an SRv6 environment needs further study.

4.3. Control Plane

Enhanced VPN would likely be based on a hybrid control mechanism, which takes advantage of the logically centralized controller for on-demand provisioning and global optimization, whilst still relies on distributed control plane to provide scalability, high reliability, fast reaction, automatic failure recovery etc. Extension and optimization to the distributed control plane is needed to support the enhanced properties of VPN+.

RSVP-TE provides the signaling mechanism of establishing a TE-LSP with end-to-end resource reservation. It can be used to bind the VPN to specific network resource allocated within the underlay, but there are the above mentioned scalability concerns.

SR does not have the capability of signaling the resource reservation along the path, nor do its currently specified distributed link state routing protocols. On the other hand, the SR approach provides a way of efficiently binding the network underlay and the enhanced VPN overlay, as it reduces the amount of state to be maintained in the network. An SR-based approach with per-slice resource reservation can easily create dedicated SR network slices, and the VPN services can be bound to a particular SR network slice. A centralized controller can perform resource planning and reservation from the controller's point of view, but this does not ensure resource reservation is actually done in the network nodes. Thus, if a distributed control plane is needed, either in place of an SDN controller or as an assistant to it, the design of the control system needs to ensure that resources are uniquely allocated in the network nodes for the correct service, and not allocated to multiple services causing unintended resource conflict.

4.4. Management Plane

The management plane mechanisms for enhanced VPN can be based on the VPN service models as defined in [RFC8299] [I-D.ietf-l2sm-l2vpn-service-model], possible augmentations and extensions to these models may be needed, which is out of the scope of this document.

Abstraction and Control of Traffic Engineered Networks (ACTN) [I-D.ietf-teas-actn-framework] specifies the SDN based architecture for the control of TE networks. The ACTN related data models such as [I-D.ietf-teas-actn-vn-yang] and [I-D.lee-teas-te-service-mapping-yang] can be applicable in the provisioning of enhanced VPN service. The details are described in [I-D.lee-rtgwg-actn-applicability-enhanced-vpn].

5. Scalability Considerations

Enhanced VPN provides the performance guaranteed services in packet networks, with the cost of introducing necessary additional states into the network. There are at least three ways of adding the state needed for VPN+:

- o Introduce the complete state into the packet, as is done in SR. This allows the controller to specify the detailed series of forwarding and processing instructions for the packet as it transits the network. The cost of this is an increase in the packet header size. The cost is also that systems will have capabilities enabled in case they are called upon by a service. This is a type of latent state, and increases as we more precisely specify the path and resources that need to be exclusively available to a VPN.

- o Introduce the state to the network. This is normally done by creating a path using RSVP-TE, which can be extended to introduce any element that needs to be specified along the path, for example explicitly specifying queuing policy. It is of course possible to use other methods to introduce path state, such as via a Software Defined Network (SDN) controller, or possibly by modifying a routing protocol. With this approach there is state per path per path characteristic that needs to be maintained over its life-cycle. This is more state than is needed using SR, but the packet are shorter.

- o Provide a hybrid approach based on using binding SIDs to create path fragments, and bind them together with SR.

Dynamic creation of a VPN path using SR requires less state maintenance in the network core at the expense of larger VPN headers on the packet. The packet size can be lower if a form of loose source routing is used (using a few nodal SIDs), and it will be lower if no specific functions or resource on the routers are specified. Reducing the state in the network is important to enhanced VPN, as it requires the overlay to be more closely integrated with the underlay than with traditional VPNs. This tighter coupling would normally mean that more state needed to be created and maintained in the network, as the state about fine granularity processing would need to be loaded and maintained in the routers. However, a segment routed approach allows much of this state to be spread amongst the network ingress nodes, and transiently carried in the packets as SIDs.

These approaches are for further study.

5.1. Maximum Stack Depth of SR

One of the challenges with SR is the stack depth that nodes are able to impose on packets [I-D.ietf-isis-segment-routing-msd]. This leads to a difficult balance between adding state to the network and minimizing stack depth, or minimizing state and increasing the stack depth.

5.2. RSVP Scalability

The traditional method of creating a resource allocated path through an MPLS network is to use the RSVP protocol. However there have been concerns that this requires significant continuous state maintenance in the network. There are ongoing works to improve the scalability of RSVP-TE LSPs in the control plane [RFC8370].

There is also concern at the scalability of the forwarder footprint of RSVP as the number of paths through an LSR grows [I-D.sitaraman-mpls-rsvp-shared-labels] proposes to address this by employing SR within a tunnel established by RSVP-TE.

6. OAM Considerations

A study of OAM in SR networks has been documented in [RFC8403].

The enhanced VPN OAM design needs to consider the following requirements:

o Instrumentation of the underlay so that the network operator can be sure that the resources committed to a tenant are operating correctly and delivering the required performance.

o Instrumentation of the overlay by the tenant. This is likely to be transparent to the network operator and to use existing methods. Particular consideration needs to be given to the need to verify the isolation and the various committed performance characteristics.

o Instrumentation of the overlay by the network provider to proactively demonstrate that the committed performance is being delivered. This needs to be done in a non-intrusive manner, particularly when the tenant is deploying a performance sensitive application

o Verification of the conformity of the path to the service requirement. This may need to be done as part of a commissioning test.

These issues will be discussed in a future version of this document.

7. Enhanced Resiliency

Each enhanced VPN has a life-cycle, and needs modification during deployment as the needs of its tenant change. Additionally, as the network as a whole evolves, there will need to be garbage collection performed to consolidate resources into usable quanta.

Systems in which the path is imposed such as SR, or some form of explicit routing tend to do well in these applications, because it is possible to perform an atomic transition from one path to another. This is a single action by the head-end changes the path without the need for coordinated action by the routers along the path. However, implementations and the monitoring protocols need to make sure that the new path is up and meet the required SLA before traffic is transitioned to it. It is possible for deadlocks arise as a result of the network becoming fragmented over time, such that it is impossible to create a new path or modify a existing path without impacting the SLA of other paths. Resolution of this situation is as much a commercial issue as it is a technical issue and is outside the scope of this document.

There are however two manifestations of the latency problem that are for further study in any of these approaches:

- The problem of packets overtaking one and other if a path latency reduces during a transition.

- The problem of the latency transient in either direction as a path migrates.

There is also the matter of what happens during failure in the underlay infrastructure. Fast reroute is one approach, but that still produces a transient loss with a normal goal of rectifying this within 50ms [RFC5654] . An alternative is some form of N+1 delivery such as has been used for many years to support protection from service disruption. This may be taken to a different level using the techniques proposed by the IETF deterministic network work with multiple in-network replication and the culling of later packets [I-D.ietf-detnet-architecture].

In addition to the approach used to protect high priority packets, consideration has to be given to the impact of best effort traffic on the high priority packets during a transient. Specifically if a conventional re-convergence process is used there will inevitably be micro-loops and whilst some form of explicit routing will protect the high priority traffic, lower priority traffic on best effort shortest paths will micro-loop without the use of a loop prevention technology. To provide the highest quality of service to high priority traffic, either this traffic must be shielded from the micro-loops, or micro-loops must be prevented.

8. Security Considerations

All types of virtual network require special consideration to be given to the isolation between the tenants. In this regard enhanced VPNs neither introduce, no experience a greater security risk than another VPN of the same base type. However, in an enhanced virtual network service the isolation requirement needs to be considered. If a service requires a specific latency then it can be damaged by simply delaying the packet through the activities of another tenant. In a network with virtual functions, depriving a function used by another tenant of compute resources can be just as damaging as delaying transmission of a packet in the network. The measures to address these dynamic security risks must be specified as part to the specific solution.

9. IANA Considerations

There are no requested IANA actions.

10. Acknowledgements

The authors would like to thank Charlie Perkins, James N Guichard and Adrian Farrel for their review and valuable comments.

11. References

11.1. 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>.

11.2. Informative References

[BBF-SD406] "BBF SD-406: End-to-End Network Slicing", 2016, <https://wiki.broadband-forum.org/display/BBF/SD-406+End-to-End+Network+Slicing>.

[DETNET] "Deterministic Networking", March , <https://datatracker.ietf.org/wg/detnet/about/>.

[FLEXE] "Flex Ethernet Implementation Agreement", March 2016, <http://www.oiforum.com/wp-content/uploads/OIF-FLEXE-01.0.pdf>.

[I-D.ietf-detnet-architecture]
Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-08 (work in progress), September 2018.

[I-D.ietf-detnet-dp-sol-ip]
Korhonen, J. and B. Varga, "DetNet IP Data Plane Encapsulation", draft-ietf-detnet-dp-sol-ip-00 (work in progress), July 2018.

[I-D.ietf-detnet-use-cases]
Grossman, E., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-19 (work in progress), October 2018.

[I-D.ietf-isis-segment-routing-msd]
Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling MSD (Maximum SID Depth) using IS-IS", draft-ietf-isis-segment-routing-msd-19 (work in progress), October 2018.

[I-D.ietf-l2sm-l2vpn-service-model]
Wen, B., Fioccola, G., Xie, C., and L. Jalil, "A YANG Data Model for L2VPN Service Delivery", draft-ietf-l2sm-l2vpn-service-model-10 (work in progress), April 2018.

[I-D.ietf-teas-actn-framework]
Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework-15 (work in progress), May 2018.

[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 ACTN VN Operation", draft-ietf-teas-actn-vn-yang-02 (work in progress), September 2018.

[I-D.lee-rtgwg-actn-applicability-enhanced-vpn]
King, D., Lee, Y., Tantsura, J., Wu, Q., and D. Ceccarelli, "Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Enhanced VPN", draft-lee-rtgwg-actn-applicability-enhanced-vpn-03 (work in progress), July 2018.

[I-D.lee-teas-te-service-mapping-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Tantsura, J., Fioccola, G., and Q. Wu, "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang-12 (work in progress), October 2018.

[I-D.sitaraman-mpls-rsvp-shared-labels]
Sitaraman, H., Beeram, V., Parikh, T., and T. Saad, "Signaling RSVP-TE tunnels on a shared MPLS forwarding plane", draft-sitaraman-mpls-rsvp-shared-labels-03 (work in progress), December 2017.

[NGMN-NS-Concept]
"NGMN NS Concept", 2016, <https://www.ngmn.org/fileadmin/user_upload/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf>.

[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>.

[RFC2764] Gleeson, B., Lin, A., Heinanen, J., Armitage, G., and A. Malis, "A Framework for IP Based Virtual Private Networks", RFC 2764, DOI 10.17487/RFC2764, February 2000, <https://www.rfc-editor.org/info/rfc2764>.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000, <https://www.rfc-editor.org/info/rfc2992>.

[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>.

[RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, DOI 10.17487/RFC3758, May 2004, <https://www.rfc-editor.org/info/rfc3758>.

[RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <https://www.rfc-editor.org/info/rfc3931>.

[RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <https://www.rfc-editor.org/info/rfc3985>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <https://www.rfc-editor.org/info/rfc4448>.

[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>.

[RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <https://www.rfc-editor.org/info/rfc4664>.

[RFC4719] Aggarwal, R., Ed., Townsley, M., Ed., and M. Dos Santos, Ed., "Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)", RFC 4719, DOI 10.17487/RFC4719, November 2006, <https://www.rfc-editor.org/info/rfc4719>.

[RFC5654] Niven-Jenkins, B., Ed., Brungard, D., Ed., Betts, M., Ed., Sprecher, N., and S. Ueno, "Requirements of an MPLS Transport Profile", RFC 5654, DOI 10.17487/RFC5654, September 2009, <https://www.rfc-editor.org/info/rfc5654>.

[RFC7209] Sajassi, A., Aggarwal, R., Uttaro, J., Bitar, N., Henderickx, W., and A. Isaac, "Requirements for Ethernet VPN (EVPN)", RFC 7209, DOI 10.17487/RFC7209, May 2014, <https://www.rfc-editor.org/info/rfc7209>.

[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>.

[RFC8370] Beeram, V., Ed., Minei, I., Shakir, R., Pacella, D., and T. Saad, "Techniques to Improve the Scalability of RSVP-TE Deployments", RFC 8370, DOI 10.17487/RFC8370, May 2018, <https://www.rfc-editor.org/info/rfc8370>.

[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>.

[RFC8403] Geib, R., Ed., Filsfils, C., Pignataro, C., Ed., and N. Kumar, "A Scalable and Topology-Aware MPLS Data-Plane Monitoring System", RFC 8403, DOI 10.17487/RFC8403, July 2018, <https://www.rfc-editor.org/info/rfc8403>.

[SFC] "Deterministic Networking", March , <https://datatracker.ietf.org/wg/sfc/about>.

[TS23501] "3GPP TS23.501", 2016, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>.

[TS28530] "3GPP TS28.530", 2016, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3273>.

[TSN] "Time-Sensitive Networking", March , <https://1.ieee802.org/tsn/>.

Authors’ Addresses

Jie Dong
Huawei

Email: jie.dong@huawei.com

Stewart Bryant
Huawei

Email: stewart.bryant@gmail.com

Zhenqiang Li
China Mobile

Email: lizhenqiang@chinamobile.com

Takuya Miyasaka
KDDI Corporation

Email: ta-miyasaka@kddi.com

TEAS Working Group
Internet Draft
Updates (if published): RFC 4872
Intended status: Standards Track

Expires: April 20, 2019

October 22, 2018

J. He
I. Busi
Huawei

GMPLS Signaling Extensions for Shared Mesh Protection
draft-he-teas-gmpls-signaling-smp-00.txt

Status of this Memo

Copyright Notice

Abstract

ITU-T Recommendation G.808.3 [G808.3] defines the generic aspects of a shared mesh protection (SMP) mechanism, where the difference between SMP and shared mesh restoration (SMR) is also identified. ITU-T Recommendation G.873.3 [G873.3] defines the protection switching operation and associated protocol for shared mesh protection (SMP) at the optical data unit (ODU) layer. RFC 7412 provides requirements for any mechanism that would be used to implement SMP in an MPLS-TP network.

This document updates RFC 4872 to provide the extensions to the Generalized Multi-Protocol Label Switching (GMPLS) signaling to support the control of the shared mesh protection.

Table of Contents

1. Introduction

RFC 4872 [RFC4872] defines extension of RSVP-TE to support shared mesh restoration (SMR) mechanism. Shared mesh restoration can be seen as a particular case of pre-planned LSP rerouting that reduces the recovery resource requirements by allowing multiple protecting LSPs to share common link and node resources. The recovery resources for the protecting LSPs are pre-reserved during the provisioning phase, and an explicit restoration signaling is required to activate (i.e., commit resource allocation at the data plane) a specific protecting LSP instantiated during the provisioning phase.

ITU-T Recommendation G.808.3 [G808.3] defines the generic aspects of a shared mesh protection (SMP) mechanism. ITU-T Recommendation G.873.3 [G873.3] defines the protection switching operation and associated protocol for shared mesh protection (SMP) at the optical data unit (ODU) layer. RFC 7412 provides requirements for any mechanism that would be used to implement SMP in an MPLS-TP network.

SMP differs from SMR in the activation/protection switching operation. The former activates a protecting LSP via the automatic protection switching (APS) protocol in the data plane when the working LSP fails, while the latter via the control plane signaling. It is therefore necessary to distinguish SMP from SMR during provisioning so that each node involved behaves appropriately in the recovery phase when activation of a protecting LSP is done.

This document updates RFC 4872 to provide the extensions to the Generalized Multi-Protocol Label Switching (GMPLS) signaling to support the control of the shared mesh protection mechanism. Only the generic aspects for signaling SMP are addressed by this document. The technology-specific aspects are expected to be addressed by other drafts.

2. Conventions used in this document

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.

In addition, the reader is assumed to be familiar with the terminology used in [RFC4872] and [RFC4426].

3. SMP Definition

ITU-T Recommendation G.808.3 [G808.3] defines the generic aspects of a shared mesh protection (SMP) mechanism. ITU-T Recommendation G.873.3 [G873.3] defines the protection switching operation and associated protocol for shared mesh protection (SMP) at the optical data unit (ODU) layer. RFC 7412 provides requirements for any mechanism that would be used to implement SMP in an MPLS-TP network.

The SMP mechanism is based on pre-computed protection transport entities that are pre-configured into the network elements. Preconfiguration here means pre-reserving resources for the protecting LSPs without activating a particular protecting LSP (e.g. in circuit networks, the cross-connects in the intermediate nodes of the protecting LSP are not pre-established). Pre-configuring but not activating the protecting LSP allows the common link and node resources in a protecting LSP to be shared by multiple working LSPs that are physically (i.e., link, node, SRLG, etc.) disjoint. Protecting LSPs are activated in response to failures of working LSPs or operator's commands by means of the APS protocol that operates in the data plane. SMP is always revertive.

SMP has a lot of similarity to SMR except that the activation in case of SMR is achieved by control plan signaling during the recovery operation while SMP is done by APS protocol in the data plane. SMP has advantages with regard to the recovery speed compared with SMR.

4. GMPLS Signaling Extension for SMP

Consider the following network topology:

```
A---B---C---D
 \         /
  E---F---G
 /         \
H---I---J---K
```

The working LSPs [A,B,C,D] and [H,I,J,K] could be protected by [A,E,F,G,D] and [H,E,F,G,K], respectively. Per [RFC3209], in order to achieve resource sharing during the signaling of these protecting LSPs, they must have the same Tunnel Endpoint Address (as part of their SESSION object). However, these addresses are not the same in this example. Similar to SMR, a new LSP Protection Type of the secondary LSP is defined as "Shared Mesh Protection" (see PROTECTION object defined in [RFC4872]) to allow resource sharing along nodes E, F, and G. In this case, the protecting LSPs are not merged (which is useful since the paths diverge at G), but the resources along E, F, G can be shared.

When a failure is detected on one of the working LSPs (say working LSP [A,B,C,D]), the switching operation for the egress node (say node A) will be triggered by an Signal Degrade (SD) or Signal Fail (SF) on the working LSP. The egress node A will send a protection switching request APS message (for example SF) to its adjacent (downstream) intermediate node (say node E) to activate setting up the corresponding protecting LSP. If the protection resource is available, Node E will send a confirmation message to the egress node A and forward the switching request APS message to its adjacent (downstream) node (say node F). When the confirmation message is received by node A and the protection resource is available, the cross-connection on node A is established. At this time the traffic is bridged to and selected from the protecting LSP at node A. The node E will wait for the confirmation message from node F, which triggers node E to set up the cross-connection for the protection transport entity being activated. If the protection resource is not available (due to failure or being used by higher priority connections), the switching will not be successful; the intermediate node may send a message to notify the end node, or keep trying until the resource is available or the switching request is cancelled. If the resource is in use by a lower priority protection entity, the lower priority service will be removed and then the intermediate node will follow the procedure as described for the case when the resource is available.

The following subsections detail how shared mesh protection can be implemented in an interoperable fashion using GMPLS RSVP-TE extensions (see [RFC3473]). This includes:

(1) the ability to identify a "secondary protecting LSP" (hereby called the "secondary LSP") used to recover another primary working LSP (hereby called the "protected LSP")

(2) the ability to associate the secondary LSP with the protected LSP

(3) the capability to include information about the resources used by the protected LSP while instantiating the secondary LSP.

(4) the capability to instantiate during the provisioning phase several secondary LSPs in an efficient manner.

(5) the capability to support activation of a secondary LSP after failure occurrence via APS protocol in the data plane.

4.1. Identifiers

To simplify association operations, both LSPs (i.e., the protected and the secondary LSPs) belong to the same session. Thus, the SESSION object MUST be the same for both LSPs. The LSP ID, however, MUST be different to distinguish between the protected LSP carrying working traffic and the secondary LSP.

A new LSP Protection Type "Shared Mesh Protection" is introduced to the LSP Flags of PROTECTION object (see [RFC4872]) to set up the two LSPs. This LSP Protection Type value is applicable to both uni- and bidirectional LSPs.

4.2. Signaling Primary LSPs

The PROTECTION object (see [RFC4872]) is included in the Path message during signaling of the primary working LSPs, with the LSP Protection Type value set to "Shared Mesh Protection".

Primary working LSPs are signaled by setting in the POTECTION object the S bit to 0, the P bit to 0, the N bit to 1 and in the ASSOCIATION object, the Association ID to the associated secondary protecting LSP_ID.

Note: N bit is set to indicate that the protection switching signaling is done via data plane.

4.3. Signaling Secondary LSPs

The PROTECTION object (see [RFC4872]) is included in the Path message during signaling of the secondary protecting LSPs, with the LSP Protection Type value set to "Shared Mesh Protection".

Secondary protecting LSPs are signaled by setting in the PROTECTION object the S bit and the P bit to 1, the N bit to 1 and in the ASSOCIATION object, the Association ID to the associated primary working LSP_ID, which MUST be known before signaling of the secondary LSP. Moreover, the Path message used to instantiate the secondary LSP SHOULD include at least one PRIMARY_PATH_ROUTE object (see [RFC4872]) that further allows for recovery resource sharing at each intermediate node along the secondary path.

With this setting, the resources for the secondary LSP SHOULD be pre-reserved, but not committed at the data plane level, meaning that the internals of the switch need not be established until explicit action is taken to activate this LSP. Activation of a secondary LSP and protection switching to the activated protecting LSP is done using APS protocol in the data plane.

After protection switching completes the protecting LSP SHOULD be signaled with the S bit set to 0 and O bit set to 1 in the PROTECTION object. At this point, the link and node resources must be allocated for this LSP that becomes a primary LSP (ready to carry normal traffic). The formerly working LSP MAY be signaled with the A bit set in the ADMIN_STATUS object (see [RFC3473]).

5. Updates to PROTECTION Object

GMPLS extension requirements for SMP introduce several updates to the Protection Object (see [RFC4872]).

5.1. New Protection Type

A new LSP protection type "Shared Mesh Protection" is added in the protection object. This LSP Protection Type value is applicable to both uni- and bidirectional LSPs.

LSP (Protection Type) Flags

0x11   Shared Mesh Protection

5.2. Other Updates

N bit and O bit in the Protection object as defined in [RFC4872] are also updated to include applicability to SMP.

Notification (N): 1 bit

When set to 1, this bit indicates that the control plane message exchange is only used for notification during protection switching. When set to 0 (default), it indicates that the control plane message exchanges are used for protection-switching purposes. The N bit is only applicable when the LSP Protection Type Flag is set to either 0x04 (1:N Protection with Extra-Traffic), or 0x08 (1+1 Unidirectional Protection), or 0x10 (1+1 Bidirectional Protection), or 0x11 (Shared Mesh Protection). The N bit MUST be set to 0 in any other case.

Operational (O): 1 bit

When set to 1, this bit indicates that the protecting LSP is carrying the normal traffic after protection switching. The O bit is only applicable when the P bit is set to 1, and the LSP Protection Type Flag is set to either 0x04 (1:N Protection with Extra-Traffic), or 0x08 (1+1 Unidirectional Protection), or 0x10 (1+1 Bidirectional Protection), or 0x11 (Shared Mesh Protection). The O bit MUST be set to 0 in any other case.

6. Security Considerations

No further security considerations than [RFC4872].

7. IANA Considerations

There are no IANA actions required.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC4426] Lang, J., Rajagopalan, B., and D. Papadimitriou, "Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification", RFC 4426, March 2006.

[RFC4872] Lang, J.P., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, May 2007.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017.

[G808.3] ITU-T, "Generic protection switching - Shared mesh protection", G.808.3, October 2012.

8.2. Informative References

[G873.3] ITU-T, "Optical transport network - Shared mesh protection", G.873.3, September 2017.

[RFC7412] Weingarten, Y., Aldrin, S., Pan, P., Ryoo, J., Mirsky, G., "Requirements for MPLS Transport Profile (MPLS-TP) Shared Mesh Protection", RFC 7412, December 2014.

Authors' Addresses

Jia He
Huawei Technologies Co.,Ltd.
F3-1B, R&D Center, Huawei Industrial Base, Bantian, Longgang
District, Shenzhen, China

Email: hejia@huawei.com

Italo Busi
Huawei

Email: italo.busi@huawei.com

TEAS Working Group
Internet Draft
Intended Status: Standard Track
Expires: March 19, 2019

Y. Lee (Editor)
Dhruv Dhody (Editor)
Huawei
D. Ceccarelli
Ericsson
Igor Bryskin
Huawei
Bin Yeong Yoon
ETRI
Qin Wu
Huawei
Peter Park
KT

September 19, 2018

A Yang Data Model for ACTN VN Operation

draft-ietf-teas-actn-vn-yang-02

Abstract

This document provides a YANG data model for the Abstraction and Control of Traffic Engineered (TE) networks (ACTN) Virtual Network Service (VNS) operation.

Status of this Memo

This Internet-Draft will expire on March 19, 2019.

Copyright Notice

Table of Contents

1. Introduction

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].

This document provides a YANG data model for the Abstraction and Control of Traffic Engineered (TE) networks (ACTN) Virtual Network Service (VNS) operation that is going to be 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 model defined in this document can also work together with other customer service models such as L3SM [RFC8299], L2SM [L2SM] and L1CSM [L1CSM] to provide a complete life-cycle service management and operations.

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

o Characteristics of Access Points (APs) that describe customer's end point characteristics;

o 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;

o Characteristics of Virtual Networks (VNs) that describe the customer's VNs in terms of VN Members comprising a VN, multi-source and/or multi-destination characteristics of 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 [TE-Topo] which provides TE network topology abstraction and management operation. Once TE-topology Model is used in triggering VN instantiation over the networks, TE-tunnel [TE-tunnel] Model will inevitably interact with TE-Topology model for setting up actual tunnels and LSPs under the tunnels.

The ACTN 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.

1.2. 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-actn-vn	[RFCXXXX]
nw	ietf-network	[RFC8345]
te-types	ietf-te-types	[TE-Tunnel]
te-topo	ietf-te-topology	[TE-TOPO]

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. ACTN CMI context

The model presented in this document has the following ACTN context.

```
                   +-------+
                   |  CNC  |
                   +-------+
                       |
                       |    VN YANG + TE-topology YANG
                       |
            +------------------------+
            |          MDSC          |
            +------------------------+
```

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:

```
                        +---------------+
               L1 ------|               |------ L4
               L2 ------|     AN 1      |------ L7
               L3 ------|               |------ L8
                        +---------------+
```

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. In some cases, more than one abstract nodes can be employed to express their VN.

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 2 VN is always built on top of a Type 1 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 on top of the Type 1 VN.

```
              +----------------------------------------------+
              |             S1               S2              |
              |              O---------------O               |
              |     ________/ \_______        \              |
              |    /                 \         \             |
              |S3 /                   \ S4      \ S5         |
        L1----|-O----------------------O---------O-----------|------L4
              |   \                     \         \          |
              |    \                     \         \         |
              |     \ S6                  \ S7      \ S8     |
              |      O     ----------------O---------O-------|------L7
              |     / \   /                 \   ____/        |
              |S9  /   \ /S10                \ /             |
       L2-----|---O-----O---------------------O--------------|------L8
              |  /                          S11              |
       L3-----|--                                            |
              |                                              |
              +----------------------------------------------+
```

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 Customer End-Point, respectively.

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

```
                        +---------------+
               L1 ------|               |------ L4
               L2 ------|     AN 1      |------ L7
               L3 ------|               |------ L8
                        +---------------+
```

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

```
              +--------+                                 +--------+
               |  CNC  |                                  |  MDSC  |
               +--------+                                 +--------+
                    |                                          |
                    |                                          |
   CNC POST TE-topo |  POST /nw:networks/nw:network/           |
   model(with Conn. |  nw:node/te-node-id/tet:connectivity-    |
   Matrix on one    |  matrices/tet:connectivity-matrix        |
   Abstract node    |----------------------------------------->|
                    |                          HTTP 200        |
                    |<-----------------------------------------|
                    |                                          |
   CNC POST the ACTN|  POST /ACTN VN                           |
   VN identifying   |----------------------------------------->| If there is
   AP, VNAP and VN- |                                          | multi-dest’n
   Members and maps |                                          | module, then
   to the TE-topo   |                          HTTP 200        | MDSC selects a
                    |<-----------------------------------------| src or dest’n
                    |                                          | and update
                    |                                          | ACTN VN YANG
   CNC GET the ACTN |  GET /ACTN VN                            |
   VN YANG status   |----------------------------------------->|
                    |                                          |
                    |  HTTP 200 (ACTN VN with status: selected |
                    |  VN-members in case of multi s-d         |
                    |<-----------------------------------------|
                    |                                          |
```

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

                  VN-Member 3       L2-L4

                  VN-Member 4       L3-L8 (via S10)
```

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

```
                            S1               S2
                             O---------------O
                    ________/ \______         \
                   /                 \         \
               S3 /                   \ S4      \ S5
        L1------O----------------------O---------O------------------L4
                  \                     \         \
                   \                     \         \
                    \ S6                  \ S7      \ S8
                     O     ----------------O---------O--------------L5
                    / \   /                 \  _____/ \_____________L6
               S9  /   \ /S10                \ /
       L2---------O-----O---------------------O---------------------L7
                 /                          S11\____________________L8
       L3--------
```

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

```
              +--------+                                +--------+
               |  CNC   |                                |  MDSC  |
               +--------+                                +--------+
                    |                                         |
                    |                                         |
   CNC POST TE-topo |  POST /nw:networks/nw:network/          |
   model(with Conn. |  nw:node/te-node-id/tet:connectivity-  |
   Matrix on one    |  matrices/tet:connectivity-matrix      |
   Abstract node and|---------------------------------------->|
   Explicit paths in|                                          |
   The conn. Matrix |                        HTTP 200          |
                    |<-----------------------------------------|
                    |                                          |
   CNC POST the ACTN|  POST /ACTN VN                           |
   VN identifying   |----------------------------------------->|
   AP, VNAP and VN- |                                          |
   Members and maps |                                          |
   to the TE-topo   |                          HTTP 200        |
                    |<-----------------------------------------|
                    |                                          |
                    |                                          |
   CNC GET the ACTN |  GET /ACTN VN                            |
   VN YANG status   |----------------------------------------->|
                    |                                          |
                    |  HTTP 200 (ACTN VN with status)          |
                    |<-----------------------------------------|
                    |                                          |
```

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

```
               +--------+                                 +--------+
               |  CNC   |                                 |  MDSC  |
               +--------+                                 +--------+
                    |                                          |
                    |                                          |
   CNC POST ACTN VN |                                          |
   Identifying AP,  |                                          |
   VNAP and VN-     |  POST /ACTN VN                           | MDSC populates
   Members          |----------------------------------------->| a single Abst.
                    |                        HTTP 200          | node topology
                    |<-----------------------------------------| by itself
                    |                                          |
   CNC POST the ACTN|  POST /ACTN VN                           |
   VN identifying   |----------------------------------------->|
   AP, VNAP and VN- |                                          |
   Members and maps |                                          |
   to the TE-topo   |                          HTTP 200        |
                    |<-----------------------------------------|
                    |                                          |
                    |                                          |
   CNC GET the ACTN |  GET /ACTN VN                            |
   VN YANG status   |----------------------------------------->|
                    |                                          |
                    |  HTTP 200 (ACTN VN with status)          |
                    |<-----------------------------------------|
                    |                                          |
```

4. ACTN 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 [ACTN-INFO]. 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 ACTN 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 ACTN 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

ACTN VN 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.

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 actn
       . . .
       +--rw vn
          +--rw vn-list* [vn-id]
             +--rw vn-id              uint32
             +--rw vn-name?           string
             +--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             uint32
             |  +--rw src
             |  |  +--rw src?            -> /actn/ap/access-point-list/access-po
int-id
             |  |  +--rw src-vn-ap-id?   -> /actn/ap/access-point-list/vn-ap/vn-
ap-id
             |  |  +--rw multi-src?      boolean {multi-src-dest}?
             |  +--rw dest
             |  |  +--rw dest?            -> /actn/ap/access-point-list/access-p
oint-
id
             |  |  +--rw dest-vn-ap-id?   -> /actn/ap/access-point-list/vn-ap/vn
-ap-id
             |  |  +--rw multi-dest?      boolean {multi-src-dest}?
             |  +--rw connetivity-matrix-id?   -> /nw:networks/network/node/tet:
te/te-
node-attributes/connectivity-matrices/connectivity-matrix/id
             |  +--ro oper-status?            identityref
             +--ro if-selected?      boolean {multi-src-dest}?
             +--rw admin-status?     identityref
             +--ro oper-status?      identityref
```

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 [TE-MAP].

The VN Yang model can be extended to support telemetry, performance monitoring and network autonomics as described in [ACTN-PM].

4.3.4. Summary

This section summarizes the innovative service features of the ACTN 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. ACTN VN YANG Model (Tree Structure)

```
   module: ietf-actn-vn
       +--rw actn
          +--rw ap
          |  +--rw access-point-list* [access-point-id]
          |     +--rw access-point-id      uint32
          |     +--rw access-point-name?   string
          |     +--rw max-bandwidth?       te-types:te-bandwidth
          |     +--rw avl-bandwidth?       te-types:te-bandwidth
          |     +--rw vn-ap* [vn-ap-id]
          |        +--rw vn-ap-id         uint32
          |        +--rw vn?              -> /actn/vn/vn-list/vn-id
          |        +--rw abstract-node?   -> /nw:networks/network/node/tet:te-no
de-id
          |        +--rw ltp?             te-types:te-tp-id
          +--rw vn
             +--rw vn-list* [vn-id]
                +--rw vn-id                 uint32
                +--rw vn-name?              string
                +--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             uint32
                |  +--rw src
                |  |  +--rw src?            -> /actn/ap/access-point-list/access
-point-id
                |  |  +--rw src-vn-ap-id?   -> /actn/ap/access-point-list/vn-ap/
vn-ap-id
                |  |  +--rw multi-src?      boolean {multi-src-dest}?
                |  +--rw dest
                |  |  +--rw dest?            -> /actn/ap/access-point-list/acces
s-point-id
                |  |  +--rw dest-vn-ap-id?   -> /actn/ap/access-point-list/vn-ap
/vn-ap-id
                |  |  +--rw multi-dest?      boolean {multi-src-dest}?
                |  +--rw connetivity-matrix-id?   -> /nw:networks/network/node/t
et:te/te-node-
   attributes/connectivity-matrices/connectivity-matrix/id
                |  +--ro oper-status?             identityref
                +--ro if-selected?          boolean {multi-src-dest}?
                +--rw admin-status?         identityref
                +--ro oper-status?          identityref
                +--rw vn-level-diversity?   vn-disjointness

     rpcs:
       +---x vn-compute
          +---w input
          |  +---w abstract-node?        -> /nw:networks/network/node/tet:te-nod
e-id
          |  +---w vn-member-list* [vn-member-id]
          |  |  +---w vn-member-id             uint32
          |  |  +---w src
          |  |  |  +---w src?            -> /actn/ap/access-point-list/access-po
int-id
          |  |  |  +---w src-vn-ap-id?   -> /actn/ap/access-point-list/vn-ap/vn-
ap-id
          |  |  |  +---w multi-src?      boolean {multi-src-dest}?
          |  |  +---w dest
          |  |  |  +---w dest?            -> /actn/ap/access-point-list/access-p
oint-id
          |  |  |  +---w dest-vn-ap-id?   -> /actn/ap/access-point-list/vn-ap/vn
-ap-id
          |  |  |  +---w multi-dest?      boolean {multi-src-dest}?
          |  |  +---w connetivity-matrix-id?   -> /nw:networks/network/node/tet:
te/te-node-
   attributes/connectivity-matrices/connectivity-matrix/id
          |  +---w vn-level-diversity?   vn-disjointness
          +--ro output
             +--ro vn-member-list* [vn-member-id]
                +--ro vn-member-id             uint32
                +--ro src
                |  +--ro src?            -> /actn/ap/access-point-list/access-po
```

```
int-id
                |  +--ro src-vn-ap-id?   -> /actn/ap/access-point-list/vn-ap/vn-
ap-id
                |  +--ro multi-src?      boolean {multi-src-dest}?
                +--ro dest
                |  +--ro dest?            -> /actn/ap/access-point-list/access-p
oint-id
                |  +--ro dest-vn-ap-id?   -> /actn/ap/access-point-list/vn-ap/vn
-ap-id
```

```
                |  +--ro multi-dest?      boolean {multi-src-dest}?
                +--ro connetivity-matrix-id?   -> /nw:networks/network/node/tet:
te/te-node-
   attributes/connectivity-matrices/connectivity-matrix/id
                +--ro if-selected?            boolean {multi-src-dest}?
                +--ro compute-status?         identityref
```

6. ACTN-VN YANG Code

The YANG code is as follows:

```
   <CODE BEGINS> file "ietf-actn-vn@2018-02-27.yang"

module ietf-actn-vn {
    namespace "urn:ietf:params:xml:ns:yang:ietf-actn-vn";
    prefix "vn";

    /* Import network */
    import ietf-network {
        prefix "nw";
    }

    /* Import TE generic types */
    import ietf-te-types {
        prefix "te-types";
    }

    /* Import Abstract TE Topology */
    import ietf-te-topology {
        prefix "tet";
    }

    organization
        "IETF Traffic Engineering Architecture and Signaling (TEAS)
        Working Group";
    contact
        "Editor: Young Lee <leeyoung@huawei.com>
               : Dhruv Dhody <dhruv.ietf@gmail.com>";
    description
        "This module contains a YANG module for the ACTN 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.";
    revision 2018-02-27 {
        description
            "initial version.";
        reference
            "TBD";
    }
    /*
     * Features
     */
    feature multi-src-dest {
        description
          "Support for selection of one src or destination
          among multiple.";
    }

    /*identity path-metric-delay {
       base te-types:path-metric-type;
       description
         "delay  path metric";
     }
     identity path-metric-delay-variation {
       base te-types:path-metric-type;
       description
         "delay-variation path metric";
     }
     identity path-metric-loss {
       base te-types:path-metric-type;
       description
         "loss path metric";
     }*/

     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-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-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-computatione-failed {
         base vn-compute-state-type;
         description
           "State path compute failed";
     }
    /*
     * Groupings
     */

     typedef vn-disjointness {
          type bits {
               bit node {
                    position 0;
                    description "node disjoint";
              }
               bit link {
                    position 1;
                    description "link disjoint";
               }
               bit srlg {
                    position 2;
                    description "srlg disjoint";
               }
          }
          description
              "type of the resource disjointness for
               VN level applied across all VN members
               in a VN";
     }

    grouping vn-ap {
        description
            "VNAP related information";
        leaf vn-ap-id {
            type uint32;
            description
                "unique identifier for the referred
                VNAP";
        }
        leaf vn {
            type leafref {
               path "/actn/vn/vn-list/vn-id";
            }
            description
                 "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";
          }
        leaf ltp {
            type te-types:te-tp-id;
            description
                "Reference LTP in the TE-topology";
        }
    }
    grouping access-point{
        description
            "AP related information";
        leaf access-point-id {
            type uint32;
            description
                "unique identifier for the referred
                access point";
        }
        leaf access-point-name {
            type string;
            description
                "ap name";
        }

        leaf max-bandwidth {
            type te-types:te-bandwidth;
            description
                "max bandwidth of the AP";
        }
        leaf avl-bandwidth {
            type te-types:te-bandwidth;
            description
                "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";
        }
    }//access-point
    grouping vn-member {
        description
            "vn-member is described by this container";
        leaf vn-member-id {
            type uint32;
            description
                "vn-member identifier";
        }
        container src
        {
            description
                "the source of VN Member";
            leaf src {
                type leafref {
                    path "/actn/ap/access-point-list/access-point-id";
                }
                description
                    "reference to source AP";
            }
            leaf src-vn-ap-id{
                type leafref {
                    path "/actn/ap/access-point-list/vn-ap/vn-ap-id";
                }
                description
                    "reference to source VNAP";
            }
            leaf multi-src {
                if-feature multi-src-dest;
                type boolean;
                description
                    "Is 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 "/actn/ap/access-point-list/access-point-id";
                }
                description
                    "reference to destination AP";
            }
            leaf dest-vn-ap-id{
                type leafref {
                    path "/actn/ap/access-point-list/vn-ap/vn-ap-id";
                }
                description
                    "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 connetivity-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
                "reference to connetivity-matrix";
        }
    }//vn-member
    /*
    grouping policy {
        description
            "policy related to vn-member-id";
        leaf local-reroute {
            type boolean;
            description
                "Policy to state if reroute
                can be done locally";
        }
        leaf push-allowed {
            type boolean;
            description
                "Policy to state if changes
                can be pushed to the customer";
        }
        leaf incremental-update {
            type boolean;
            description
                "Policy to allow only the
                changes to be reported";
        }
    }//policy
    */
      grouping vn-policy {
           description
                 "policy for VN-level diverisity";
           leaf vn-level-diversity {
                    type vn-disjointness;
                    description
                         "the type of disjointness on the VN level
                         (i.e., across all VN members)";
            }
      }
    /*
    grouping metrics-op {
        description
            "metric related information";
        list metric{
            key "metric-type";
            config false;
            description
                "The list of metrics for VN";
            leaf metric-type {
                type identityref {
                    base te-types:path-metric-type;
                }
                description
                    "The VN metric type.";
            }
            leaf value{
                type uint32;
                description
                    "The limit value";
            }
        }
    }
    */
    /*
    grouping metrics {
        description
            "metric related information";
        list metric{
            key "metric-type";
            description
                "The list of metrics for VN";
            uses te:path-metrics-bounds_config;
            container optimize{
                description
                    "optimizing constraints";
                leaf enabled{
                    type boolean;
                    description
                        "Metric to optimize";
                 }
                 leaf value{
                     type uint32;
                     description
```

```
                        "The computed value";
                }
            }
        }
    }
    */
    /*
    grouping service-metric {
        description
            "service-metric";
        uses te:path-objective-function_config;
        uses metrics;
        uses te-types:common-constraints_config;
        uses te:protection-restoration-params_config;
        uses policy;
    }//service-metric
    */
    /*
     * Configuration data nodes
     */
    container actn {
        description
            "actn is described by this container";
        container ap {
            description
                "AP configurations";
         list access-point-list {
                key "access-point-id";
                description
                    "access-point identifier";
                uses access-point{
                    description
                        "access-point information";
                }
         }
        }
        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 uint32;
                    description
                        "a unique vn identifier";
                }
                leaf vn-name {
                    type string;
                    description "vn name";
                }
                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;
                    /*uses metrics-op;*/
                    leaf oper-status {
                        type identityref {
                            base vn-state-type;
                        }
                        config false;
                        description
                            "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";
                  }
                /*
                container multi-src-dest{
                    if-feature multi-src-dest;
                    config false;
                    description
                        "The selected VN Member when multi-src
                        and/or mult-destination is enabled.";
                    leaf selected-vn-member{
                        type leafref {
                            path "/actn/vn/vn-list/vn-member-list"
                                 + "/vn-member-id";
                        }
                        description
                            "The selected VN Member along the set
                            of source and destination configured
                            with multi-source and/or multi-destination";
                    }
                }
                */
                /*uses service-metric;*/
                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
        }//vn
    }//actn
    /*
    * Notifications - TBD
    */
    /*
    * 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";
            }
            list vn-member-list{
                key "vn-member-id";
                description
                    "List of VN-members in a VN";
                uses vn-member;
            }
               uses vn-policy;
            /*uses service-metric;*/
        }
        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";
                  }
                /*uses metrics-op;*/
                leaf compute-status {
                    type identityref {
                        base vn-compute-state-type;
                    }
                    description
                        "VN-member compute state.";
                }
            }
            /*
            container multi-src-dest{
                if-feature multi-src-dest;
                description
                    "The selected VN Member when multi-src
                    and/or mult-destination is enabled.";
                leaf selected-vn-member-id{
                    type uint32;
                    description
                        "The selected VN Member-id from the
                        input";
                }
            }*/
        }
    }
}
```

<CODE ENDS>

7. JSON Example

This section provides json implementation examples as to how ACTN 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).

- o 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;

- o 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 ACTN VN YANG model also include the AP and VNAP which shows various VN using the same AP.

7.1. ACTN VN JSON

```
   {
      "actn":{
         "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:
  - o access-point-id
  - o max-bandwidth
  - o avl-bandwidth

- vn-ap:
  - o vn-ap-id
  - o vn
  - o abstract-node
  - o ltp

- vn-list
  - o vn-id
  - o vn-topology-id
  - o abstract-node

```
   - vn-member-id
        o src
        o src-vn-ap-id
        o dest
        o dest-vn-ap-id
        o 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-actn-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-actn-vn
   namespace:    urn:ietf:params:xml:ns:yang:ietf-actn-vn
   reference:    RFC XXXX (TDB)
   --------------------------------------------------------------------
```

10. Acknowledgments

The authors would like to thank Xufeng Liu for his helpful comments and valuable suggestions.

11. References

11.1. Normative References

[TE-TOPO] X. Liu, et al., "YANG Data Model for TE Topologies", work in progress: draft-ietf-teas-yang-te-topo.

[TE-tunnel] T. Saad, et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", work in progress: draft-ietf-teas-yang-te.

11.2. Informative References

[RFC7926] A. Farrel (Ed.), "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC 7926, July 2016.

[RFC8453] D. Ceccarelli and Y. Lee (Editors), "Framework for Abstraction and Control of Traffic Engineered Networks", RFC 8453, August 2018.

[TE-MAP] Y. Lee, D. Dhody, and D. Ceccarelli, "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang, work in progress.

[ACTN-PM] Y. Lee, et al., "YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics", draft-lee-teas-actn-pm-telemetry-autonomics, work in progress.

[L1CSM] G. Fioccola, Ed. & Y. Lee, Ed., "A Yang Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang, work in progress.

[L2SM] G. Fioccola, Ed., "A YANG Data Model for L2VPN Service Delivery", draft-ietf-l2sm-l2vpn-service-model, work in progress.

[RFC8299] Q. Wu, Ed., S. Litkowski, L. Tomotaki, and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, January 2018.

[RFC8309] Q. Wu, W. Cheng, and A. Farrel. "Service Models Explained", RFC 8309, January 2018.

[RFC8340] M. Bjorklund and L. Berger (Editors), "YANG Tree Diagrams", RFC 8340, March 2018.

[RFC8345] A. Clemm, et al, "A YANG Data Model for Network Topologies", RFC 8345, March 2018.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.

12. Contributors

Contributor's Addresses

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

Authors' Addresses

Young Lee (ed.)
Huawei Technologies
Email: leeyoung@huawei.com

Dhruv Dhody (ed.)
Huawei Technologies
Email: dhruv.ietf@gmail.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden
Email: daniele.ceccarelli@ericsson.com

Igor Bryskin
Huawei
Email: Igor.Bryskin@huawei.com

Bin Yeong Yoon
ETRI
Email: byyun@etri.re.kr

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

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

TEAS WG

Internet Draft
Intended status: Informational

Expires: February 23, 2019

Young Lee
Haomian Zheng
Huawei

Daniele Ceccarelli
Ericsson

Bin Yeong Yoon
ETRI

Oscar Gonzalez de Dios
Telefonica

Jong Yoon Shin
SKT

Sergio Belotti
Nokia

August 22, 2018

Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks

draft-ietf-teas-actn-yang-02

Status of this Memo

Copyright Notice

Abstract

Abstraction and Control of TE Networks (ACTN) refers to the set of virtual network operations needed to orchestrate, control and manage large-scale multi-domain TE networks, so as to facilitate network programmability, automation, efficient resource sharing, and end-to-end virtual service aware connectivity and network function virtualization services.

This document explains how the different types of YANG models defined in the Operations and Management Area and in the Routing Area are applicable to the ACTN framework. This document also shows how the ACTN architecture can be satisfied using classes of data model that have already been defined, and discusses the applicability of specific data models that are under development. It also highlights where new data models may need to be developed.

Table of Contents

1. Introduction

Abstraction and Control of TE Networks (ACTN) describes a method for operating a Traffic Engineered (TE) network (such as an MPLS-TE network or a layer 1 transport network) to provide connectivity and virtual network services for customers of the TE network. The services provided can be tuned to meet the requirements (such as traffic patterns, quality, and reliability) of the applications hosted by the customers. More details about ACTN can be found in Section 2.

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].

This document shows how the ACTN architecture can be satisfied using various classes of data model that have already been defined, and discusses the applicability of specific data models that are under development. It also highlights where new data models may need to be developed.

2. Abstraction and Control of TE Networks (ACTN) Architecture

[ACTN-Frame] describes the architecture model for ACTN including the entities (Customer Network Controller (CNC), Multi-domain Service

Coordinator (MDSC), and Provisioning Network Controller (PNC)) and their interfaces.

Figure 1 depicts a high-level control and interface architecture for ACTN and is a reproduction of Figure 3 from [ACTN-Frame]. A number of key ACTN interfaces exist for deployment and operation of ACTN-based networks. These are highlighted in Figure 1 (ACTN Interfaces) below:

```
                +--------------+        +---------------+        +--------------
+
                |    CNC-A     |        |     CNC-B     |        |     CNC-C
|
                |(DC provider) |        |     (ISP)     |        |     (MVNO)
|
                +--------------+        +---------------+        +--------------
+
                     \                          |                           /
      Business        \                         |                          /
      Boundary  =======\========================|=========================/=====
==
      Between           \                       | CMI                    /
      Customer &         -----------            |          --------------
      Network Operator              \           |         /
                                   +-----------------------+
                                   |          MDSC         |
                                   +-----------------------+
                                    /           |         \
                        ------------            |MPI       ----------------
                       /                        |                          \
                  +-------+                 +-------+                   +-------
+
                  |  PNC  |                 |  PNC  |                   |  PNC
|
                  +-------+                 +-------+                   +-------
+
                     | GMPLS               /      |                      /   \
                     | trigger            /       |SBI              SBI /     \
                  --------           -----        |                    /       \
                 (        )         (     )       |                   /
\
                -         -        ( Phys. )      |                  /       ---
--
                (  GMPLS   )        ( Net )       |                 /       (
  )
               (  Physical  )        ----         |                /       ( Phy
s. )
                (  Network )                   -----        -----           ( Ne
t )
                 -        -                   (     )      (     )           ---
--
                  (       )                  (  Phys. )   (  Phys. )
                  --------                    ( Net )      ( Net )
                                               -----        -----
```

Figure 1 : ACTN Interfaces

The interfaces and functions are described below (without modifying the definitions) in [ACTN-Frame]:

- The CNC-MDSC Interface (CMI) is an interface between a CNC and an MDSC. This interface is used to communicate the service request or application demand. A request will include specific service properties, for example, services type, bandwidth and constraint information. These constraints SHOULD be measurable by MDSC and therefore visible to CNC via CMI. The CNC can also request the creation of the virtual network service based on underlying physical resources to provide network services for the applications. The CNC can provide the end-point information/characteristics together with traffic matrix specifying specific customer constraints. The MDSC may also report potential network topology availability if queried for current capability from the Customer Network Controller. Performance monitoring is also applicable in CMI, which enables the MDSC to report network parameters/telemetries that may guide the CNC to create/change their services.

- The MDSC-PNC Interface (MPI) is an interface between an MDSC and a PNC. It allows the MDSC to communicate requests to create/delete connectivity or to modify bandwidth reservations in the physical network. In multi-domain environments, each PNC is responsible for a separate domain. The MDSC needs to establish multiple MPIs, one for each PNC and perform coordination between them to provide cross-domain connectivity. MPI plays an important role for multi-vendor operations; inter-operability can be achieved by standardized interface modules.

- The South-Bound Interface (SBI) is the provisioning interface for creating forwarding state in the physical network, requested via the PNC. The SBI is not in the scope of ACTN, however, it is included in this document so that it can be compared to models in [Service-Yang].

3. Service Models

[RFC8309] introduces a reference architecture to explain the nature and usage of service YANG models in the context of service orchestration. Figure 2 below depicts this relationship and is a reproduction of Figure 2 from [RFC8309]. Four models depicted in Figure 2 are defined as follows:

- Customer Service Model: A customer service model is used to describe a service as offer or delivered to a customer by a network operator.
- Service Delivery Model: A service delivery model is used by a network operator to define and configure how a service is provided by the network.

- . Network Configuration Model: A network configuration model is used by a network orchestrator to provide network-level configuration model to a controller.
- . Device Configuration Model: A device configuration model is used by a controller to configure physical network elements.

```
                                                 Customer
                            ------------------   Service  ----------
                           |                  |  Model   |          |
                           |     Service      |<-------->| Customer |
                           |   Orchestrator   |          |          |
                           |                  |           ----------
                            ------------------
                              .            .              -----------
                             .              .      ......|Application|
                            .                .     :     |  BSS/OSS  |
                           .                  .    :      -----------
                          .  Service Delivery  .   :
                          .       Model        .   :
                 ------------------    ------------------
                |                  |  |                  |
                |     Network      |  |     Network      |
                |   Orchestrator   |  |   Orchestrator   |
                |                  |  |                  |
                .------------------    ------------------.
               .         :                       :        .
              .          : Network Configuration :         .
              .          :        Model          :         .
      ------------     ------------     ------------    ------------
     |            |   |            |   |            |  |            |
     | Controller |   | Controller |   | Controller |  | Controller |
     |            |   |            |   |            |  |            |
      ------------     ------------     ------------    ------------
         :              .       .                 :            :
         :             .         .      Device    :            :
         :            .           . Configuration :            :
         :            .           .     Model     :            :
     ---------     ---------   ---------     ---------      ---------
    | Network |   | Network | | Network |   | Network |    | Network |
    | Element |   | Element | | Element |   | Element |    | Element |
     ---------     ---------   ---------     ---------      ---------
```

Figure 2: An SDN Architecture with a Service Orchestrator

4. Service Model Mapping to ACTN

YANG models coupled with the RESTCONF/NETCONF protocol [RFC6241][RFC8040] provides solutions for the ACTN framework. This section explains which types of YANG models apply to each of the ACTN interfaces.

Refer to Figure 5 of [ACTN-Frame] for details of the mapping between ACTN functions and service models. In summary, the following mappings are held between and Service Yang Models and the ACTN interfaces.

- Customer Service Model <-> CMI
- Network Configuration Model <-> MPI
- Device Configuration Model <-> SBI

4.1. Customer Service Models in the ACTN Architecture (CMI)

Customer Service Models, which are used between a customer and a service orchestrator as in [Service-YANG], should be used between the CNC and MDSC (e.g., CMI) serving as providing a simple intent-like model/interface.

Among the key functions of Customer Service Models on the CMI is the service request. A request will include specific service properties, including: service type and its characteristics, bandwidth, constraint information, and end-point characteristics.

The following table provides a list of functions needed to build the CMI. They are mapped with Customer Service Models.

Function	Yang Model
VN Service Request	[ACTN-VN-YANG]
VN Computation Request	[ACTN-VN-YANG]*
TE & Service Mapping	[TE-Service-Mapping]**
VN Performance Monitoring Telemetry	[ACTN-PM-Telemetry]***
Topology Abstraction	[TE-topology]****
Layer 1 Connectivity Service Model	[L1CSM]
Layer 2 VPN Service Model	[L2SM]
Layer 3 VPN Service Model	[RFC8299]

*VN computation request in the CMI context means network path computation request based on customer service connectivity request constraints prior to the instantiation of a VN creation.

**[TE-Service-Mapping] provides a mapping and cross-references between service models (e.g., L3SM, L2SM, L1CSM) and TE models via [ACTN-VN-YANG] and [TE-topology]. This model can be used as either Customer Service Models, or Service Delivery model described in Section 4.2.

***[ACTN-PM-Telemetry] describes performance telemetry for e2e tunnels and VNs. This module also allows autonomic traffic engineering scaling intent configuration mechanism on both the e2e tunnel and the VN level. Scale in/out criteria might be used for network automation in order the controller to react to a certain set of variations in monitored parameters. Moreover, this module also provides mechanism to define aggregated telemetry parameters as a grouping of underlying Tunnel and VN level telemetry parameters.

****TE-Topology’s Connectivity Matrices/Matrix construct can be used to instantiate VN Service via a suitable referencing and mapping with [ACTN-VN-YANG].

4.2. Service Delivery Models in ACTN Architecture

The Service Delivery Models where the service orchestration and the network orchestration could be implemented as separate components as seen in [RFC8309]. On the other hand, from an ACTN architecture point of view, the service delivery model between the service orchestrator and the network orchestrator is an internal interface between sub-components of the MDSC in a single MDSC model.

In the MDSC hierarchical model where there are multiple MDSCs, the interface between the top MDSC and the bottom MDSC can be mapped to service delivery models.

4.3. Network Configuration Models in ACTN Architecture (MPI)

The Network Configuration Models is used between the network orchestrator and the controller in [Service-YANG]. In ACTN, this model is used primarily between a MDSC and a PNC. The Network Configuration Model can be also used for the foundation of more advanced models, like hierarchical MDSCs (see Section 4.5)

The Network Configuration Model captures the parameters which are network wide information.

The following table provides a list of functions needed to build the MPI. They are mapped with Network Configuration Yang Models. Note that various Yang models are work in progress.

Function	Yang Model
Configuration Scheduling	[Schedule]
Path computation	[PATH_COMPUTATION-API]
Tunnel/LSP Provisioning	[TE-Tunnel]
Topology Abstraction	[TE-topology]
Client Signal Description	[Client-signal]
Service Provisioning	TBD*
OTN Topology Abstraction	[OTN-topo]
WSON Topology Abstraction	[WSON-topo]
Flexi-grid Topology Abstraction	[Flexi-topo]
Microwave Topology Abstraction	[MW-topo]
OTN Tunnel Model	[OTN-Tunnel]
WSON TE Tunnel Model	[WSON-Tunnel]
Flexi-grid Tunnel Model	[Flexigrid-Tunnel]

* This function needs to be investigated further. This can be a part of [TE-Tunnel] which is to be determined. Service provisioning is an optional function that builds on top the path provisioning one.

[TE-topo-tunnel] provides tutorials for the clarification and example usage for TE topology model [TE-topology] and TE tunnel model [TE-Tunnel]. [T-NBI Applicability] provides a summary on the applicability of existing YANG model usage in the current network configuration, especially for transport network.

4.4. Device Models in ACTN Architecture (SBI)

Note that SBI is not in the scope of ACTN, as there is already mature protocol solutions for various purpose on the device level of ACTN architecture, such as RSVP-TE, OSPF-TE and so on. The interworking of such protocols and ACTN controller hierarchies can be found in [gmpls-controller-inter-work].

For the device YANG models are used for per-device configuration purpose, they can be used between the PNC and the physical network/devices. One example of Device Models is ietf-te-device yang module defined in [TE-tunnel].

5. Examples of Using Different Types of YANG Models

This section provides some examples on the usage of IETF YANG models in the network operation. A few typical generic scenarios are involved. In [T-NBI Applicability], there are more transport-related scenarios and examples.

5.1. Topology Collection

Before any connection is requested and delivered, the controller needs to understand the network topology. The topology information is exchanged among controllers with topology models, such as [te-topology]. Moreover, technology-specific topology reporting may use the model described in [OTN-topo] [WSON-topo], and [Flexi-topo] for OTN, WSON and Flexi-grid, respectively. By collecting the network topology, each controller can therefore construct a local database, which can be used for the further service deployment.

There can be different types of abstraction applied between each pair of controllers, corresponding method can be found in [ACTN-frame]. The technology-specific features may be hidden after abstraction, to make the network easier for the user to operate.

When there is a topology change in the physical network, the PNC should report the change to upper level of controllers via updating messages using topology models. Accordingly, such changes is propagated between different controllers for further synchronization.

5.2. Connectivity over Two Nodes

The service models, such as described in [RFC8299], [L2SM] and [L1CSM] provide a connectivity service model which can be used in connection-oriented networks.

It would be used as follows in the ACTN architecture:

- A CNC uses the service models to specify the two client nodes that are to be connected, and also indicates the amount of traffic (i.e., the bandwidth required) and payload type. What may be additionally specified is the SLA that describes the required quality and resilience of the service.

- The MDSC uses the information in the request to pick the right network (domain) and also to select the provider edge nodes corresponding to the customer edge nodes.

  If there are multiple domains, then the MDSC needs to coordinate across domains to set up network tunnels to deliver a service. Thus coordination includes, but is not limited to, picking the right domain sequence to deliver a service.

  Additionally, an MDSC can initiate the creation of a tunnel (or tunnel segment) in order to fulfill the service request from CNC based on path computation upon the overall topology information it synthesized from different PNCs. The based model that can cater this purpose is the TE tunnel model specified in [te-tunnel]. Technology-specific tunnel configuration may use the model described in [OTN-Tunnel] [WSON-Tunnel], and [Flexigrid-Tunnel] for OTN, WSON and Flexi-grid, respectively.

- Then, the PNCs need to decide the explicit route of such a tunnel or tunnel segment (in case of multiple domains) for each domain, and then create such a tunnel using protocols such as PCEP and RSVP-TE or using per-hop configuration.

5.3. VN Service Example

The service model defined in [ACTN-VN-YANG] describes a virtual network (VN) as a service which is a set of multiple connectivity services:

- A CNC will request VN to the MDSC by specifying a list of VN members. Each VN member specifies either a single connectivity service, or a source with multiple potential destination points in the case that the precise destination sites are to be determined by MDSC.

  - In the first case, the procedure is the same as the connectivity service, except that in this case, there is a list of connections requested.

  - In the second case, where the CNC requests the MDSC to select the right destination out of a list of candidates, the MDSC needs to evaluate each candidate and then choose the best one and reply with the chosen destination for a given VN member. After this is selected, the connectivity request setup procedure is the same as in the connectivity example in section 5.2.

After the VN is set up, a successful reply message is sent from MDSC to CNC, indicating the VN is ready. This message can also be achieved by using the model defined in [ACTN-VN-YANG].

5.4. Data Center-Interconnection Example

This section describes more concretely how existing YANG models described in Section 4 map to an ACTN data center interconnection use case. Figure 3 shows a use-case which shows service policy-driven Data Center selection and is a reproduction of Figure A.1 from [ACTN-Info].

```
                             +----------------+
                             |       CNC      |
                             |   (Global DC   |
                             |   Operation    |
                             |    Control)    |
                             +--------+-------+
                                      | |   VN Requirement/Policy:
                    CMI:              | |  - Endpoint/DC location info
                 Service model        | |  - Endpoint/DC dynamic
                                      | |    selection policy
                                      | |    (for VM migration, DR, LB)
                                      | v
                            +---------+---------+
                            |  Multi-domain     | Service policy-driven
                            |Service Coordinator| dynamic DC selection
              MPI:          +-----+---+---+-----+
    Network Configuration         |   |   |
    Model                         |   |   |
                 +----------------+   |   +---------------+
                 |                    |                   |
           +-----+-----+       +------+-----+      +------+-----+
           |  PNC for  |       |  PNC for   |      |  PNC for   |
           | Transport |       | Transport  |      | Transport  |
           | Network A |       | Network B  |      | network C  |
           +-----------+       +------------+      +------------+
   Device        |                    |                   |
   Model         |                    |                   |
                 |                    |                   |
+---+      ------               ------              ------       +---+
|DC1|--////      \\\\       ////      \\\\      ////      \\\\---+DC5|
+---+ |              |     |              |    |              |  +---+
      |     TN A     +-----+     TN B     +----+     TN C     |
      /              |     |              |    |              |
     / \\\\      ////     / \\\\      ////      \\\\      ////
   +---+   ------        /      ------    \         ------ \
   |DC2|                /                  \                \+---+
   +---+               /                    \                |DC6|
                     +---+                   \ +---+         +---+
                     |DC3|                    \|DC4|
                     +---+                     +---+

                                                DR: Disaster Recovery
                                                LB: Load Balancing
```

Figure 3: Service Policy-driven Data Center Selection

Figure 3 shows how VN policies from the CNC (Global Data Center Operation) are incorporated by the MDSC to support multi-destination applications. Multi-destination applications refer to applications in which the selection of the destination of a network path for a given source needs to be decided dynamically to support such applications.

Data Center selection problems arise for VM mobility, disaster recovery and load balancing cases. VN's policy plays an important role for virtual network operation. Policy can be static or dynamic. Dynamic policy for data center selection may be placed as a result of utilization of data center resources supporting VMs. The MDSC would then incorporate this information to meet the objective of this application.

5.4.1. CMI (CNC-MDSC Interface)

[ACTN-VN-YANG] is used to express the definition of a VN, its VN creation request, the service objectives (metrics, QoS parameters, etc.), dynamic service policy when VM needs to be moved from one Data Center to another Data Center, etc. This service model is used between the CNC and the MDSC (CMI). The CNC in this use-case is an external entity that wants to create a VN and operates on the VN.

5.4.2. MPI (MDSC-PNC Interface)

The Network Configuration Model is used between the MDSC and the PNCs. Based on the Customer Service Model's request, the MDSC will need to translate the service model into the network configuration model to instantiate a set of multi-domain connections between the prescribed sources and the destinations. The MDSC will also need to dynamically interact with the CNC for dynamic policy changes initiated by the CNC. Upon the determination of the multi-domain connections, the MDSC will need to use the network configuration model such as [TE-Tunnel] to interact with each PNC involved on the path. [TE-Topology] is used to for the purpose of underlying domain network abstraction from the PNC to the MDSC.

5.4.3. SBI (Southbound interface between PNC and devices)

The Device Model can be used between the PNC and its underlying devices that are controlled by the PNC. The PNC will need to trigger signaling using any mechanisms it employees (e.g. [RSVP-TE-YANG]) to provision its domain path segment. There can be a plethora of choices how to control/manage its domain network. The PNC is responsible to abstract its domain network resources and update it to the MDSC. Note that this interface is not in the scope of ACTN. This section is provided just for an illustration purpose.

6. Security

This document is an informational draft. When the models mentioned in this draft are implemented, detailed security consideration will be given in such work.

How security fits into the whole architecture has the following components:

- the use of Restconf security between components

- the use of authentication and policy to govern which services can be requested by different parties.

- how security may be requested as an element of a service and mapped down to protocol security mechanisms as well as separation (slicing) of physical resources)

7. Acknowledgements

We thank Adrian Farrel for providing useful comments and suggestions for this draft.

8. References

8.1. Informative References

[RFC8309] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", RFC 8309.

[RFC8199] D. Bogdanovic, B. Claise, and C. Moberg, "YANG Module Classification", RFC 8199.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.

[RFC8040] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol", RFC 8040.

[ACTN-Frame] D. Ceccarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework, work in progress.

[TE-Topology] X. Liu, et. al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.

[TE-Tunnel] T. Saad (Editor), "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.

[ACTN-VN-YANG] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

[L1CSM] G. Fioccola, K. Lee, Y. Lee, D. Dhody, O. Gonzalez de-Dios, D. Ceccarelli, "A Yang Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang, work in progress.

[L2SM] B. Wen, G. Fioccola, C. Xie, L. Jalil, "A YANG Data Model for L2VPN Service Delivery", draft-ietf-l2sm-l2vpn-service-model, work in progress.

[RFC8299] Q. Wu, S. Litkowski, L. Tomotaki, K.Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC8299.

[ACTN-Info] Y. Lee & S. Belotti, "Information Model for Abstraction and Control of TE Networks (ACTN)", draft-ietf-teas-actn-info, work in progress.

[PATH-COMPUTATION-API] I.Busi/S.Belotti et al. "Path Computation API", draft-ietf-teas-yang-path-computation, work in progress

[RSVP-TE-YANG] T. Saad (Editor), "A YANG Data Model for Resource Reservation Protocol (RSVP)", draft-ietf-teas-yang-rsvp, work in progress.

[Schedule] X. Liu, et. al., "A YANG Data Model for Configuration Scheduling", draft-liu-netmod-yang-schedule, work in progress.

[OTN-topo] H. Zheng, et. al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang, work in progress.

[WSON-topo] Y. Lee, et. al., "A Yang Data Model for WSON Optical Networks", draft-ietf-ccamp-wson-yang, work in progress.

[Flexi-topo] J.E. Lopez de Vergara, et. al., "YANG data model for Flexi-Grid Optical Networks", draft-vergara-ccamp-flexigrid-yang, work in progress.

[MW-topo] M. Ye, et. al., "A YANG Data Model for Microwave Topology", draft-ye-ccamp-mw-topo-yang, work in progress.

[OTN-Tunnel] H. Zheng, et. al., "OTN Tunnel YANG Model", draft-ietf-ccamp-otn-tunnel-model, work in progress.

[ACTN-PM-Telemetry] Y. Lee, D. Dhody, S. Karunanithi, R. Vilalta, D. King, and D. Ceccarelli, "YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics", draft-lee-teas-actn-pm-telemetry-autonomics, work in progress.

[WSON-Tunnel] Y. Lee, D. Dhody, V. Lopez, D. King, B. Yoon, and R. Vilalta, "A Yang Data Model for WSON Tunnel", draft-ietf-ccamp-wson-tunnel-model, work in progress.

[Flexigrid-Tunnel] J. Vergara, D. Perdices, V. Lopez, O. Gonzalez de Dios, D. King, Y. Lee, and G. Galimberti, "YANG data model for Flexi-Grid media-channels", draft-ietf-ccamp-flexigrid-media-channel-yang, work in progress.

[TE-Service-Mapping] Y. Lee, et al, "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang, work in progress.

[Client-signal] H. Zheng, et al, "A YANG Data Model for Optical Transport Network Client Signals", draft-zheng-ccamp-client-signal-yang, work in progress.

[TE-topo-Tunnel] I.Bryskin, et. al., "TE Topology and Tunnel Modeling for Transport Networks", draft-ietf-teas-te-topo-and-tunnel-modeling, work in progress.

[T-NBI Applicability] I. Busi, et al, "Transport Northbound Interface Applicability Statement and Use Cases", draft-ietf-ccamp-transport-nbi-app-statement, work in progress.

[gmpls-controller-inter-work] H. Zheng, et al, "Interworking of GMPLS Control and Centralized Controller System", draft-zheng-ccamp-gmpls-controller-inter-work, work in progress.

9. Contributors

Contributor's Addresses

Dhruv Dhody
Huawei Technologies

Email: dhruv.ietf@gmail.com

Xian Zhang
Huawei Technologies

Email: zhang.xian@huawei.com

Authors' Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive
Plano, TX 75023, USA
Phone: (469)277-5838

Email: leeyoung@huawei.com

Haomian Zheng
Huawei Technologies

Email: zhenghaomian@huawei.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Bin Yeong Yoon
ETRI

Email: byyun@etri.re.kr

Oscar Gonzalez de Dios
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

Jong Yoon Shin
SKT

Email: jongyoon.shin@sk.com

Sergio Belotti
Nokia

Email: sergio.belotti@nokia.com

Network Working Group   I. Bryskin
Internet-Draft   Huawei Technologies
Intended status: Informational   X. Liu
Expires: March 25, 2019   Volta Networks
                                                                  Y. Lee
                                                             J. Guichard
                                                     Huawei Technologies
                                                             L. Contreras
                                                              Telefonica
                                                           D. Ceccarelli
                                                                Ericsson
                                                              J. Tantsura
                                                          Nuage Networks
                                                      September 21, 2018

SF Aware TE Topology YANG Model
draft-ietf-teas-sf-aware-topo-model-02

Abstract

This document describes a YANG data model for TE network topologies that are network service and function aware.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Normally network connectivity services are discussed as a means to inter-connect various abstract or physical network topological elements, such as ports, link termination points and nodes [I-D.ietf-teas-yang-te-topo] [I-D.ietf-teas-yang-te]. However, the connectivity services, strictly speaking, interconnect not the network topology elements per-se, rather, located on/associated with the various network and service functions [RFC7498] [RFC7665]. In many scenarios it is beneficial to decouple the service/network functions from the network topology elements hosting them, describe them in some unambiguous and identifiable way (so that it would be possible, for example, to auto-discover on the network topology service/network functions with identical or similar functionality and characteristics) and engineer the connectivity between the service/network functions, rather than between their current topological locations.

Today a network offers to its clients far more services than just connectivity across the network. Large variety of physical, logical and/or virtual service functions, network functions and transport functions (collectively named in this document as SFs) could be allocated for and assigned to a client. As described in the appendix of this document, there are some important use cases, in which the network needs to represent to the client SFs at the client's disposal as topological elements in relation to other elements of a topology (i.e. nodes, links, link and tunnel termination points) used by the network to describe itself to the client. Not only would such information allow for the client to auto-discover the network's SFs available for the services provisioned for the client, it would also allow for the client selecting the SFs, duel-optimizing the selection on the SF location on the network and connectivity means (e.g. TE tunnels) to inter-connect the SFs. Consequently thus would give to both the network and the client powerful means for the service function chain (SFC [RFC7498] [RFC7665]) negotiation to achieve most efficient and cost effective (from the network point of view) and most optimal yet satisfying all necessary constraints of SFCs (from the client's point of view).

This document defines a YANG data model that allows service functions to be represented along with TE topology elements.

1.1. Terminology

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 BCP 14, [RFC2119].

- Network Function (NF): A functional block within a network infrastructure that has well-defined external interfaces and well-defined functional behaviour [ETSI-NFV-TERM]. Such functions include message router, CDN, session border controller, WAN cceleration, DPI, firewall, NAT, QoE monitor, PE router, BRAS, and radio/fixed access network nodes.

- Network Service: Composition of Network Functions and defined by its functional and behavioural specification. The Network Service contributes to the behaviour of the higher layer service, which is characterized by at least performance, dependability, and security specifications. The end-to-end network service behaviour is the result of the combination of the individual network function behaviours as well as the behaviours of the network infrastructure composition mechanism [ETSI-NFV-TERM].

- Service Function (SF): A function that is responsible for specific treatment of received packets. A service function can act at various layers of a protocol stack (e.g., at the network layer or other OSI layers). As a logical component, a service function can be realized as a virtual element or be embedded in a physical network element. One or more service functions can be embedded in the same network element. Multiple occurrences of the service function can exist in the same administrative domain. A non-exhaustive list of service functions includes: firewalls, WAN and application acceleration, Deep Packet Inspection (DPI), server load balancers, NAT44 [RFC3022], NAT64 [RFC6146], HTTP header enrichment functions, and TCP optimizers. The generic term "L4-L7 services" is often used to describe many service functions [RFC7498].

- Service Function Chain (SFC): A service function chain defines an ordered or partially ordered set of abstract service functions and ordering constraints that must be applied to packets, frames, and/or flows selected as a result of classification. An example of an abstract service function is a firewall. The implied order may not be a linear progression as the architecture allows for SFCs that copy to more than one branch, and also allows for cases where there is flexibility in the order in which service functions need to be applied. The term "service chain" is often used as shorthand for "service function chain" [RFC7498].

- Connectivity Service: Any service between layer 0 and layer 3 aiming at delivering traffic among two or more end customer edge nodes connected to provider edge nodes. Examples include L3VPN, L2VPN etc.

- Link Termination Point (LTP): A conceptual point of connection of a TE node to one of the TE links, terminated by the TE node. Cardinality between an LTP and the associated TE link is 1:0..1 [I-D.ietf-teas-yang-te-topo].

- Tunnel Termination Point (TTP): An element of TE topology representing one or several of potential transport service termination points (i.e. service client adaptation points such as WDM/OCh transponder). TTP is associated with (hosted by) exactly one TE node. TTP is assigned with the TE node scope unique ID. Depending on the TE node's internal constraints, a given TTP hosted by the TE node could be accessed via one, several or all TE links terminated by the TE node [I-D.ietf-teas-yang-te-topo].

The following terms are defined in [RFC7950] and are not redefined here:

- augment

- data model

- data node

1.2. Tree Diagrams

A simplified graphical representation of the data model is presented in this document, by using the tree format defined in [I-D.ietf-netmod-yang-tree-diagrams].

1.3. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
nw	ietf-network	[I-D.ietf-i2rs-yang-network-topo]
nt	ietf-network-topology	[I-D.ietf-i2rs-yang-network-topo]
tet	ietf-te-topology	[I-D.ietf-teas-yang-te-topo]

Table 1: Prefixes and Corresponding YANG Modules

2. Modeling Considerations

The model introduced in this document is an augmentation of the TE Topology model defined in [I-D.ietf-teas-yang-te-topo]. SFs are modeled as child elements of a TE node similarly to how Link Termination Points (LTPs) and Tunnel Termination Points (TTPs) are modeled in the TE Topology model. The SFs are defined as opaque objects identified via topology unique service-function-id's. Each SF has one or more Connection Points (CPs) identified via SF-unique sf-connection-point-id's, over which the SF could be connected to other SFs resided on the same TE node, as well as to other elements of the TE node, in particular, to the node's LTPs and/or TTPs. An interested client may use service-function-id's to look up the SFs in TOSCA or YANG data store(s) defined by [ETSI-NFV-MAN] to retrieve the details of the SFs, for example, to understand the SF's mutual substitutability.

The TE Topology model introduces a concept of Connectivity Matrix (CM), and uses the CM to describe which and at what costs a TE node's LTPs could be inter-connected internally across the TE node. The model defined in this document heavily uses the same concept to describe the SF connectivity via introducing 3 additional CMs:

1. SF2SF CM. This CM describes which pairs of SFs could be locally inter-connected, and, if yes, in which direction, via which CPs and at what costs. In other words, the SF2SF CM describes how SFs residing on the same TE node could be inter-connected into local from the TE node's perspective SFCs;

2. SF2LTP CM. This CM describes how, in which direction and at what costs the TE node's SFs could be connected to the TE node's LTPs and hence to SFs residing on neighboring TE nodes that are connected to LTPs at the remote ends of corresponding TE links;

3. SF2TTP CM. This CM describes how, in which direction and at what costs the TE node's SFs could be connected to the TE node's TTPs and hence to SFs residing on other TE nodes on the topology that could be inter-connected with the TE node in question via TE tunnels terminated by the corresponding TTPs.

In addition to SF2SF CM, the local SF chaining could be described with the help of ETSI models Virtual Links (VLs) [ETSI-NFV-MAN]. This option is especially useful when the costs of the local chaining are negligible as compared to ones of the end-to-end SFCs said local SFCs are part of.

Section 3 and 4 provide the YANG model structure and the YANG module for SF-aware Topology. Section 5 and 6 provide the YANG model structure and the YANG module for Data Center Compute Node resource abstraction. This provides an example of SF2LTP CM where DC compute nodes are connected to LTPs at the remote ends of the corresponding TE links. This use-case is described in Section 10 of Appendix C.

3. Model Structure

```
   module: ietf-te-topology-sf
     augment /nw:networks/nw:network/nw:network-types/tet:te-topology:
       +--rw sf!
     augment /nw:networks/nw:network/nw:node/tet:te
   /tet:te-node-attributes:
       +--rw service-function
          +--rw connectivity-matrices
          |  +--rw connectivity-matrix* [id]
          |     +--rw id                  uint32
          |     +--rw from
          |     |  +--rw service-function-id?      string
          |     |  +--rw sf-connection-point-id?   string
          |     +--rw to
          |     |  +--rw service-function-id?      string
          |     |  +--rw sf-connection-point-id?   string
          |     +--rw enabled?            boolean
          |     +--rw direction?          connectivity-direction
          |     +--rw virtual-link-id?   string
          +--rw link-terminations
             +--rw link-termination* [id]
                +--rw id            uint32
                +--rw from
                |  +--rw tp-ref?   -> ../../../../../../..
   /nt:termination-point/tp-id
                +--rw to
                |  +--rw service-function-id?      string
                |  +--rw sf-connection-point-id?   string
                +--rw enabled?     boolean
                +--rw direction?   connectivity-direction
     augment /nw:networks/nw:network/nw:node/tet:te
   /tet:information-source-entry:
       +--ro service-function
          +--ro connectivity-matrices
          |  +--ro connectivity-matrix* [id]
          |     +--ro id                  uint32
          |     +--ro from
          |     |  +--ro service-function-id?      string
          |     |  +--ro sf-connection-point-id?   string
          |     +--ro to
          |     |  +--ro service-function-id?      string
          |     |  +--ro sf-connection-point-id?   string
          |     +--ro enabled?           boolean
          |     +--ro direction?         connectivity-direction
          |     +--ro virtual-link-id?   string
          +--ro link-terminations
             +--ro link-termination* [id]
                +--ro id           uint32
                +--ro from
                +--ro to
                |  +--ro service-function-id?      string
                |  +--ro sf-connection-point-id?   string
                +--ro enabled?     boolean
                +--ro direction?   connectivity-direction
     augment /nw:networks/nw:network/nw:node/tet:te
   /tet:tunnel-termination-point:
       +--rw service-function
          +--rw tunnel-terminations
             +--rw tunnel-termination* [id]
                +--rw id                       uint32
                +--rw service-function-id?      string
                +--rw sf-connection-point-id?   string
                +--rw enabled?                  boolean
                +--rw direction?                connectivity-direction
```

4. YANG Modules

```
   <CODE BEGINS> file "ietf-te-topology-sf@2018-02-27.yang"
   module ietf-te-topology-sf {
     yang-version 1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology-sf";

     prefix "tet-sf";

     import ietf-network {
       prefix "nw";
     }

     import ietf-network-topology {
       prefix "nt";
     }

     import ietf-te-topology {
       prefix "tet";
     }

     organization
       "Traffic Engineering Architecture and Signaling (TEAS)
```

```
        Working Group";

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

        Editors:  Igor Bryskin
                  <mailto:Igor.Bryskin@huawei.com>

                  Xufeng Liu
                  <mailto:Xufeng_Liu@jabil.com>";

     description
       "Network service and function aware aware TE topology model.

        Copyright (c) 2018 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).";

     revision 2018-02-27 {
       description "Initial revision";
       reference "TBD";
     }

     /*
      * Typedefs
      */
     typedef connectivity-direction {
       type enumeration {
         enum "to" {
           description
           "The direction is uni-directional, towards the 'to'
            entity direction.";
         }
         enum "from" {
           description
           "The direction is uni-directional, from the 'to'
            entity direction.";
         }
         enum "bidir" {
           description
           "The direction is bi-directional.";
         }
       }
       description
         "A type used to indicates whether a connectivity is
          uni-directional, or bi-directional. If the relation is
          uni-directional, the value of this type indicates the
          direction.";
     } // connectivity-direction

     /*
      * Groupings
      */
     grouping service-function-node-augmentation {
       description
         "Augmenting a TE node to be network service and function
          aware.";
       container service-function {
         description
           "Containing attributes related to network services and
            network functions";
         container connectivity-matrices {
           description
             "Connectivity relations between network services/functions
              on a TE node, which can be either abstract or physical.";
           reference
             "ETSI GS NFV-MAN 01: Network Functions Virtualisation
              (NFV); Management and Orchestration.
              RFC7665: Service Function Chaining (SFC) Architecture.";
           list connectivity-matrix {
             key "id";
             description
               "Represents the connectivity relations between network
                services/functions on a TE node.";
             leaf id {
               type uint32;
               description "Identifies the connectivity-matrix entry.";
             }

             container from {
               description
                 "Reference to the source network service or
                  network function.";
               leaf service-function-id {
                 type string;
                 description
                   "Reference to a network service or a network
                    function.";
               }
               leaf sf-connection-point-id {
                 type string;
                 description
                   "Reference to a connection point on a network
                    service or a network function.";
               }
             } // from
             container to {
               description
                 "Reference to the destination network service or
                  network function.";
               leaf service-function-id {
                 type string;
                 description
                   "Reference to a network service or a network
                    function.";
                 }
               leaf sf-connection-point-id {
                 type string;
                 description
                   "Reference to a connection point on a network
                    service or a network function.";
               }
             } // to
             leaf enabled {
               type boolean;
               description
                 "'true' if this connectivity entry is enabled.";
             }
             leaf direction {
               type connectivity-direction;
               description
                 "Indicates whether this connectivity is
                  uni-directional, or bi-directional. If the
                  relation is uni-directional, the value of
                  this leaf indicates the direction.";
             }
             leaf virtual-link-id {
               type string;
               description
                 "Reference to a virtual link that models this
                  conectivity relation in the network function
                  model.";
             }
           } // connectivity-matrix
         } // connectivity-matrices

         container link-terminations {
           description
             "Connectivity relations between network services/functions
              and link termination points on a TE node, which can be
              either abstract or physical.";
           reference
             "ETSI GS NFV-MAN 01: Network Functions Virtualisation
              (NFV); Management and Orchestration.
              RFC7665: Service Function Chaining (SFC) Architecture.";
           list link-termination {
             key "id";
             description
               "Each entry of the list represents the connectivity
                relation between a network service/function and
                a link termination point on a TE node.";
             leaf id {
               type uint32;
               description "Identifies the termination entry.";
             }

             container from {
               description
                 "Reference to the link termination point.";
             } // from
             container to {
               description
                 "Reference to the network service or network
                  function.";
               leaf service-function-id {
                 type string;
                 description
                   "Reference to a network service or a network
                    function.";
                 }
               leaf sf-connection-point-id {
                 type string;
                 description
                   "Reference to a connection point on a network
                    service or a network function.";
               }
             } // to
             leaf enabled {
               type boolean;
               description
                 "'true' if this connectivity entry is enabled.";
             }
             leaf direction {
               type connectivity-direction;
               description
```

```
            "Indicates whether this connectivity is
             uni-directional, or bi-directional. If the
             relation is uni-directional, the value of
             this leaf indicates the direction.";
        }
      } // link-termination
    }
  }
} // service-function-node-augmentation

grouping service-function-ttp-augmentation {
  description
    "Augmenting a tunnel termination point to be network service
     aware.";
  container service-function {
    description
      "Containing attributes related to network services and
       network functions";
    container tunnel-terminations {
      description
        "Connectivity relations between network services/functions
         and tunnel termination points on a TE node, which can be
         either abstract or physical.";
      reference
        "ETSI GS NFV-MAN 01: Network Functions Virtualisation
         (NFV); Management and Orchestration.
         RFC7665: Service Function Chaining (SFC) Architecture.";
      list tunnel-termination {
        key "id";
        description
          "Each entry of the list represents the connectivity
           relation between a network service/function and
           a tunnel termination point on a TE node.";
        leaf id {
          type uint32;
          description "Identifies the termination entry.";
        }

        leaf service-function-id {
          type string;
          description
            "Reference to a network service or a network
             function.";
        }
        leaf sf-connection-point-id {
          type string;
          description
            "Reference to a connection point on a network
```

```
                  service or a network function.";
             }
             leaf enabled {
               type boolean;
               description
                 "'true' if this connectivity entry is enabled.";
             }
             leaf direction {
               type connectivity-direction;
               description
                 "Indicates whether this connectivity is
                  uni-directional, or bi-directional. If the
                  relation is uni-directional, the value of
                  this leaf indicates the direction.";
             }
           } // link-termination
         }
       }
     } // service-function-ttp-augmentation

     grouping sf-topology-type {
       description
         "Identifies the SF aware TE topology type.";
       container sf {
         presence "Indidates that the TE topology is SF aware.";
         description
           "Its presence identifies that the TE topology is SF aware.";
       }
     } // sf-topology-type

     /*
      * Augmentations
      */
     /* Augmentations to network-types/te-topology */
     augment "/nw:networks/nw:network/nw:network-types/"
       + "tet:te-topology" {
       description
         "Defines the SF aware TE topology type.";
       uses sf-topology-type;
     }

     /* Augmentations to te-node-attributes */
     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes" {
       description
         "Parameters for SF aware TE topology.";
       uses service-function-node-augmentation;
     }
     augment "/nw:networks/nw:network/nw:node/tet:te/"
           + "tet:information-source-entry" {
       description
         "Parameters for SF aware TE topology.";
       uses service-function-node-augmentation;
     }

     /* Augmentations to tunnel-termination-point */
     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point" {
       description
         "Parameters for SF aware TE topology.";
       uses service-function-ttp-augmentation;
     }

     /* Augmentations to connectivity-matrix */
     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet-sf:service-function/"
       + "tet-sf:link-terminations/tet-sf:link-termination/"
       + "tet-sf:from" {
       description
         "Add reference to the link termination point.
          This portion cannot be shared with the state module.";
       leaf tp-ref {
         type leafref {
           path "../../../../../../../nt:termination-point/"
             + "nt:tp-id";
         }
         description
           "Reference to the link termination point.";
       }
     }
   }
   <CODE ENDS>
```

5. Model Structure

```
   module: ietf-cso-dc
       +--rw cso
          +--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*   -> /cso/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?   -> /cso/dc/hypervisor/id
          |  |  +--rw port*         -> /cso/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?     -> /cso/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 cso-ref?      -> /cso/cso-id
          |  +--rw ap*                    -> /actn-vn:actn/ap
   /access-point-list/access-point-id
          |  +--rw cso-ref?               -> /cso/cso-id
          |  +--rw id                     string
          |  +--rw name?                  string
          +--rw cso-id?   string
```

6. YANG Modules

```
   <CODE BEGINS> file "ietf-cso-dc@2017-01-16.yang"
   module ietf-cso-dc
   {
     namespace "urn:ietf:params:xml:ns:yang:ietf-cso-dc";
     prefix "dc";

     import ietf-inet-types {
       prefix "inet";
     }
     import ietf-actn-vn {
       prefix "actn-vn";
     }

     revision 2017-01-16 {
       description
         "Initial revision. This YANG file defines
          the reusable base types for CSO DC description.";
       reference
         "Derived from earlier versions of base YANG files";
     }

     // Abstract models
     grouping resource-element {
       leaf id { type string; }
       leaf name { type string; }
     }

     grouping resource-instance {
       leaf project{ type string; }
       leaf status {
         type enumeration {
           enum active;
           enum inactive;
           enum pending;
         }
       }
       uses resource-element;
     }

     // Compute models
     grouping format {
       leaf container {
         type enumeration {
           enum ami;
           enum ari;
           enum aki;
           enum bare;
           enum ovf;
         }
         default bare;
       }
       leaf disk {
         type enumeration {
           enum ami;
           enum ari;
           enum aki;
           enum vhd;
           enum vmdk;
           enum raw;
           enum qcow2;
           enum vdi;
           enum iso;
         }
         default qcow2;
       }
     }

     grouping image {
       leaf checksum { type string; mandatory true; }
       leaf size { type uint32; units 'Bytes'; mandatory true; }

       container format {
         uses format;
       }

       uses resource-element;
     }

     grouping flavor {
       leaf disk  { type uint32; units 'GB'; default 0; }
       leaf ram   { type uint32; units 'MB'; default 0; }
       leaf vcpus { type uint16; default 0; }
       uses resource-element;
     }

     grouping ram {
       leaf total { type uint32; units 'MB'; }
       leaf used { type uint32; units 'MB'; }
       leaf free { type uint32; units 'MB'; }
     }

     grouping disk {
       leaf total { type uint32; units 'GB'; }
       leaf used { type uint32; units 'GB'; }
       leaf free { type uint32; units 'GB'; }
     }

     grouping vcpu {
       leaf total { type uint16; }
       leaf used { type uint16; }
       leaf free { type uint16; }
     }

     grouping hypervisor {
       container ram {
         uses ram;
       }

       container disk {
         uses disk;
       }

       container vcpu {
         uses vcpu;
       }

       leaf-list instance {
         type leafref { path '/cso/dc/instance/id'; } }
       uses resource-element;
     }

     grouping instance {
       container flavor { uses flavor; }
       container image { uses image; }
       leaf hypervisor {
         type leafref { path '/cso/dc/hypervisor/id'; } }
       leaf-list port { type leafref {
           path '/cso/dc/network/subnetwork/port/id'; } }
       uses resource-instance;
     }

     grouping dc-monitoring-param {
       leaf name {
         description "dc-monitoring-param identifier"; type string; }
       leaf value-string {
         description
           "Current value for a string parameter";
         type string;
       }
     }

     grouping dc {

       list hypervisor {
         key id;
         uses hypervisor;
       }

       list instance {
         key id;
         uses instance;
       }
       list image {
         key id;
         uses image;
       }

       list flavor {
         key id;
         uses flavor;
       }

       list dc-monitoring-param {
           key "name";
           uses dc-monitoring-param;
       }

       list network {
           key id;
           uses network;
       }

       leaf-list ap { type leafref {
           path
             '/actn-vn:actn/actn-vn:ap/actn-vn:access-point-list/'
             + 'actn-vn:access-point-id';
         }
       }
       leaf cso-ref { type leafref { path "/cso/cso-id"; } }
       uses resource-element;
     }



     container cso {
       list dc {
           key id;
           uses dc;
       }

       leaf cso-id { type string; }
     }

     // Network models
     grouping ip-address {
       leaf ip-address { type inet:ip-address; }
     }

     grouping dhcp-agent {
       leaf enabled { type boolean; }
       list pools {
         key ip-address;
         uses ip-address;
       }
       uses resource-instance;
     }

     grouping network {
       list subnetwork {
         key id;
         uses subnetwork;
       }
       list dhcp-agent {
         key id;
         uses dhcp-agent;
       }
       uses resource-instance;
       leaf cso-ref { type leafref { path "/cso/cso-id"; } }
     }

     grouping subnetwork {
       list port {
         key id;
         uses port;
       }
       uses resource-instance;
     }

     grouping port {
       leaf ip-address { type inet:ip-address; }
       leaf instance { type leafref { path '/cso/dc/instance/id'; } }
       uses resource-instance;
     }



   }
   <CODE ENDS>
```

7. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

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

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

   --------------------------------------------------------------------
   URI: urn:ietf:params:xml:ns:yang:ietf-te-topology-sf-state
   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-topology-sf
   namespace:    urn:ietf:params:xml:ns:yang:ietf-te-topology-packet
   prefix:       tet-sf
   reference:    RFC XXXX
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   name:         ietf-te-topology-sf-state
   namespace:  urn:ietf:params:xml:ns:yang:ietf-te-topology-packet-state
   prefix:       tet-sf-s
   reference:    RFC XXXX
   --------------------------------------------------------------------
```

8. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The data-model by itself does not create any security implications. The security considerations for the NETCONF protocol are applicable. The NETCONF protocol used for sending the data supports authentication and encryption.

9. References

9.1. Normative References

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

[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>.

[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>.

[ETSI-NFV-MAN] ETSI, "Network Functions Virtualisation (NFV); Management and Orchestration", ETSI GS NFV-MAN 001 V1.1.1, December 2014.

[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-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-18 (work in progress), June 2018.

[I-D.ietf-netmod-revised-datastores] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-10 (work in progress), January 2018.

9.2. Informative References

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[I-D.ietf-netmod-yang-tree-diagrams]
Bjorklund, M. and L. Berger, "YANG Tree Diagrams", draft-ietf-netmod-yang-tree-diagrams-06 (work in progress), February 2018.

9.3. Normative References

[ETSI-NFV-TERM]
ETSI, "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV", ETSI GS NFV 003 V1.2.1, December 2014.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-16 (work in progress), July 2018.

[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>.

[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>.

[I-D.ietf-sfc-hierarchical]
Dolson, D., Homma, S., Lopez, D., and M. Boucadair, "Hierarchical Service Function Chaining (hSFC)", draft-ietf-sfc-hierarchical-11 (work in progress), June 2018.

[I-D.defoy-netslices-3gpp-network-slicing]
Foy, X. and A. Rahman, "Network Slicing - 3GPP Use Case", draft-defoy-netslices-3gpp-network-slicing-02 (work in progress), October 2017.

[_3GPP.28.801]
3GPP, "Study on management and orchestration of network slicing for next generation network", 3GPP TR 28.801 V2.0.0, September 2017, <http://www.3gpp.org/ftp/Specs/html-info/28801.htm>.

[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. Companion YANG Model for Non-NMDA Compliant Implementations

The YANG module ietf-te-topology-sf defined in this document is designed to be used in conjunction with implementations that support the Network Management Datastore Architecture (NMDA) defined in [I-D.ietf-netmod-revised-datastores]. In order to allow implementations to use the model even in cases when NMDA is not supported, the following companion module, ietf-te-topology-sf-state, is defined as state model, which mirrors the module ietf-te-topology-sf defined earlier in this document. However, all data nodes in the companion module are non-configurable, to represent the applied configuration or the derived operational states.

The companion module, ietf-te-topology-sf-state, is redundant and SHOULD NOT be supported by implementations that support NMDA.

As the structure of the companion module mirrors that of the coorespinding NMDA model, the YANG tree of the companion module is not depicted separately.

A.1. SF Aware TE Topology State Module

```
<CODE BEGINS> file "ietf-te-topology-sf-state@2018-02-27.yang"
module ietf-te-topology-sf-state {
  yang-version 1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology-sf-state";

  prefix "tet-sf-s";

  import ietf-te-topology-sf {
    prefix "tet-sf";
  }

  import ietf-network-state {
    prefix "nw-s";
  }

  import ietf-network-topology-state {
    prefix "nt-s";
  }

  import ietf-te-topology-state {
    prefix "tet-s";
  }

  organization
    "Traffic Engineering Architecture and Signaling (TEAS)
```

```
        Working Group";

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

        Editors:  Igor Bryskin
                  <mailto:Igor.Bryskin@huawei.com>

                  Xufeng Liu
                  <mailto:Xufeng_Liu@jabil.com>";

     description
       "Network service and function aware aware TE topology operational
        state model for non-NMDA compliant implementations.

        Copyright (c) 2018 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).";

     revision 2018-02-27 {
       description "Initial revision";
       reference "TBD";
     }

     /*
      * Augmentations
      */
     /* Augmentations to network-types/te-topology */
     augment "/nw-s:networks/nw-s:network/nw-s:network-types/"
       + "tet-s:te-topology" {
       description
         "Defines the SF aware TE topology type.";
       uses tet-sf:sf-topology-type;
     }

     /* Augmentations to connectivity-matrix */
     augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
       + "tet-s:te-node-attributes" {
       description
         "Parameters for SF aware TE topology.";
       uses tet-sf:service-function-node-augmentation;
     }

     augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
           + "tet-s:information-source-entry" {
       description
         "Parameters for SF aware TE topology.";
       uses tet-sf:service-function-node-augmentation;
     }

     /* Augmentations to tunnel-termination-point */
     augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
       + "tet-s:tunnel-termination-point" {
       description
         "Parameters for SF aware TE topology.";
       uses tet-sf:service-function-ttp-augmentation;
     }

     /* Augmentations to connectivity-matrix */
     augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
       + "tet-s:te-node-attributes/tet-sf-s:service-function/"
       + "tet-sf-s:link-terminations/tet-sf-s:link-termination/"
       + "tet-sf-s:from" {
       description
         "Add reference to the link termination point.
          This portion cannot be shared with the state module.";
       leaf tp-ref {
         type leafref {
           path "../../../../../../../nt-s:termination-point/"
             + "nt-s:tp-id";
         }
         description
           "Reference to the link termination point.";
       }
     }
   }
   <CODE ENDS>
```

Appendix B. Data Examples

B.1. A Topology with Multiple Connected Network Functions

```
                            Node-1
         +----o--o--------------------------o-------+
         |    |  |                          |       |
         |    \__/                          \__     |
         |    *\/ TTP-1   * * * * * * * * * *\/*    |
   LTP-4 |* * *         *                TTP-2  *   | LTP-1
         o------------*-----------------------------o
         |          *                             * |
   LTP-3 |* * * * *                                *| LTP-2
         o---                                  -----o
         |   \                                /     |
         |    \                              /      |
         |     \ CP01                   CP02/       |
         | +----o--------------------------o------+ |
         | | VL1|                       VL4|      | |
         | |    |CP11                      |CP33  | |
         | |  +-o--+        +----+       +-o--+   | |
         | |  |VNF1|        |VNF2|       |VNF3|   | |
         | |  +-o-o+  VL2   +--o-+  VL2  +-o-o+   | |
         | |CP12| |\----------/ \---------/| |CP32| |
         | |    | |CP13      CP21      CP31| |    | |
         | |    | |      VL2               | |    | |
         | |    | +------------------------+ |    | |
         | |    +----------------------------+    | |
         | |             VL3                      | |
         | |                  Network Service 1   | |
         | +--------------------------------------+ |
         +------------------------------------------+
```

The configuration instance data for Node-1 in the above figure could be as follows:

```
   {
     "networks": {
       "network": [
         {
           "network-types": {
             "te-topology": {
               "sf": {}
             }
           },
           "network-id": "network-sf-aware",
           "provider-id": 201,
           "client-id": 300,
           "te-topology-id": "te-topology:network-sf-aware",
           "node": [
             {
               "node-id": "Node-1",
               "te-node-id": "2.0.1.1",
               "te": {
                 "te-node-attributes": {
                   "domain-id": 1,
                   "is-abstract": [null],
                   "connectivity-matrices": {
                   },
                   "service-function": {
                     "connectivity-matrices": {
                       "connectivity-matrix": [
                         {
                           "id": 10,
                           "from": {
                             "service-function-id": "Network Service 1",
                             "sf-connection-point-id": "CP01"
                           },
                           "to": {
                             "service-function-id": "VNF1",
                             "sf-connection-point-id": "CP11"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL1"
                         },
                         {
                           "id": 13,
                           "from": {
                             "service-function-id": "VNF1",
                             "sf-connection-point-id": "CP12"
                           },
                           "to": {
                             "service-function-id": "VNF3",
                             "sf-connection-point-id": "CP32"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL3"
                         },
                         {
                           "id": 12,
                           "from": {
                             "service-function-id": "VNF1",
                             "sf-connection-point-id": "CP13"
                           },
                           "to": {
                             "service-function-id": "VNF2",
                             "sf-connection-point-id": "CP21"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL2"
                         },
                         {
                           "id": 23,
                           "from": {
                             "service-function-id": "VNF2",
                             "sf-connection-point-id": "CP21"
                           },
                           "to": {
                             "service-function-id": "VNF3"
                             "sf-connection-point-id": "CP31"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL2"
                         },
                         {
                           "id": 30,
                           "from": {
                             "service-function-id": "Network Service 1",
                             "sf-connection-point-id": "CP02"
                           },
                           "to": {
                             "service-function-id": "VNF3",
                             "sf-connection-point-id": "CP33"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL4"
                         }
                       ]
                     },
                     "link-terminations": {
                       "link-termination": [
                         {
                           "id": 2,
                           "from": {
                             "tp-ref": "LTP-2"
                           },
                           "to": {
                             "service-function-id": "Network Service 1",
                             "sf-connection-point-id": "CP02"
                           }
                           "direction": "bidir"
                         },
                         {
                           "id": 3,
                           "from": {
                             "tp-ref": "LTP-3"
                           },
                           "to": {
                             "service-function-id": "Network Service 1",
                             "sf-connection-point-id": "CP01"
                           }
                           "direction": "bidir"
                         }
                       ]
                     }
                   }
                 }
                 "tunnel-termination-point": [
                   {
                     "tunnel-tp-id": 10001,
                     "name": "TTP-1",
                     "service-function-terminations": {
                     }
                   },
                   {
                     "tunnel-tp-id": 10002,
                     "name": "TTP-2",
                     "service-function-terminations": {
                     }
                   }
                 ]
               },
               "termination-point": [
                 {
                   "tp-id": "LTP-1",
                   "te-tp-id": 10001
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-l2sc",
                         "encoding": "lsp-encoding-ethernet"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "LTP-2",
                   "te-tp-id": 10002
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-l2sc",
                         "encoding": "lsp-encoding-ethernet"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "LTP-3",
                   "te-tp-id": 10003
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-l2sc",
                         "encoding": "lsp-encoding-ethernet"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "LTP-4",
                   "te-tp-id": 10004
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-l2sc",
                         "encoding": "lsp-encoding-ethernet"
                       }
                     ]
                   }
                 }
               ]
             }
           ]
         }
       ]
     }
   }
```

B.2. A Topology with an Encapsulated Network Service

In this example, a network service consists of several inter-connected network functions (NFs), and is represented by this model as an encapsulated opaque object without the details between its internals.

```
                            Node-1
         +----o--o--------------------------o-------+
         |    |  |                          |       |
         |    \__/                          \__     |
         |    *\/ TTP-1   * * * * * * * * * *\/*    |
   LTP-4 |* * *         *                TTP-2  *   | LTP-1
         o------------*-----------------------------o
         |          *                             * |
   LTP-3 |* * * * *                                *| LTP-2
         o---                                  -----o
         |   \                                /     |
         |    \                              /      |
         |     \ CP01                   CP02/       |
         | +----o--------------------------o------+ |
         | |                                      | |
         | |                  Network Service 1   | |
         | +--------------------------------------+ |
         +------------------------------------------+
```

The configuration instance data for Node-1 in the above figure could be as follows:

```
{
  "networks": {
    "network": [
      {
        "network-types": {
          "te-topology": {
            "sf": {}
          }
        },
        "network-id": "network-sf-aware",
        "provider-id": 201,
        "client-id": 300,
        "te-topology-id": "te-topology:network-sf-aware",
        "node": [
          {
            "node-id": "Node-1",
            "te-node-id": "2.0.1.1",
            "te": {
              "te-node-attributes": {
                "domain-id": 1,
                "is-abstract": [null],
                "connectivity-matrices": {
                },
                "service-function": {
                  "connectivity-matrices": {
                  },
                  "link-terminations": {
                    "link-termination": [
                      {
                        "id": 2,
                        "from": {
                          "tp-ref": "LTP-2"
                        },
                        "to": {
                          "service-function-id": "Network Service 1",
                          "sf-connection-point-id": "CP02"
                        }
                        "direction": "bidir"
                      },
                      {
                        "id": 3,
                        "from": {
                          "tp-ref": "LTP-3"
                        },
                        "to": {
                          "service-function-id": "Network Service 1",
                          "sf-connection-point-id": "CP01"
                        }
                        "direction": "bidir"
                      }
                    ]
                  }
                }
              }
              "tunnel-termination-point": [
                {
                  "tunnel-tp-id": 10001,
                  "name": "TTP-1",
                  "service-function-terminations": {
                  }
                },
                {
                  "tunnel-tp-id": 10002,
                  "name": "TTP-2",
                  "service-function-terminations": {
                  }
                }
              ]
            },
            "termination-point": [
              {
                "tp-id": "LTP-1",
                "te-tp-id": 10001
                "te": {
                  "interface-switching-capability": [
                    {
                      "switching-capability": "switching-l2sc",
                      "encoding": "lsp-encoding-ethernet"
                    }
                  ]
                }
              },
              {
                "tp-id": "LTP-2",
                "te-tp-id": 10002
                "te": {
                  "interface-switching-capability": [
                    {
                      "switching-capability": "switching-l2sc",
                      "encoding": "lsp-encoding-ethernet"
                    }
                  ]
                }
              },
              {
                "tp-id": "LTP-3",
                "te-tp-id": 10003
                "te": {
                  "interface-switching-capability": [
                    {
                      "switching-capability": "switching-l2sc",
                      "encoding": "lsp-encoding-ethernet"
                    }
                  ]
                }
              },
              {
                "tp-id": "LTP-4",
                "te-tp-id": 10004
                "te": {
                  "interface-switching-capability": [
                    {
                      "switching-capability": "switching-l2sc",
                      "encoding": "lsp-encoding-ethernet"
                    }
                  ]
                }
              }
            ]
          }
        ]
      }
```

```
       ]
     }
   }
```

Appendix C. Use Cases for SF Aware Topology Models

C.1. Exporting SF/NF Information to Network Clients and Other Network SDN Controllers

In the context of Service Function Chain (SFC) orchestration one existing problem is that there is no way to formally describe a Service or Network Function in a standard way (recognizable/ understood by a third party) as a resource of a network topology node.

One implication of this is that there is no way for the orchestrator to give a network client even a ball-park idea as to which network's SFs/NFs are available for the client's use/control and where they are located in the network even in terms of abstract topologies/virtual networks configured and managed specifically for the client. Consequently, the client has no say on how the SFCs provided for the client by the network should be set up and managed (which SFs are to be used and how they should be chained together, optimized, manipulated, protected, etc.).

Likewise, there is no way for the orchestrator to export SF/NF information to other network controllers. The SFC orchestrator may serve, for example, a higher level controller (such as Network Slicing Orchestrator), with the latter wanting at least some level of control as to which SFs/NFs it wants on its SFCs and how the Service Function Paths (SFPs) are to be routed and provisioned, especially, if it uses services of more than one SFC orchestrator.

The issue of exporting of SF/NF information could be addressed by defining a model, in which formally described/recognizable SF/NF instances are presented as topological elements, for example, hosted by TE, L3 or L2 topology nodes (see Figure 1). The model could describe whether, how and at what costs the SFs/NFs hosted by a given node could be chained together, how these intra-node SFCs could be connected to the node's Service Function Forwarders (SFFs, entities dealing with SFC NSHs and metadata), and how the SFFs could be connected to the node's Tunnel and Link Termination Points (TTPs and LTPs) to chain the intra-node SFCs across the network topology.

Figure 1: SF/NF aware TE topology

C.2. Flat End-to-end SFCs Managed on Multi-domain Networks

SFCs may span multiple administrative domains, each of which controlled by a separate SFC controller. The usual solution for such a scenario is the Hierarchical SFCs (H-SFCs), in which the higher level orchestrator controls only SFs located on domain border nodes. Said higher level SFs are chained together into higher level SFCs via lower level (intra-domain) SFCs provisioned and controlled independently by respective domain controllers. The decision as to which higher level SFCs are connected to which lower level SFCs is driven by packet re-classification every time the packet enters a given domain. Said packet re-classification is a very time-consuming operation. Furthermore, the independent nature of higher and lower level SFC control is prone to configuration errors, which may lead to long lasting loops and congestions. It is highly desirable to be able to set up and manage SFCs spanning multiple domains in a flat way as far as the data plane is concerned (i.e. with a single packet classification at the ingress into the multi-domain network but without re-classifications on domain ingress nodes).

One way to achieve this is to have the domain controllers expose SF/NF- aware topologies, and have the higher level orchestrator operate on the network-wide topology, the product of merging of the topologies catered by the domain controllers. This is similar in spirit to setting up, coordinating and managing the transport connectivity (TE tunnels) on a multi-domain multi-vendor transport network.

C.3. Managing SFCs with TE Constraints

Some SFCs require per SFC link/element and end-to-end TE constrains (bandwidth, delay/jitter, fate sharing/diversity. etc.). Said constraints could be ensured via carrying SFPs inside overlays that are traffic engineered with the constrains in mind. A good analogy would be orchestrating delay constrained L3 VPNs. One way to support such L3 VPNs is to carry MPLS LSPs interconnecting per-VPN VRFs inside delay constrained TE tunnels interconnecting the PEs hosting the VRFs.

Figure 2: L3 VPN with delay constraints

Planning, computing and provisioning of TE overlays to constrain arbitrary SFCs, especially those that span multiple administrative domains with each domain controlled by a separate controller, is a very difficult challenge. Currently it is addressed by pre-provisioning on the network of multiple TE tunnels with various TE characteristics, and "nailing down" SFs/NFs to "strategic" locations (e.g. nodes terminating many of such tunnels) in a hope that an adequate set of tunnels could be found to carry the SFP of a given TE-constrained SFC. Such an approach is especially awkward in the case when some or all of the SFs/NFs are VNFs (i.e. could be instantiated at multiple network locations).

SF/NF-aware TE topology model in combination with TE tunnel model will allow for the network orchestrator (or a client controller) to compute, set up and manipulate the TE overlays in the form of TE tunnel chains (see Figure 3).

Said chains could be duel-optimized compromising on optimal SF/NF locations with optimal TE tunnels interconnecting them. The TE tunnel chains (carrying multiple similarly constrained SFPs) could be adequately constrained both at individual TE tunnel level and at the chain end-to-end level.

Figure 3: SFC with TE constraints

C.4. SFC Protection and Load Balancing

Currently the combination of TE topology & tunnel models offers to a network controller various capabilities to recover an individual TE tunnel from network failures occurred on one or more network links or transit nodes on the TE paths taken by the TE tunnel's connection(s). However, there is no simple way to recover a TE tunnel from a failure affecting its source or destination node. SF/NF-aware TE topology model can decouple the association of a given SF/NF with its location on the network topology by presenting multiple, identifiable as mutually substitutable SFs/NFs hosted by different TE topology nodes. So, for example, if it is detected that a given TE tunnel destination node is malfunctioning or has gone out of service, the TE tunnel could be re-routed to terminate on a different node hosting functionally the same SFs/NFs as ones hosted by the failed node (see Figures 6).

This is in line with the ACTN edge migration and function mobility concepts [RFC8453]. It is important to note that the described strategy works much better for the stateless SFs/NFs. This is because getting the alternative stateful SFs/NFs into the same respective states as the current (i.e. active, affected by failure) are is a very difficult challenge.

Figure 4: SFC recovery: SF2 on node NE1 fails

At the SFC level the SF/NF-aware TE topology model can offer SFC dynamic restoration capabilities against failed/malfunctioning SFs/NFs by identifying and provisioning detours to a TE tunnel chain, so that it starts carrying the SFC's SFPs towards healthy SFs/NFs that are functionally the same as the failed ones. Furthermore, multiple parallel TE tunnel chains could be pre-provisioned for the purpose of SFC load balancing and end-to-end protection. In the latter case said parallel TE tunnel chains could be placed to be sufficiently disjoint from each other.

Figure 5: SFC recovery: SFC SF1-SF2-SF6 is recovered after SF2 on node N1 has failed

Figure 6: SFC recovery: SFC SF1-SF2-SF6 is recovered after node N1 has failed

C.5. Network Clock Synchronization

Many current and future network applications (including 5g and IoT applications) require very accurate time services (PTP level, ns resolution). One way to implement the adequate network clock synchronization for such services is via describing network clocks as NFs on an NF-aware TE topology optimized to have best possible delay variation characteristics. Because such a topology will contain delay/delay variation metrics of topology links and node cross-connects, as well as costs in terms of delay/delay variation of connecting clocks to hosting them node link and tunnel termination points, it will be possible to dynamically select and provision bi-directional time-constrained deterministic paths or trees connecting clocks (e.g. grand master and boundary clocks) for the purpose of exchange of clock synchronization information. Note that network clock aware TE topologies separately provided by domain controllers will enable multi-domain network orchestrator to set up and manipulate the clock synchronization paths/trees spanning multiple network domains.

C.6. Client - Provider Network Slicing Interface

3GPP defines network slice as "a set of network functions and the resources for these network functions which are arranged and configured, forming a complete logical network to meet certain network characteristics" [I-D.defoy-netslices-3gpp-network-slicing] [_3GPP.28.801]. Network slice could be also defined as a logical partition of a provider's network that is owned and managed by a tenant. SF/NF-aware TE topology model has a potential to support a very important interface between network slicing clients and providers because, on the one hand, the model can describe holistically and hierarchically the client's requirements and preferences with respect to a network slice functional, topological and traffic engineering aspects, as well as of the degree of resource separation/ sharing between the slices, thus allowing for the client (up to agreed upon extent) to dynamically (re-)configure the slice or (re-)schedule said (re-)configurations in time, while, on the other hand, allowing for the provider to convey to the client the slice's operational state information and telemetry the client has expressed interest in.

C.7. Dynamic Assignment of Regenerators for L0 Services

On large optical networks, some of provided to their clients L0 services could not be provisioned as single OCh trails, rather, as chains of such trails interconnected via regenerators, such as 3R regenerators. Current practice of the provisioning of such services requires configuration of explicit paths (EROs) describing identity and location of regenerators to be used. A solution is highly desirable that could:

- Identify such services based, for example, on optical impairment computations;

- Assign adequate for the services regenerators dynamically out of the regenerators that are grouped together in pools and strategically scattered over the network topology nodes;

- Compute and provision supporting the services chains of optical trails interconnected via so selected regenerators, optimizing the chains to use minimal number of regenerators, their optimal locations, as well as optimality of optical paths interconnecting them;

- Ensure recovery of such chains from any failures that could happen on links, nodes or regenerators along the chain path.

NF-aware TE topology model (in this case L1 NF-aware L0 topology model) is just the model that could provide a network controller (or even a client controller operating on abstract NF-aware topologies provided by the network) to realize described above computations and orchestrate the service provisioning and network failure recovery operations (see Figure 7).

Figure 7: Optical tunnel as TE-constrained SFC of 3R regenerators. Red trail (not regenerated) is not optically reachable, but blue trail (twice regenerated) is

C.8. Dynamic Assignment of OAM Functions for L1 Services

OAM functionality is normally managed by configuring and manipulating TCM/MEP functions on network ports terminating connections or their segments over which OAM operations, such as performance monitoring, are required to be performed. In some layer networks (e.g. Ethernet) said TCMs/MEPs could be configured on any network ports. In others (e.g. OTN/ODUk) the TCMs/MEPs could be configured on some (but not all network ports) due to the fact that the OAM functionality (i.e. recognizing and processing of OAM messages, supporting OAM protocols and FSMs) requires in these layer networks certain support in the data plane, which is not available on all network nodes. This makes TCMs/MEPs good candidates to be modeled as NFs. This also makes TCM/MEP aware topology model a good basis for placing dynamically an ODUk connection to pass through optimal OAM locations without mandating the client to specify said locations explicitly.

Figure 8: Compute/storage resource aware topology

C.9. SFC Abstraction and Scaling

SF/NF-aware topology may contain information on native SFs/NFs (i.e. SFs/NFs as known to the provider itself) and/or abstract SFs/NFs (i.e. logical/macro SFs/NFs representing one or more SFCs each made of native and/or lower level abstract SFs/NFs). As in the case of abstracting topology nodes, abstracting SFs/NFs is hierarchical in nature - the higher level of SF/NF-aware topology, the "larger" abstract SFs/NFs are, i.e. the larger data plane SFCs they represent. This allows for managing large scale networks with great number of SFs/NFs (such as Data Center interconnects) in a hierarchical, highly scalable manner resulting in control of very large number of flat in the data plane SFCs that span multiple domains.

C.10. Dynamic Compute/VM/Storage Resource Assignment

In a distributed data center network, virtual machines for compute resources may need to be dynamically re-allocated due to various reasons such as DCI network failure, compute resource load balancing, etc. In many cases, the DCI connectivity for the source and the destination is not predetermined. There may be a pool of sources and a pool of destination data centers associated with re-allocation of compute/VM/storage resources. There is no good mechanism to date to capture this dynamicity nature of compute/VM/storage resource reallocation. Generic Compute/VM/Storage resources can be described and announced as a SF, where a DC hosting these resources can be modeled as an abstract node. Topology interconnecting these abstract nodes (DCs) in general is of multi-domain nature. Thus, SF-aware topology model can facilitate a joint optimization of TE network resources and Compute/VM/Storage resources and solve Compute/VM/Storage mobility problem within and between DCs (see Figure 8).

C.11. Application-aware Resource Operations and Management

Application stratum is the functional grouping which encompasses application resources and the control and management of these resources. These application resources are used along with network services to provide an application service to clients/end-users. Application resources are non-network resources critical to achieving the application service functionality. Examples of application resources include: caches, mirrors, application specific servers, content, large data sets, and computing power. Application service is a networked application offered to a variety of clients (e.g., server backup, VM migration, video cache, virtual network on-demand, 5G network slicing, etc.). The application servers that host these application resources can be modeled as an abstract node. There may be a variety of server types depending on the resources they host. Figure 9 shows one example application aware topology for video cache server distribution.

Figure 9: Application aware topology

C.12. IANA Considerations

This document has no actions for IANA.

C.13. Security Considerations

This document does not define networking protocols and data, hence is not directly responsible for security risks.

C.14. Acknowledgements

The authors would like to thank Maarten Vissers, Joel Halpern, and Greg Mirsky for their helpful comments and valuable contributions.

Authors' Addresses

Igor Bryskin
Huawei Technologies

EMail: Igor.Bryskin@huawei.com

Xufeng Liu
Volta Networks

EMail: xufeng.liu.ietf@gmail.com

Young Lee
Huawei Technologies

EMail: leeyoung@huawei.com

Jim Guichard
Huawei Technologies

EMail: james.n.guichard@huawei.com

Luis Miguel Contreras Murillo
Telefonica

EMail: luismiguel.contrerasmurillo@telefonica.com

Daniele Ceccarelli
Ericsson

EMail: daniele.ceccarelli@ericsson.com

Jeff Tantsura
Nuage Networks

EMail: jefftant.ietf@gmail.com

```
TEAS Working Group                                  Italo Busi (Ed.)
Internet Draft                                                Huawei
Intended status: Standard Track                 Sergio Belotti (Ed.)
Expires: April 2019                                            Nokia
                                                         Victor Lopez
                                             Oscar Gonzalez de Dios
                                                          Telefonica
                                                       Anurag Sharma
                                                              Google
                                                             Yan Shi
                                                        China Unicom
                                                      Ricard Vilalta
                                                                CTTC
                                                  Karthik Sethuraman
                                                                 NEC

                                                    October 22, 2018
```

Yang model for requesting Path Computation
draft-ietf-teas-yang-path-computation-03.txt

Status of this Memo

This Internet-Draft will expire on April 22, 2019.

Copyright Notice

Abstract

There are scenarios, typically in a hierarchical SDN context, where the topology information provided by a TE network provider may not be sufficient for its client to perform end-to-end path computation. In these cases the client would need to request the provider to calculate some (partial) feasible paths.

This document defines a YANG data model for a stateless RPC to request path computation. This model complements the stateful solution defined in [TE-TUNNEL].

Moreover this document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

Table of Contents

1. Introduction

There are scenarios, typically in a hierarchical SDN context, where the topology information provided by a TE network provider may not be sufficient for its client to perform end-to-end path computation. In these cases the client would need to request the provider to calculate some (partial) feasible paths, complementing his topology knowledge, to make his end-to-end path computation feasible.

This type of scenarios can be applied to different interfaces in different reference architectures:

- ABNO control interface [RFC7491], in which an Application Service Coordinator can request ABNO controller to take in charge path calculation (see Figure 1 in [RFC7491]).

- ACTN [ACTN-frame], where a controller hierarchy is defined, the need for path computation arises on both interfaces CMI (interface between Customer Network Controller (CNC) and Multi Domain Service Coordinator (MDSC)) and/or MPI (interface between MSDC-PNC). [ACTN-Info] describes an information model for the Path Computation request.

Multiple protocol solutions can be used for communication between different controller hierarchical levels. This document assumes that the controllers are communicating using YANG-based protocols (e.g., NETCONF or RESTCONF).

Path Computation Elements, Controllers and Orchestrators perform their operations based on Traffic Engineering Databases (TED). Such TEDs can be described, in a technology agnostic way, with the YANG Data Model for TE Topologies [TE-TOPO]. Furthermore, the technology specific details of the TED are modeled in the augmented TE topology models (e.g. [OTN-TOPO] for OTN ODU technologies).

The availability of such topology models allows providing the TED using YANG-based protocols (e.g., NETCONF or RESTCONF). Furthermore, it enables a PCE/Controller performing the necessary abstractions or modifications and offering this customized topology to another PCE/Controller or high level orchestrator.

Note: This document assumes that the client of the YANG data model defined in this document may not implement a "PCE" functionality, as defined in [RFC4655].

The tunnels that can be provided over the networks described with the topology models can be also set-up, deleted and modified via YANG-based protocols (e.g., NETCONF or RESTCONF) using the TE-Tunnel Yang model [TE-TUNNEL].

This document proposes a YANG model for a path computation request defined as a stateless RPC, which complements the stateful solution defined in [TE-TUNNEL].

Moreover, this document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

1.1. Terminology

TED: The traffic engineering database is a collection of all TE information about all TE nodes and TE links in a given network.

PCE: A Path Computation Element (PCE) is an entity that is capable of computing a network path or route based on a network graph, and of applying computational constraints during the computation. The PCE entity is an application that can be located within a network node or component, on an out-of-network server, etc. For example, a

PCE would be able to compute the path of a TE LSP by operating on the TED and considering bandwidth and other constraints applicable to the TE LSP service request. [RFC4655]

2. Use Cases

This section presents different use cases, where a client needs to request underlying SDN controllers for path computation.

The presented uses cases have been grouped, depending on the different underlying topologies: a) Packet-Optical integration; b) Multi-domain Traffic Engineered (TE) Networks; and c) Data center interconnections.

2.1. Packet/Optical Integration

In this use case, an Optical network is used to provide connectivity to some nodes of a Packet network (see Figure 1).

```
                                +----------------+
                                |                |
                                | Packet/Optical |
                                |  Coordinator   |
                                |                |
                                +---+------+-----+
                                    |      |
                       +------------+      |
                       |                   +------------+
                +------V-----+                          |
                |            |                  +-------V-----+
                | Packet     |                  |             |
                | Network    |                  | Optical     |
                | Controller |                  | Network     |
                |            |                  | Controller  |
                +------+-----+                  +-------+-----+
                       |                                |
              .........V.........................       |
              :          Packet Network         :       |
          +----+                               +----+   |
          | R1 |= = = = = = = = = = = = = = = =| R2 |   |
          +-+--+                               +--+-+   |
            | :                                 : |     |
            | :................................ : |     |
            |                                     |     |
            |               +-----+               |     |
            |    ...........| Opt |...........    |     |
            |    :          |  C  |          :    |     |
            |    :         /+--+--+\         :    |     |
            |    :        /    |    \        :    |     |
            |    :       /     |     \       :    |     |
            |   +-----+ /   +--+--+   \ +-----+   |     |
            |   | Opt |/    | Opt |    \| Opt |   |     |
            +---|  A  |     |  D  |     |  B  |---+     |
                +-----+\    +--+--+    /+-----+         |
                 :      \      |      /      :          |
                 :       \     |     /       :          |
                 :        \ +--+--+  / Optical<---------+
                 :         \| Opt |/  Network:
                 :..........|  E  |..........:
                            +-----+
```

Figure 1 - Packet/Optical Integration Use Case

Figure 1 as well as Figure 2 below only show a partial view of the packet network connectivity, before additional packet connectivity is provided by the Optical network.

It is assumed that the Optical network controller provides to the packet/optical coordinator an abstracted view of the Optical network. A possible abstraction could be to represent the whole optical network as one "virtual node" with "virtual ports" connected to the access links, as shown in Figure 2.

It is also assumed that Packet network controller can provide the packet/optical coordinator the information it needs to setup connectivity between packet nodes through the Optical network (e.g., the access links).

The path computation request helps the coordinator to know the real connections that can be provided by the optical network.

```
                       ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,.
                      ,  Packet/Optical Coordinator view          ,
                     ,                              +----+       , .
                    ,                               |    |      ,
                   ,                                | R2 |     ,   .
                  ,  +----+  +------------ +       /+----+    ,
                 ,   |    |  |             |/-----/ / /      ,     .
                ,    | R1 |--O VP1     VP4 O       / /      ,
               ,     |    |\ |             | /----/ /      ,       .
              ,      +----+ \|             |/      /      ,
             ,        /      O VP2     VP5 O      /      ,         .
            ,        /       |             |  +----+    ,
           ,        /        |             |  |    |   ,           .
          ,        /         O VP3     VP6 O--| R4 |  ,
         ,     +----+ /-----/|_____________|  +----+ ,             .
        ,      |    |/       +------------ +        ,
       ,       | R3 |                              ,               .
      ,        +----+                             ,,,,,,,,,,,,,,,,,
     ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,                ,.
     . Packet Network Controller view               +----+       ,
       only packet nodes and packet links           |    |      ,  .
     . with access links to the optical network     | R2 |     ,
                  ,  +----+                        /+----+    ,    .
     .           ,   |    |                 /-----/ / /      ,
                ,    | R1 |---                     / /      ,      .
     .         ,     +----+\                 /----/ /      ,
              ,        /    \               /      /      ,        .
     .       ,        /                           /      ,
            ,        /                        +----+    ,          .
     .     ,        /                         |    |   ,
          ,        /                       ---| R4 |  ,            .
     .   ,     +----+ /-----/                 +----+ ,
        ,      |    |/                              ,              .
     . ,       | R3 |                              ,
      ,        +----+                             ,,,,,,,,,,,,,,,,,.
     .,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,                ,
       Optical Network Controller view                           , .
     . only optical nodes,        +--+                          ,
       optical links and         /|OF|                         ,   .
     . access links from the  +--++--+             /          ,
       packet network         |OA|    \     /-----/ /        ,     .
     .          ,          ---+--+--\  +--+/       /        ,
               ,           \   |  \  \-|OE|-------/        ,       .
     .        ,             \  |   \ /-+--+               ,
             ,               \+--+  X    |               ,         .
     .      ,                 |OB|-/ \   |              ,
           ,                  +--+-\  \+--+            ,           .
     .    ,                  /   \  \--|OD|---        ,
         ,            /-----/     +--+ +--+          ,             .
     .  ,            /            |OC|/             ,
       ,                          +--+             ,               .
     .,                                           ,,,,,,,,,,,,,,,,,,
      ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,                ,
     . Actual Physical View                         +----+       ,
                    ,             +--+              |    |      ,
     .             ,             /|OF|              | R2 |     ,
                  ,  +----+   +--++--+             /+----+    ,
     .           ,   |    |   |OA|    \     /-----/ / /      ,
                ,    | R1 |---+--+--\  +--+/       / /      ,
     .         ,     +----+\   |  \  \-|OE|-------/ /      ,
              ,        /    \  |   \ /-+--+        /      ,
     .       ,        /      \+--+  X    |        /      ,
            ,        /        |OB|-/ \   |    +----+    ,
     .     ,        /         +--+-\  \+--+   |    |   ,
          ,        /         /   \  \--|OD|---| R4 |  ,
     .   ,     +----+ /-----/     +--+ +--+   +----+ ,
        ,      |    |/            |OC|/             ,
     . ,       | R3 |             +--+             ,
      ,        +----+                             ,
     .,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
```

Figure 2 - Packet and Optical Topology Abstractions

In this use case, the coordinator needs to setup an optimal underlying path for an IP link between R1 and R2.

As depicted in Figure 2, the coordinator has only an "abstracted view" of the physical network, and it does not know the feasibility or the cost of the possible optical paths (e.g., VP1-VP4 and VP2-VP5), which depend from the current status of the physical resources within the optical network and on vendor-specific optical attributes.

The coordinator can request the underlying Optical domain controller to compute a set of potential optimal paths, taking into account optical constraints. Then, based on its own constraints, policy and knowledge (e.g. cost of the access links), it can choose which one of these potential paths to use to setup the optimal end-to-end path crossing optical network.

```
                    ............................
                    :                          :
                    O VP1                  VP4 O
           cost=10 /:\                        /:\ cost=10
                  / : \----------------------/ : \
          +----+ /  :         cost=50          :  \ +----+
          |    |/   :                          :   \|    |
          | R1 |    :                          :    | R2 |
          |    |\   :                          :   /|    |
          +----+ \  :  /--------------------\  :  / +----+
                  \ : /       cost=55        \ : /
            cost=5 \:/                        \:/ cost=5
                    O VP2                  VP5 O
                    :                          :
                    :..........................:
```

Figure 3 - Packet/Optical Path Computation Example

For example, in Figure 3, the Coordinator can request the Optical network controller to compute the paths between VP1-VP4 and VP2-VP5 and then decide to setup the optimal end-to-end path using the VP2-VP5 Optical path even this is not the optimal path from the Optical domain perspective.

Considering the dynamicity of the connectivity constraints of an Optical domain, it is possible that a path computed by the Optical network controller when requested by the Coordinator is no longer valid/available when the Coordinator requests it to be setup up. This is further discussed in section 3.3.

2.2. Multi-domain TE Networks

In this use case there are two TE domains which are interconnected together by multiple inter-domains links.

A possible example could be a multi-domain optical network.

```
                            +--------------+
                            | Multi-domain |
                            | Controller   |
                            +---+------+---+
                                |      |
                   +------------+      |
                   |                   +-----------+
            +------V-----+                         |
            |            |                         |
            | TE Domain  |                  +------V-----+
            | Controller |                  |            |
            |      1     |                  | TE Domain  |
            +------+-----+                  | Controller |
                   |                        |      2     |
                   |                        +------+-----+
          .........V..........                     |
          :                  :                     |
         +-----+             :                     |
         |     |             :            .........V..........
         |  X  |             :            :                  :
         |     |          +-----+      +-----+                :
         +-----+          |     |      |     |               :
          :               |  C  |------|  E  |               :
      +-----+    +-----+ /|     |      |     |\ +-----+    +-----+
      |     |    |     |/ +-----+      +-----+ \|     |    |     |
      |  A  |----|  B  |     :            :     |  G  |----|  H  |
      |     |    |     |\    :            :    /|     |    |     |
      +-----+    +-----+ \+-----+      +-----+/ +-----+    +-----+
          :               |     |      |     |               :
          :               |  D  |------|  F  |               :
          :               |     |      |     |          +-----+
          :               +-----+      +-----+          |     |
          :                  :            :             |  Y  |
          :                  :            :             |     |
          :    Domain 1      :            : Domain 2    +-----+
          :..................:            :.................:
```

Figure 4 - Multi-domain multi-link interconnection

In order to setup an end-to-end multi-domain TE path (e.g., between nodes A and H), the multi-domain controller needs to know the feasibility or the cost of the possible TE paths within the two TE domains, which depend from the current status of the physical resources within each TE network. This is more challenging in case of optical networks 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).

In order to setup a multi-domain TE path (e.g., between nodes A and H), the multi-domain controller can request the TE domain controllers to compute a set of intra-domain optimal paths and take decisions based on the information received. For example:

- The multi-domain controller asks TE domain controllers to provide set of paths between A-C, A-D, E-H and F-H

- TE domain controllers return a set of feasible paths with the associated costs: the path A-C is not part of this set(in optical networks, it is typical to have some paths not being feasible due to optical constraints that are known only by the optical domain controller)

- The multi-domain controller will select the path A-D-F-H since it is the only feasible multi-domain path and then request the TE domain controllers to setup the A-D and F-H intra-domain paths

- If there are multiple feasible paths, the multi-domain controller can select the optimal path knowing the cost of the intra-domain paths (provided by the TE domain controllers) and the cost of the inter-domain links (known by the multi-domain controller)

This approach may have some scalability issues when the number of TE domains is quite big (e.g. 20).

In this case, it would be worthwhile using the abstract TE topology information provided by the TE domain controllers to limit the number of potential optimal end-to-end paths and then request path computation to fewer TE domain controllers in order to decide what the optimal path within this limited set is.

For more details, see section 3.2.3.

2.3. Data center interconnections

In these use case, there is a TE domain which is used to provide connectivity between data centers which are connected with the TE domain using access links.

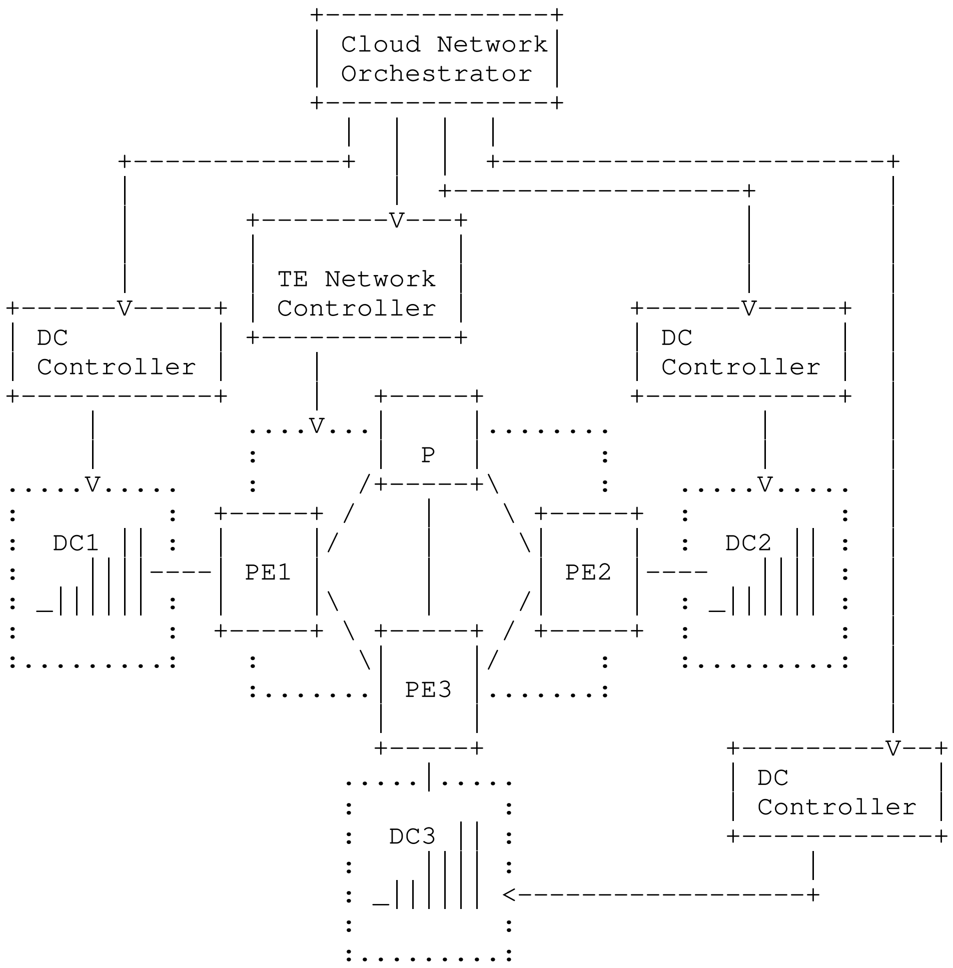

Figure 5 - Data Center Interconnection Use Case

In this use case, there is need to transfer data from Data Center 1 (DC1) to either DC2 or DC3 (e.g. workload migration).

The optimal decision depends both on the cost of the TE path (DC1-DC2 or DC1-DC3) and of the data center resources within DC2 or DC3.

The cloud network orchestrator needs to make a decision for optimal connection based on TE Network constraints and data centers resources. It may not be able to make this decision because it has only an abstract view of the TE network (as in use case in 2.1).

The cloud network orchestrator can request to the TE network controller to compute the cost of the possible TE paths (e.g., DC1-DC2 and DC1-DC3) and to the DC controller to provide the information it needs about the required data center resources within DC2 and DC3 and then it can take the decision about the optimal solution based on this information and its policy.

3. Motivations

This section provides the motivation for the YANG model defined in this document.

Section 3.1 describes the motivation for a YANG model to request path computation.

Section 3.2 describes the motivation for a YANG model which complements the TE Topology YANG model defined in [TE-TOPO].

Section 3.3 describes the motivation for a stateless YANG RPC which complements the TE Tunnel YANG model defined in [TE-TUNNEL].

3.1. Motivation for a YANG Model

3.1.1. Benefits of common data models

The YANG data model for requesting path computation is closely aligned with the YANG data models that provide (abstract) TE topology information, i.e., [TE-TOPO] as well as that are used to configure and manage TE Tunnels, i.e., [TE-TUNNEL].

There are many benefits in aligning the data model used for path computation requests with the YANG data models used for TE topology information and for TE Tunnels configuration and management:

- There is no need for an error-prone mapping or correlation of information.

- It is possible to use the same endpoint identifiers in path computation requests and in the topology modeling.

- The attributes used for path computation constraints are the same as those used when setting up a TE Tunnel.

3.1.2. Benefits of a single interface

The system integration effort is typically lower if a single, consistent interface is used by controllers, i.e., one data modeling language (i.e., YANG) and a common protocol (e.g., NETCONF or RESTCONF).

Practical benefits of using a single, consistent interface include:

1. Simple authentication and authorization: The interface between different components has to be secured. If different protocols have different security mechanisms, ensuring a common access control model may result in overhead. For instance, there may be a need to deal with different security mechanisms, e.g., different credentials or keys. This can result in increased integration effort.

2. Consistency: Keeping data consistent over multiple different interfaces or protocols is not trivial. For instance, the sequence of actions can matter in certain use cases, or transaction semantics could be desired. While ensuring consistency within one protocol can already be challenging, it is typically cumbersome to achieve that across different protocols.

3. Testing: System integration requires comprehensive testing, including corner cases. The more different technologies are involved, the more difficult it is to run comprehensive test cases and ensure proper integration.

4. Middle-box friendliness: Provider and consumer of path computation requests may be located in different networks, and middle-boxes such as firewalls, NATs, or load balancers may be deployed. In such environments it is simpler to deploy a single protocol. Also, it may be easier to debug connectivity problems.

5. Tooling reuse: Implementers may want to implement path computation requests with tools and libraries that already exist in controllers and/or orchestrators, e.g., leveraging the rapidly growing eco-system for YANG tooling.

3.1.3. Extensibility

Path computation is only a subset of the typical functionality of a controller. In many use cases, issuing path computation requests comes along with the need to access other functionality on the same system. In addition to obtaining TE topology, for instance also configuration of services (setup/modification/deletion) may be required, as well as:

1. Receiving notifications for topology changes as well as integration with fault management

2. Performance management such as retrieving monitoring and telemetry data

3. Service assurance, e.g., by triggering OAM functionality

4. Other fulfilment and provisioning actions beyond tunnels and services, such as changing QoS configurations

YANG is a very extensible and flexible data modeling language that can be used for all these use cases.

3.2. Interactions with TE Topology

The use cases described in section 2 have been described assuming that the topology view exported by each underlying SDN controller to the orchestrator is aggregated using the "virtual node model", defined in [RFC7926].

TE Topology information, e.g., as provided by [TE-TOPO], could in theory be used by an underlying SDN controllers to provide TE information to its client thus allowing a PCE available within its client to perform multi-domain path computation by its own, without requesting path computations to the underlying SDN controllers.

In case the client does not implement a PCE function, as discussed in section 1, it could not perform path computation based on TE Topology information and would instead need to request path computation to the underlying controllers to get the information it needs to compute the optimal end-to-end path.

This section analyzes the need for a client to request underlying SDN controllers for path computation even in case it implements a PCE functionality, as well as how the TE Topology information and the path computation can be complementary.

In nutshell, there is a scalability trade-off between providing all the TE information needed by PCE, when implemented by the client, to take optimal path computation decisions by its own versus sending too many requests to underlying SDN Domain Controllers to compute a set of feasible optimal intra-domain TE paths.

3.2.1. TE Topology Aggregation

Using the TE Topology model, as defined in [TE-TOPO], the underlying SDN controller can export the whole TE domain as a single abstract TE node with a "detailed connectivity matrix".

The concept of a "detailed connectivity matrix" is defined in [TE-TOPO] to provide specific TE attributes (e.g., delay, SRLGs and summary TE metrics) as an extension of the "basic connectivity matrix", which is based on the "connectivity matrix" defined in [RFC7446].

The information provided by the "detailed connectivity matrix" would be equivalent to the information that should be provided by "virtual link model" as defined in [RFC7926].

For example, in the Packet/Optical integration use case, described in section 2.1, the Optical network controller can make the information shown in Figure 3 available to the Coordinator as part of the TE Topology information and the Coordinator could use this information to calculate by its own the optimal path between R1 and R2, without requesting any additional information to the Optical network Controller.

However, when designing the amount of information to provide within the "detailed connectivity matrix", there is a tradeoff to be considered between accuracy (i.e., providing "all" the information that might be needed by the PCE available to Orchestrator) and scalability.

Figure 6 below shows another example, similar to Figure 3, where there are two possible Optical paths between VP1 and VP4 with different properties (e.g., available bandwidth and cost).

```
                    ............................
                    :  /--------------------\  :
                    : /       cost=65        \ :
                    :/    available-bw=10G    \:
                    O VP1                  VP4 O
           cost=10 /:\                        /:\ cost=10
                  / : \----------------------/ : \
          +----+ /  :         cost=50          :  \ +----+
          |    |/   :     available-bw=2G      :   \|    |
          | R1 |    :                          :    | R2 |
          |    |\   :                          :   /|    |
          +----+ \  :  /--------------------\  :  / +----+
                  \ : /       cost=55        \ : /
            cost=5 \:/    available-bw=3G     \:/ cost=5
                    O VP2                  VP5 O
                    :                          :
                    :..........................:
```

Figure 6 - Packet/Optical Path Computation Example with multiple choices

Reporting all the information, as in Figure 6, using the "detailed connectivity matrix", is quite challenging from a scalability perspective. The amount of this information is not just based on number of end points (which would scale as N-square), but also on many other parameters, including client rate, user constraints/policies for the service, e.g. max latency < N ms, max cost, etc., exclusion policies to route around busy links, min OSNR margin, max preFEC BER etc. All these constraints could be different based on connectivity requirements.

Examples of how the "detailed connectivity matrix" can be dimensioned are described in Appendix A.

It is also worth noting that the "connectivity matrix" has been originally defined in WSON, [RFC7446], to report the connectivity constrains of a physical node within the WDM network: the information it contains is pretty "static" and therefore, once taken and stored in the TE data base, it can be always being considered valid and up-to-date in path computation request.

Using the "basic connectivity matrix" with an abstract node to abstract the information regarding the connectivity constraints of an Optical domain, would make this information more "dynamic" since the connectivity constraints of an Optical domain can change over time because some optical paths that are feasible at a given time may become unfeasible at a later time when e.g., another optical path is established. The information in the "detailed connectivity matrix" is even more dynamic since the establishment of another optical path may change some of the parameters (e.g., delay or available bandwidth) in the "detailed connectivity matrix" while not changing the feasibility of the path.

The "connectivity matrix" is sometimes confused with optical reach table that contain multiple (e.g. k-shortest) regen-free reachable paths for every A-Z node combination in the network. Optical reach tables can be calculated offline, utilizing vendor optical design and planning tools, and periodically uploaded to the Controller: these optical path reach tables are fairly static. However, to get the connectivity matrix, between any two sites, either a regen free path can be used, if one is available, or multiple regen free paths are concatenated to get from src to dest, which can be a very large combination. Additionally, when the optical path within optical domain needs to be computed, it can result in different paths based on input objective, constraints, and network conditions. In summary, even though "optical reachability table" is fairly static, which regen free paths to build the connectivity matrix between any source and destination is very dynamic, and is done using very sophisticated routing algorithms.

There is therefore the need to keep the information in the "detailed connectivity matrix" updated which means that there another tradeoff between the accuracy (i.e., providing "all" the information that might be needed by the client's PCE) and having up-to-date information. The more the information is provided and the longer it takes to keep it up-to-date which increases the likelihood that the client's PCE computes paths using not updated information.

It seems therefore quite challenging to have a "detailed connectivity matrix" that provides accurate, scalable and updated information to allow the client's PCE to take optimal decisions by its own.

Instead, if the information in the "detailed connectivity matrix" is not complete/accurate, we can have the following drawbacks considering for example the case in Figure 6:

- If only the VP1-VP4 path with available bandwidth of 2 Gb/s and cost 50 is reported, the client's PCE will fail to compute a 5 Gb/s path between routers R1 and R2, although this would be feasible;

- If only the VP1-VP4 path with available bandwidth of 10 Gb/s and cost 60 is reported, the client's PCE will compute, as optimal, the 1 Gb/s path between R1 and R2 going through the VP2-VP5 path within the Optical domain while the optimal path would actually be the one going thought the VP1-VP4 sub-path (with cost 50) within the Optical domain.

Using the approach proposed in this document, the client, when it needs to setup an end-to-end path, it can request the Optical domain controller to compute a set of optimal paths (e.g., for VP1-VP4 and VP2-VP5) and take decisions based on the information received:

- When setting up a 5 Gb/s path between routers R1 and R2, the Optical domain controller may report only the VP1-VP4 path as the only feasible path: the Orchestrator can successfully setup the end-to-end path passing though this Optical path;

- When setting up a 1 Gb/s path between routers R1 and R2, the Optical domain controller (knowing that the path requires only 1 Gb/s) can report both the VP1-VP4 path, with cost 50, and the VP2-VP5 path, with cost 65. The Orchestrator can then compute the optimal path which is passing thought the VP1-VP4 sub-path (with cost 50) within the Optical domain.

3.2.2. TE Topology Abstraction

Using the TE Topology model, as defined in [TE-TOPO], the underlying SDN controller can export an abstract TE Topology, composed by a set of TE nodes and TE links, representing the abstract view of the topology controlled by each domain controller.

Considering the example in Figure 4, the TE domain controller 1 can export a TE Topology encompassing the TE nodes A, B, C and D and the TE Link interconnecting them. In a similar way, TE domain controller 2 can export a TE Topology encompassing the TE nodes E, F, G and H and the TE Link interconnecting them.

In this example, for simplicity reasons, each abstract TE node maps with each physical node, but this is not necessary.

In order to setup a multi-domain TE path (e.g., between nodes A and H), the multi-domain controller can compute by its own an optimal end-to-end path based on the abstract TE topology information provided by the domain controllers. For example:

- o Multi-domain controller's PCE, based on its own information, can compute the optimal multi-domain path being A-B-C-E-G-H, and then request the TE domain controllers to setup the A-B-C and E-G-H intra-domain paths

- o But, during path setup, the domain controller may find out that A-B-C intra-domain path is not feasible (as discussed in section 2.2, in optical networks it is typical to have some paths not being feasible due to optical constraints that are known only by the optical domain controller), while only the path A-B-D is feasible

- o So what the multi-domain controller computed is not good and need to re-start the path computation from scratch

As discussed in section 3.2.1, providing more extensive abstract information from the TE domain controllers to the multi-domain controller may lead to scalability problems.

In a sense this is similar to the problem of routing and wavelength assignment within an Optical domain. It is possible to do first routing (step 1) and then wavelength assignment (step 2), but the chances of ending up with a good path is low. Alternatively, it is possible to do combined routing and wavelength assignment, which is known to be a more optimal and effective way for Optical path setup. Similarly, it is possible to first compute an abstract end-to-end path within the multi-domain Orchestrator (step 1) and then compute an intra-domain path within each Optical domain (step 2), but there are more chances not to find a path or to get a suboptimal path that performing per-domain path computation and then stitch them.

3.2.3. Complementary use of TE topology and path computation

As discussed in section 2.2, there are some scalability issues with path computation requests in a multi-domain TE network with many TE domains, in terms of the number of requests to send to the TE domain controllers. It would therefore be worthwhile using the TE topology information provided by the domain controllers to limit the number of requests.

An example can be described considering the multi-domain abstract topology shown in Figure 7. In this example, an end-to-end TE path between domains A and F needs to be setup. The transit domain should be selected between domains B, C, D and E.

```
                          .........B.........
                          : _ _ _ _ _ _ _ _ :
                          :/               \:
                      +---O  NOT FEASIBLE   O---+
                cost=5|   :                 :   |
    ......A......     |   :.................:   |     ......F......
    :           :     |                         |     :           :
    :           O-----+   .........C.........   +-----O           :
    :           :         : /-------------\ :         :           :
    :           :         :/               \:         :           :
    :  cost<=20 O---------O   cost <= 30    O---------O cost<=20  :
    :          /: cost=5  :                 : cost=5  :\          :
    :  /------/ :         :.................:         : \------\  :
    : /         :                                     :         \ :
    :/ cost<=25 :         .........D.........         : cost<=25 \:
    O-----------O-------+ : /-------------\ : +-------O-----------O
    :\          : cost=5| :/               \: |cost=5 :          /:
    : \         :       +-O   cost <= 30    O-+       :         / :
    :  \------\ :         :                 :         : /------/  :
    : cost>=30 \:         :.................:         :/ cost>=30 :
    :           O-----+                         +-----O           :
    :...........:     |   .........E.........   |     :...........:
                      |   : /-------------\ :   |
                cost=5|   :/               \:   |cost=5
                      +---O   cost >= 30    O---+
                          :                 :
                          :.................:
```

Figure 7 - Multi-domain with many domains (Topology information)

The actual cost of each intra-domain path is not known a priori from the abstract topology information. The Multi-domain controller only knows, from the TE topology provided by the underlying domain controllers, the feasibility of some intra-domain paths and some upper-bound and/or lower-bound cost information. With this information, together with the cost of inter-domain links, the Multi-domain controller can understand by its own that:

- Domain B cannot be selected as the path connecting domains A and E is not feasible;

- Domain E cannot be selected as a transit domain since it is know from the abstract topology information provided by domain controllers that the cost of the multi-domain path A-E-F (which is 100, in the best case) will be always be higher than the cost of the multi-domain paths A-D-F (which is 90, in the worst case) and A-E-F (which is 80, in the worst case)

Therefore, the Multi-domain controller can understand by its own that the optimal multi-domain path could be either A-D-F or A-E-F but it cannot known which one of the two possible option actually provides the optimal end-to-end path.

The Multi-domain controller can therefore request path computation only to the TE domain controllers A, D, E and F (and not to all the possible TE domain controllers).

```
                          .........B.........
                          :                 :
                      +---O                 O---+
    ......A......     |   :.................:   |     ......F......
    :           :     |                         |     :           :
    :           O-----+   .........C.........   +-----O           :
    :           :         : /-------------\ :         :           :
    :           :         :/               \:         :           :
    :  cost=15  O---------O    cost = 25    O---------O  cost=10  :
    :          /: cost=5  :                 : cost=5  :\          :
    :  /------/ :         :.................:         : \------\  :
    : /         :                                     :         \ :
    :/ cost=10  :         .........D.........         : cost=15  \:
    O-----------O-------+ : /-------------\ : +-------O-----------O
    :           : cost=5| :/               \: |cost=5 :           :
    :           :       +-O    cost = 15    O-+       :           :
    :           :         :                 :         :           :
    :           :         :.................:         :           :
    :           O-----+                         +-----O           :
    :...........:     |   .........E.........   |     :...........:
                      |   :                 :   |
                      +---O                 O---+
                          :.................:
```

Figure 8 - Multi-domain with many domains (Path Computation information)

Based on these requests, the Multi-domain controller can know the actual cost of each intra-domain paths which belongs to potential optimal end-to-end paths, as shown in Figure 8, and then compute the optimal end-to-end path (e.g., A-D-F, having total cost of 50, instead of A-C-F having a total cost of 70).

3.3. Stateless and Stateful Path Computation

The TE Tunnel YANG model, defined in [TE-TUNNEL], can support the need to request path computation.

It is possible to request path computation by configuring a "compute-only" TE tunnel and retrieving the computed path(s) in the LSP(s) Record-Route Object (RRO) list as described in section 3.3.1 of [TE-TUNNEL].

This is a stateful solution since the state of each created "compute-only" TE tunnel needs to be maintained and updated, when underlying network conditions change.

It is very useful to provide options for both stateless and stateful path computation mechanisms. It is suggested to use stateless mechanisms as much as possible and to rely on stateful path computation when really needed.

Stateless RPC allows requesting path computation using a simple atomic operation and it is the natural option/choice, especially with stateless PCE.

Since the operation is stateless, there is no guarantee that the returned path would still be available when path setup is requested: this does not cause major issues in case the time between path computation and path setup is short (especially if compared with the time that would be needed to update the information of a very detailed connectivity matrix).

In most of the cases, there is even no need to guarantee that the path that has been setup is the exactly same as the path that has been returned by path computation, especially if has the same or even better metrics. Depending on the abstraction level applied by the server, the client may also not know the actual computed path.

The most important requirement is that the required global objectives (e.g., multi-domain path metrics and constraints) are met. For this reason a path verification phase is necessary to verify that the actual path that has been setup meets the global objectives (for example in a multi-domain network, the resulting end-to-end path meets the required end-to-end metrics and constraints).

In most of the cases, even if the setup path is not exactly the same as the path returned by path computation, its metrics and constraints are "good enough" (the path verification passes successfully). In the few corner cases where the path verification fails, it is possible repeat the whole process (path computation, path setup and path verification).

In case the stateless solution is not sufficient, a stateful solution, based on "compute-only" TE tunnel, could be used to get notifications in case the computed path has been changed.

It is worth noting that also the stateful solution, although increasing the likelihood that the computed path is available at path setup, does not guaranteed that because notifications may not be reliable or delivered on time. Path verification is needed also when stateful path computation is used.

The stateful path computation has also the following drawbacks:

- Several messages required for any path computation

- Requires persistent storage in the provider controller

- Need for garbage collection for stranded paths

- Process burden to detect changes on the computed paths in order to provide notifications update

4. Path Computation and Optimization for multiple paths

There are use cases, where it is advantageous to request path computation for a set of paths, through a network or through a network domain, using a single request [RFC5440].

In this case, sending a single request for multiple path computations, instead of sending multiple requests for each path computation, would reduce the protocol overhead and it would consume less resources (e.g., threads in the client and server).

In the context of a typical multi-domain TE network, there could multiple choices for the ingress/egress points of a domain and the Multi-domain controller needs to request path computation between all the ingress/egress pairs to select the best pair. For example, in the example of section 2.2, the Multi-domain controller needs to request the TE network controller 1 to compute the A-C and the A-D paths and to the TE network controller 2 to compute the E-H and the F-H paths.

It is also possible that the Multi-domain controller receives a request to setup a group of multiple end to end connections. The multi-domain controller needs to request each TE domain controller to compute multiple paths, one (or more) for each end to end connection.

There are also scenarios where it can be needed to request path computation for a set of paths in a synchronized fashion.

One example could be computing multiple diverse paths. Computing a set of diverse paths in a not-synchronized fashion, leads to the possibility of not being able to satisfy the diversity requirement. In this case, it is preferable to compute a sub-optimal primary path for which a diversely routed secondary path exists.

There are also scenarios where it is needed to request optimizing a set of paths using objective functions that apply to the whole set of paths, see [RFC5541], e.g. to minimize the sum of the costs of all the computed paths in the set.

5. YANG Model for requesting Path Computation

This document define a YANG stateless RPC to request path computation as an "augmentation" of tunnel-rpc, defined in [TE-TUNNEL]. This model provides the RPC input attributes that are needed to request path computation and the RPC output attributes that are needed to report the computed paths.

```
augment /te:tunnels-rpc/te:input/te:tunnel-info:
  +---- path-request* [request-id]
  ...........

augment /te:tunnels-rpc/te:output/te:result:
  +--ro response* [response-id]
     +--ro response-id      uint32
     +--ro (response-type)?
        +--:(no-path-case)
```

```
             |  +--ro no-path!
             +--:(path-case)
                +--ro computed-path
                   +--ro path-id?           yang-types:uuid
                   +--ro path-properties
                   ...........
```

This model extensively re-uses the grouping defined in [TE-TUNNEL] to ensure maximal syntax and semantics commonality.

5.1. Synchronization of multiple path computation requests

The YANG model permits to synchronize a set of multiple path requests (identified by specific request-id) all related to a "svec" container emulating the syntax of "SVEC" PCEP object [RFC 5440].

```
       +---- synchronization* [synchronization-id]
          +---- synchronization-id    uint32
          +---- svec
          |  +---- relaxable?           boolean
          |  +---- disjointness?        te-types:te-path-disjointness
          |  +---- request-id-number*   uint32
          +---- svec-constraints
          |  +---- path-metric-bound* [metric-type]
          |     +---- metric-type    identityref
          |     +---- upper-bound?   uint64
          +---- path-srlgs-values
          |  +---- usage?    identityref
          |  +---- values*   srlg
          +---- path-srlgs-names
          |  +---- path-srlgs-name* [usage]
          |     +---- usage        identityref
          |     +---- srlg-name* [name]
          |        +---- name    string
          +---- exclude-objects
          ...........
          +---- optimizations
             +---- (algorithm)?
                +--:(metric)
                |  +---- optimization-metric* [metric-type]
                |     +---- metric-type    identityref
                |     +---- weight?        uint8
                +--:(objective-function)
                   +---- objective-function
                      +---- objective-function-type?   identityref
```

The model, in addition to the metric types, defined in [TE-TUNNEL], which can be applied to each individual path request, defines additional specific metrics types that apply to a set of synchronized requests, as referenced in [RFC5541].

```
     identity svec-metric-type {
       description
         "Base identity for svec metric type";
     }

     identity svec-metric-cumul-te {
       base svec-metric-type;
       description
         "TE cumulative path metric";
     }

     identity svec-metric-cumul-igp {
       base svec-metric-type;
       description
         "IGP cumulative path metric";
     }

     identity svec-metric-cumul-hop {
       base svec-metric-type;
       description
         "Hop cumulative path metric";
     }

     identity svec-metric-aggregate-bandwidth-consumption {
       base svec-metric-type;
       description
         "Cumulative bandwith consumption of the set of synchronized
   paths";
     }
     identity svec-metric-load-of-the-most-loaded-link {
       base svec-metric-type;
       description
         "Load of the most loaded link";
     }
```

5.2. Returned metric values

This YANG model provides a way to return the values of the metrics computed by the path computation in the output of RPC, together with other important information (e.g. srlg, affinities, explicit route), emulating the syntax of the "C" flag of the "METRIC" PCEP object [RFC 5440]:

```
     augment /te:tunnels-rpc/te:output/te:result:
       +--ro response* [response-id]
          +--ro response-id           uint32
          +--ro (response-type)?
             +--:(no-path-case)
             |  +--ro no-path!
             +--:(path-case)
                +--ro computed-path
                   +--ro path-id?           yang-types:uuid
                   +--ro path-properties
                      +--ro path-metric* [metric-type]
                      |  +--ro metric-type           identityref
                      |  +--ro accumulative-value?   uint64
                      +--ro path-affinities-values
                      |  +--ro path-affinities-value* [usage]
                      |     +--ro usage    identityref
                      |     +--ro value?   admin-groups
                      +--ro path-affinity-names
                      |  +--ro path-affinity-name* [usage]
                      |     +--ro usage            identityref
                      |     +--ro affinity-name* [name]
                      |        +--ro name    string
                      +--ro path-srlgs-values
                      |  +--ro usage?    identityref
                      |  +--ro values*   srlg
```

```
+--ro path-srlgs-names
|  +--ro path-srlgs-name* [usage]
|     +--ro usage        identityref
|     +--ro srlg-name* [name]
|        +--ro name    string
+--ro path-route-objects
...........
```

It also allows to request in the input of RPC which information (metrics, srlg and/or affinities) should be returned:

```
augment /te:tunnels-rpc/te:input/te:tunnel-info:
  +---- path-request* [request-id]
  |  +---- request-id                uint32
      ...........
  |  +---- requested-metrics* [metric-type]
  |  |  +---- metric-type    identityref
  |  +---- return-srlgs?             boolean
  |  +---- return-affinities?        Boolean
      ...........
```

This feature is essential for using a stateless path computation in a multi-domain TE network as described in section 2.2. In this case, the metrics returned by a path computation requested to a given TE network controller must be used by the client to compute the best end-to-end path. If they are missing the client cannot compare different paths calculated by the TE network controllers and choose the best one for the optimal e2e path.

6. YANG model for stateless TE path computation

6.1. YANG Tree

Figure 9 below shows the tree diagram of the YANG model defined in module ietf-te-path-computation.yang.

```
module: ietf-te-path-computation
  augment /te:tunnels-rpc/te:input/te:tunnel-info:
    +---- path-request* [request-id]
    |  +---- request-id                uint32
    |  +---- te-topology-identifier
```

```
       |  |  +---- provider-id?   te-types:te-global-id
       |  |  +---- client-id?     te-types:te-global-id
       |  |  +---- topology-id?   te-types:te-topology-id
       |  +---- source?                   inet:ip-address
       |  +---- destination?              inet:ip-address
       |  +---- src-tp-id?                binary
       |  +---- dst-tp-id?                binary
       |  +---- bidirectional?            boolean
       |  +---- encoding?                 identityref
       |  +---- switching-type?           identityref
       |  +---- explicit-route-objects
       |  |  +---- route-object-exclude-always* [index]
       |  |  |  +---- index            uint32
       |  |  |  +---- (type)?
       |  |  |     +--:(num-unnum-hop)
       |  |  |     |  +---- num-unnum-hop
       |  |  |     |     +---- node-id?      te-types:te-node-id
       |  |  |     |     +---- link-tp-id?   te-types:te-tp-id
       |  |  |     |     +---- hop-type?     te-hop-type
       |  |  |     |     +---- direction?    te-link-direction
       |  |  |     +--:(as-number)
       |  |  |     |  +---- as-number-hop
       |  |  |     |     +---- as-number?   binary
       |  |  |     |     +---- hop-type?    te-hop-type
       |  |  |     +--:(label)
       |  |  |        +---- label-hop
       |  |  |           +---- te-label
       |  |  |              +---- (technology)?
       |  |  |              |  +--:(generic)
       |  |  |              |     +---- generic?     rt-
   types:generalized-label
       |  |  |              +---- direction?   te-label-direction
       |  |  +---- route-object-include-exclude* [index]
       |  |     +---- explicit-route-usage?   identityref
       |  |     +---- index                   uint32
       |  |     +---- (type)?
       |  |        +--:(num-unnum-hop)
       |  |        |  +---- num-unnum-hop
       |  |        |     +---- node-id?      te-types:te-node-id
       |  |        |     +---- link-tp-id?   te-types:te-tp-id
       |  |        |     +---- hop-type?     te-hop-type
       |  |        |     +---- direction?    te-link-direction
       |  |        +--:(as-number)
       |  |        |  +---- as-number-hop
       |  |        |     +---- as-number?   binary
       |  |        |     +---- hop-type?    te-hop-type
       |  |        +--:(label)
       |  |        |  +---- label-hop
       |  |        |     +---- te-label
       |  |        |        +---- (technology)?
       |  |        |        |  +--:(generic)
       |  |        |        |     +---- generic?     rt-
   types:generalized-label
       |  |        |        +---- direction?   te-label-direction
       |  |        +--:(srlg)
       |  |           +---- srlg
       |  |              +---- srlg?   uint32
       |  +---- path-constraints
       |  |  +---- te-bandwidth
       |  |  |  +---- (technology)?
       |  |  |     +--:(generic)
       |  |  |        +---- generic?   te-bandwidth
       |  |  +---- setup-priority?           uint8
       |  |  +---- hold-priority?            uint8
       |  |  +---- signaling-type?           identityref
       |  |  +---- path-metric-bounds
       |  |  |  +---- path-metric-bound* [metric-type]
       |  |  |     +---- metric-type    identityref
       |  |  |     +---- upper-bound?   uint64
       |  |  +---- path-affinities-values
       |  |  |  +---- path-affinities-value* [usage]
       |  |  |     +---- usage    identityref
       |  |  |     +---- value?   admin-groups
       |  |  +---- path-affinity-names
       |  |  |  +---- path-affinity-name* [usage]
       |  |  |     +---- usage            identityref
       |  |  |     +---- affinity-name* [name]
       |  |  |        +---- name    string
       |  |  +---- path-srlgs-values
       |  |  |  +---- usage?    identityref
       |  |  |  +---- values*   srlg
       |  |  +---- path-srlgs-names
       |  |  |  +---- path-srlgs-name* [usage]
       |  |  |     +---- usage        identityref
       |  |  |     +---- srlg-name* [name]
       |  |  |        +---- name    string
       |  |  +---- disjointness?             te-types:te-path-
   disjointness
       |  +---- optimizations
       |  |  +---- (algorithm)?
       |  |     +--:(metric) {path-optimization-metric}?
       |  |     |  +---- optimization-metric* [metric-type]
       |  |     |  |  +---- metric-type
   identityref
       |  |     |  |  +---- weight?                           uint8
       |  |     |  |  +---- explicit-route-exclude-objects
       |  |     |  |  |  +---- route-object-exclude-object* [index]
       |  |     |  |  |     +---- index            uint32
       |  |     |  |  |     +---- (type)?
       |  |     |  |  |        +--:(num-unnum-hop)
       |  |     |  |  |        |  +---- num-unnum-hop
       |  |     |  |  |        |     +---- node-id?      te-types:te-
   node-id
       |  |     |  |  |        |     +---- link-tp-id?   te-types:te-
   tp-id
       |  |     |  |  |        |     +---- hop-type?     te-hop-type
       |  |     |  |  |        |     +---- direction?    te-link-
   direction
       |  |     |  |  |        +--:(as-number)
       |  |     |  |  |        |  +---- as-number-hop
       |  |     |  |  |        |     +---- as-number?   binary
       |  |     |  |  |        |     +---- hop-type?    te-hop-type
       |  |     |  |  |        +--:(label)
       |  |     |  |  |        |  +---- label-hop
       |  |     |  |  |        |     +---- te-label
       |  |     |  |  |        |        +---- (technology)?
       |  |     |  |  |        |        |  +--:(generic)
       |  |     |  |  |        |        |     +---- generic?     rt-
   types:generalized-label
       |  |     |  |  |        |        +---- direction?   te-label-
   direction
       |  |     |  |  |        +--:(srlg)
       |  |     |  |  |           +---- srlg
       |  |     |  |  |              +---- srlg?   uint32
       |  |     |  |  +---- explicit-route-include-objects
       |  |     |  |     +---- route-object-include-object* [index]
       |  |     |  |        +---- index            uint32
       |  |     |  |        +---- (type)?
       |  |     |  |           +--:(num-unnum-hop)
       |  |     |  |           |  +---- num-unnum-hop
       |  |     |  |           |     +---- node-id?      te-types:te-
   node-id
       |  |     |  |           |     +---- link-tp-id?   te-types:te-
   tp-id
       |  |     |  |           |     +---- hop-type?     te-hop-type
       |  |     |  |           |     +---- direction?    te-link-
   direction
       |  |     |  |           +--:(as-number)
       |  |     |  |           |  +---- as-number-hop
       |  |     |  |           |     +---- as-number?   binary
       |  |     |  |           |     +---- hop-type?    te-hop-type
       |  |     |  |           +--:(label)
       |  |     |  |              +---- label-hop
       |  |     |  |                 +---- te-label
       |  |     |  |                    +---- (technology)?
       |  |     |  |                    |  +--:(generic)
       |  |     |  |                    |     +---- generic?     rt-
   types:generalized-label
       |  |     |  |                    +---- direction?   te-label-
   direction
       |  |     |  +---- tiebreakers
       |  |     |     +---- tiebreaker* [tiebreaker-type]
       |  |     |        +---- tiebreaker-type    identityref
       |  |     +--:(objective-function) {path-optimization-objective-
   function}?
       |  |        +---- objective-function
       |  |           +---- objective-function-type?   identityref
       |  +---- requested-metrics* [metric-type]
       |  |  +---- metric-type    identityref
       |  +---- return-srlgs?             boolean
       |  +---- return-affinities?        boolean
       |  +---- path-in-segment!
       |  |  +---- label-restrictions
       |  |     +---- label-restriction* [index]
       |  |        +---- restriction?    enumeration
       |  |        +---- index           uint32
       |  |        +---- label-start
       |  |        |  +---- te-label
       |  |        |     +---- (technology)?
       |  |        |     |  +--:(generic)
       |  |        |     |     +---- generic?     rt-types:generalized-
   label
       |  |        |     +---- direction?   te-label-direction
       |  |        +---- label-end
       |  |        |  +---- te-label
       |  |        |     +---- (technology)?
       |  |        |     |  +--:(generic)
       |  |        |     |     +---- generic?     rt-types:generalized-
   label
       |  |        |     +---- direction?   te-label-direction
       |  |        +---- label-step
       |  |        |  +---- (technology)?
       |  |        |     +--:(generic)
       |  |        |        +---- generic?   int32
       |  |        +---- range-bitmap?   binary
       |  +---- path-out-segment!
       |     +---- label-restrictions
       |        +---- label-restriction* [index]
       |           +---- restriction?    enumeration
       |           +---- index           uint32
       |           +---- label-start
       |           |  +---- te-label
       |           |     +---- (technology)?
       |           |     |  +--:(generic)
```

```
      |           |     |     +---- generic?     rt-types:generalized-
   label
      |           |     +---- direction?   te-label-direction
      |           +---- label-end
      |           |  +---- te-label
      |           |     +---- (technology)?
      |           |     |  +--:(generic)
      |           |     |     +---- generic?     rt-types:generalized-
   label
      |           |     +---- direction?   te-label-direction
      |           +---- label-step
      |           |  +---- (technology)?
      |           |     +--:(generic)
      |           |        +---- generic?   int32
      |           +---- range-bitmap?   binary
      +---- synchronization* [synchronization-id]
         +---- synchronization-id    uint32
         +---- svec
         |  +---- relaxable?           boolean
         |  +---- disjointness?        te-types:te-path-disjointness
         |  +---- request-id-number*   uint32
         +---- svec-constraints
         |  +---- path-metric-bound* [metric-type]
         |     +---- metric-type    identityref
         |     +---- upper-bound?   uint64
         +---- path-srlgs-values
         |  +---- usage?    identityref
         |  +---- values*   srlg
         +---- path-srlgs-names
         |  +---- path-srlgs-name* [usage]
         |     +---- usage        identityref
         |     +---- srlg-name* [name]
         |        +---- name    string
         +---- exclude-objects
         |  +---- excludes* [index]
         |     +---- index            uint32
         |     +---- (type)?
         |        +--:(num-unnum-hop)
         |        |  +---- num-unnum-hop
```

```
          |        |     +---- node-id?      te-types:te-node-id
          |        |     +---- link-tp-id?   te-types:te-tp-id
          |        |     +---- hop-type?     te-hop-type
          |        |     +---- direction?    te-link-direction
          |        +--:(as-number)
          |        |  +---- as-number-hop
          |        |     +---- as-number?   binary
          |        |     +---- hop-type?    te-hop-type
          |        +--:(label)
          |           +---- label-hop
          |              +---- te-label
          |                 +---- (technology)?
          |                 |  +--:(generic)
          |                 |     +---- generic?     rt-
   types:generalized-label
          |                 +---- direction?   te-label-direction
          +---- optimizations
             +---- (algorithm)?
                +--:(metric)
                |  +---- optimization-metric* [metric-type]
                |     +---- metric-type    identityref
                |     +---- weight?        uint8
                +--:(objective-function)
                   +---- objective-function
                      +---- objective-function-type?   identityref
     augment /te:tunnels-rpc/te:output/te:result:
       +--ro response* [response-id]
          +--ro response-id      uint32
          +--ro (response-type)?
             +--:(no-path-case)
             |  +--ro no-path!
             +--:(path-case)
                +--ro computed-path
                   +--ro path-id?           yang-types:uuid
                   +--ro path-properties
                      +--ro path-metric* [metric-type]
                      |  +--ro metric-type           identityref
                      |  +--ro accumulative-value?   uint64
                      +--ro path-affinities-values
                      |  +--ro path-affinities-value* [usage]
                      |     +--ro usage    identityref
                      |     +--ro value?   admin-groups
                      +--ro path-affinity-names
                      |  +--ro path-affinity-name* [usage]
                      |     +--ro usage            identityref
                      |     +--ro affinity-name* [name]
                      |        +--ro name    string
                      +--ro path-srlgs-values
                      |  +--ro usage?    identityref
                      |  +--ro values*   srlg
                      +--ro path-srlgs-names
                      |  +--ro path-srlgs-name* [usage]
                      |     +--ro usage        identityref
                      |     +--ro srlg-name* [name]
                      |        +--ro name    string
                      +--ro path-route-objects
                         +--ro path-route-object* [index]
                            +--ro index            uint32
                            +--ro (type)?
                               +--:(num-unnum-hop)
                               |  +--ro num-unnum-hop
                               |     +--ro node-id?      te-types:te-
   node-id

                               |     +--ro link-tp-id?   te-types:te-
   tp-id

                               |     +--ro hop-type?     te-hop-type
                               |     +--ro direction?    te-link-
   direction
                               +--:(as-number)
                               |  +--ro as-number-hop
                               |     +--ro as-number?   binary
                               |     +--ro hop-type?    te-hop-type
                               +--:(label)
                                  +--ro label-hop
                                     +--ro te-label
                                        +--ro (technology)?
                                        |  +--:(generic)
                                        |     +--ro generic?     rt-
   types:generalized-label
                                        +--ro direction?   te-label-
   direction
```

Figure 9 - TE path computation YANG tree

6.2. YANG Module

```
   <CODE BEGINS>file "ietf-te-path-computation@2018-10-22.yang"
   module ietf-te-path-computation {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-te-path-computation";
     // replace with IANA namespace when assigned

     prefix "tepc";

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

     import ietf-yang-types {
       prefix "yang-types";
     }

     import ietf-te {
       prefix "te";
     }

     import ietf-te-types {
       prefix "te-types";
     }

     organization
       "Traffic Engineering Architecture and Signaling (TEAS)
        Working Group";

     contact
       "WG Web:   <http://tools.ietf.org/wg/teas/>
        WG List:  <mailto:teas@ietf.org>
        WG Chair: Lou Berger
                  <mailto:lberger@labn.net>

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

      ";

     description "YANG model for stateless TE path computation";

     revision "2018-10-22" {
       description
         "Initial revision";
       reference
         "draft-ietf-teas-yang-path-computation";
     }

     /*
      * Features
      */

     feature stateless-path-computation {
       description
         "This feature indicates that the system supports
          stateless path computation.";
     }


     /*
      * Groupings
      */

     grouping path-info {
       leaf path-id {
         type yang-types:uuid;
         config false;
         description "path-id ref.";
       }
       uses te-types:generic-path-properties;
       description "Path computation output information";
     }

     grouping requested-info {
       description
         "This grouping defines the information (e.g., metrics)
          which must be returned in the response";
       list requested-metrics {
         key 'metric-type';
         description
           "The list of the requested metrics
            The metrics listed here must be returned in the response.
            Returning other metrics in the response is optional.";
         leaf metric-type {
           type identityref {
             base te-types:path-metric-type;
           }
           description
             "The metric that must be returned in the response";
         }
       }
       leaf return-srlgs {
         type boolean;
         default false;
         description
           "If true, path srlgs must be returned in the response.
            If false, returning path srlgs in the response optional.";
       }
       leaf return-affinities {
         type boolean;
         default false;
         description
         "If true, path affinities must be returned in the response.
          If false, returning path affinities in the response is
          optional.";
       }
     }
     identity svec-metric-type {
       description
         "Base identity for svec metric type";
     }

     identity svec-metric-cumul-te {
       base svec-metric-type;
       description
         "TE cumulative path metric";
     }

     identity svec-metric-cumul-igp {
       base svec-metric-type;
       description
         "IGP cumulative path metric";
     }

     identity svec-metric-cumul-hop {
       base svec-metric-type;
       description
         "Hop cumulative path metric";
     }

     identity svec-metric-aggregate-bandwidth-consumption {
       base svec-metric-type;
       description
         "Cumulative bandwith consumption of the set of synchronized
   paths";
     }

     identity svec-metric-load-of-the-most-loaded-link {
       base svec-metric-type;
       description
         "Load of the most loaded link";
     }

     grouping svec-metrics-bounds_config {
       description "TE path metric bounds grouping for computing a set
   of
```

```
         synchronized requests";
       leaf metric-type {
         type identityref {
           base svec-metric-type;
         }
         description "TE path metric type usable for computing a set of
            synchronized requests";
       }
       leaf upper-bound {
         type uint64;
         description "Upper bound on end-to-end svec path metric";
       }
     }

     grouping svec-metrics-optimization_config {
       description "TE path metric bounds grouping for computing a set
   of
         synchronized requests";
       leaf metric-type {
         type identityref {
           base svec-metric-type;
         }
         description "TE path metric type usable for computing a set of
            synchronized requests";
       }
       leaf weight {
         type uint8;
         description "Metric normalization weight";
       }
     }

     grouping svec-exclude {
       description "List of resources to be excluded by all the paths
         in the SVEC";
       container exclude-objects {
         description "resources to be excluded";
         list excludes {
           key index;
           description
```

```
             "List of explicit route objects to always exclude
              from synchronized path computation";
           leaf index {
             type uint32;
             description "XRO subobject index";
           }
           uses te-types:explicit-route-hop;
         }
       }
     }

     grouping synchronization-constraints {
       description "Global constraints applicable to synchronized
         path computation";
       container svec-constraints {
         description "global svec constraints";
         list path-metric-bound {
           key metric-type;
           description "list of bound metrics";
           uses svec-metrics-bounds_config;
         }
       }
       uses te-types:generic-path-srlgs;
       uses svec-exclude;
     }

     grouping synchronization-optimization {
         description "Synchronized request optimization";
       container optimizations {
         description
           "The objective function container that includes
            attributes to impose when computing a synchronized set of
   paths";

         choice algorithm {
           description "Optimizations algorithm.";
           case metric {
             list optimization-metric {
               key "metric-type";
               description "svec path metric type";
               uses svec-metrics-optimization_config;
             }
           }
           case objective-function {
             container objective-function {
               description
                 "The objective function container that includes
                  attributes to impose when computing a TE path";
               uses te-types:path-objective-function_config;
             }
           }
         }
       }
     }

     grouping synchronization-info {
       description "Information for sync";
       list synchronization {
         key "synchronization-id";
         description "sync list";
         leaf synchronization-id {
           type uint32;
           description "index";
         }
         container svec {
           description
            "Synchronization VECtor";
           leaf relaxable {
             type boolean;
             default true;
             description
              "If this leaf is true, path computation process is free
   to ignore svec content.
               otherwise it must take into account this svec.";
           }
           uses te-types:generic-path-disjointness;
           leaf-list request-id-number {
             type uint32;
             description "This list reports the set of M path
   computation
               requests that must be synchronized.";
           }
         }
         uses synchronization-constraints;
         uses synchronization-optimization;
       }
     }

     grouping no-path-info {
       description "no-path-info";
       container no-path {
         presence "Response without path information, due to failure
           performing the path computation";
         description "if path computation cannot identify a path,
           rpc returns no path.";
       }
     }

     /*
      * These groupings should be removed when defined in te-types
      */

     grouping encoding-and-switching-type {
       description
         "Common grouping to define the LSP encoding and switching
   types";

       leaf encoding {
         type identityref {
           base te-types:lsp-encoding-types;
         }
         description "LSP encoding type";
         reference "RFC3945";
       }
       leaf switching-type {
         type identityref {
           base te-types:switching-capabilities;
         }
         description "LSP switching type";
         reference "RFC3945";
       }
     }

     grouping end-points {
       description
         "Common grouping to define the TE tunnel end-points";

       leaf source {
         type inet:ip-address;
         description "TE tunnel source address.";
       }
       leaf destination {
         type inet:ip-address;
         description "P2P tunnel destination address";
       }
       leaf src-tp-id {
         type binary;
         description "TE tunnel source termination point identifier.";
       }
       leaf dst-tp-id {
         type binary;
         description "TE tunnel destination termination point
   identifier.";
       }
       leaf bidirectional {
         type boolean;
         default 'false';
         description "TE tunnel bidirectional";
       }
     }

     /**
      * AUGMENTS TO TE RPC
      */

     augment "/te:tunnels-rpc/te:input/te:tunnel-info" {
       description "statelessComputeP2PPath input";
       list path-request {
         key "request-id";
         description "request-list";
         leaf request-id {
           type uint32;
           mandatory true;
           description "Each path computation request is uniquely
   identified by the request-id-number.
             It must be present also in rpcs.";
         }
         uses te-types:te-topology-identifier;
         uses end-points;
         uses encoding-and-switching-type;
         uses te-types:path-route-objects;
         uses te-types:generic-path-constraints;
         uses te-types:generic-path-optimization;
         uses requested-info;
         uses te:path-access-segment-info;
       }
       uses synchronization-info;
     }

     augment "/te:tunnels-rpc/te:output/te:result" {
       description "statelessComputeP2PPath output";
       list response {
         key response-id;
         config false;
         description "response";
         leaf response-id {
           type uint32;
           description
             "The list key that has to reuse request-id-number.";
         }
         choice response-type {
           config false;
           description "response-type";
           case no-path-case {
             uses no-path-info;
```

```
          }
          case path-case {
            container computed-path {
              uses path-info;
              description "Path computation service.";
            }
          }
        }
      }
    }
  }

  <CODE ENDS>
```

Figure 10 - TE path computation YANG module

7. Security Considerations

This document describes use cases of requesting Path Computation using YANG models, which could be used at the ABNO Control Interface [RFC7491] and/or between controllers in ACTN [ACTN-frame]. As such, it does not introduce any new security considerations compared to the ones related to YANG specification, ABNO specification and ACTN Framework defined in [RFC7950], [RFC7491] and [ACTN-frame].

The YANG module defined in this draft is designed to be accessed via the NETCONF protocol [RFC6241] or RESTCONF protocol [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 [RFC5246].

This document also defines common data types using the YANG data modeling language. The definitions themselves have no security impact on the Internet, but the usage of these definitions in concrete YANG modules might have. The security considerations spelled out in the YANG specification [RF7950] apply for this document as well.

The NETCONF access control model [RFC6536] 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.

Note - The security analysis of each leaf is for further study.

[Editor's note:] Complete the security analysis. Check if we cannot just reference the te draft since the rpc exposes the same information that can be exposed by the te-tunnel model.

8. IANA Considerations

This document registers the following URIs 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-te-path-computation
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC7950].

name: ietf-te-path-computation
namespace: urn:ietf:params:xml:ns:yang:ietf-te-path-computation
prefix: tepc

9. References

9.1. Normative References

[RFC5541] Le Roux, JL. et al., " Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC7491] Farrel, A., King, D., "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, March 2015.

[RFC7926] Farrel, A. et al., "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", RFC 7926, July 2016.

[RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.

[TE-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-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.

[ACTN-Frame] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of Traffic Engineered Networks" draft-ietf-actn-framework, work in progress.

[ACTN-Info] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., "Information Model for Abstraction and Control of Transport Networks", draft-ietf-teas-actn-info, work in progress.

9.1. Informative References

[RFC4655] Farrel, A. et al., "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC7139] Zhang, F. et al., "GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks", RFC 7139, March 2014.

[RFC7446] Lee, Y. et al., "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, February 2015.

[OTN-TOPO] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang, work in progress.

[PCEP-Service-Aware] Dhody, D. et al., "Extensions to the Path Computation Element Communication Protocol (PCEP) to compute service aware Label Switched Path (LSP)", draft-ietf-pce-pcep-service-aware, work in progress.

[ITU-T G.709-2016] ITU-T Recommendation G.709 (06/16), "Interface for the optical transport network", June 2016.

10. Acknowledgments

The authors would like to thank Igor Bryskin and Xian Zhang for participating in discussions and providing valuable insights.

The authors would like to thank the authors of the TE Tunnel YANG model [TE-TUNNEL], in particular Igor Bryskin, Vishnu Pavan Beeram, Tarek Saad and Xufeng Liu, for their inputs to the discussions and support in having consistency between the Path Computation and TE Tunnel YANG models.

This document was prepared using 2-Word-v2.0.template.dot.

Appendix A. Examples of dimensioning the "detailed connectivity matrix"

In the following table, a list of the possible constraints, associated with their potential cardinality, is reported.

The maximum number of potential connections to be computed and reported is, in first approximation, the multiplication of all of them.

```
Constraint  Cardinality
----------  ----------------------------------------------------------
```

End points N(N-1)/2 if connections are bidirectional (OTN and WDM), N(N-1) for unidirectional connections.

Bandwidth In WDM networks, bandwidth values are expressed in GHz.

On fixed-grid WDM networks, the central frequencies are on a 50GHz grid and the channel width of the transmitters are typically 50GHz such that each central frequency can be used, i.e., adjacent channels can be placed next to each other in terms of central frequencies.

On flex-grid WDM networks, the central frequencies are on a 6.25GHz grid and the channel width of the transmitters can be multiples of 12.5GHz.

For fixed-grid WDM networks typically there is only one possible bandwidth value (i.e., 50GHz) while for flex-grid WDM networks typically there are 4 possible bandwidth values (e.g., 37.5GHz, 50GHz, 62.5GHz, 75GHz).

In OTN (ODU) networks, bandwidth values are expressed as pairs of ODU type and, in case of ODUflex, ODU rate in bytes/sec as described in section 5 of [RFC7139].

For "fixed" ODUk types, 6 possible bandwidth values are possible (i.e., ODU0, ODU1, ODU2, ODU2e, ODU3, ODU4).

For ODUflex(GFP), up to 80 different bandwidth values can be specified, as defined in Table 7-8 of [ITU-T G.709-2016].

For other ODUflex types, like ODUflex(CBR), the number of possible bandwidth values depends on the rates of the clients that could be mapped over these ODUflex types, as shown in Table 7.2 of [ITU-T G.709-2016], which in theory could be a countinuum of values. However, since different ODUflex bandwidths that use the same number of TSs on each link along the path are equivalent for path computation purposes, up to 120 different bandwidth ranges can be specified.

Ideas to reduce the number of ODUflex bandwidth values in the detailed connectivity matrix, to less than 100, are for further study.

[Editor’s note:] It is possible to follow a similar approach as the one proposed for IP networks and report fewer optimal paths for ODUflex range of rates which could be one or more consecutive ranges of the theoretical 120 bandwidth ranges.

Another simplification could be not to report optimal path for bandwidth ranges for which no client mapping is defined.

More research about this alternative is needed.

Bandwidth specification for ODUCn is currently for further study but it is expected that other bandwidth values can be specified as integer multiples of 100Gb/s.

In IP we have bandwidth values in bytes/sec. In principle, this is a countinuum of values, but in practice we can identify a set of bandwidth ranges, where any bandwidth value inside the same range produces the same path.
The number of such ranges is the cardinality, which depends on the topology, available bandwidth and status of the network. Simulations (Note: reference paper submitted for publication) show that values for medium size topologies (around 50-150 nodes) are in the range 4-7 (5 on average) for each end points couple.

[Editor’s note (Francesco):] Inform us as soon as the paper is published to add the reference to this document.

Metrics   IGP, TE and hop number are the basic objective metrics defined so far. There are also the 2 objective functions defined in [RFC5541]: Minimum Load Path (MLP) and Maximum Residual Bandwidth Path (MBP). Assuming that one only metric or objective function can be optimized at once, the total cardinality here is 5.

With [PCEP-Service-Aware], a number of additional metrics are defined, including Path Delay metric, Path Delay Variation metric and Path Loss metric, both for point-to-point and point-to-multipoint paths. This increases the cardinality to 8.

Bounds Each metric can be associated with a bound in order to find a path having a total value of that metric lower than the given bound. This has a potentially very high cardinality (as any value for the bound is allowed). In practice there is a maximum value of the bound (the one with the maximum value of the associated metric) which results always in the same path, and a range approach like for bandwidth in IP should produce also in this case the cardinality. Assuming to have a cardinality similar to the one of the bandwidth (let say 5 on average) we should have 6 (IGP, TE, hop, path delay, path delay variation and path loss; we don't consider here the two objective functions of [RFC5541] as they are conceived only for optimization)*5 = 30 cardinality.

Technology
constraints For further study

[Editor's note:] Discuss further what are the impacts of these technology constraints (e.g., Modulation format, FEC, ...) to path computation.

Priority We have 8 values for setup priority, which is used in path computation to route a path using free resources and, where no free resources are available, resources used by LSPs having a lower holding priority.

Local prot It's possible to ask for a local protected service, where all the links used by the path are protected with fast reroute (this is only for IP networks, but line protection schemas are available on the other technologies as well). This adds an alternative path computation, so the cardinality of this constraint is 2.

Administrative Colors

Administrative colors (aka affinities) are typically assigned to links but when topology abstraction is used affinity information can also appear in the detailed connectivity matrix.

There are 32 bits available for the affinities. Links can be tagged with any combination of these bits, and path computation can be constrained to include or exclude any or all of them. The relevant cardinality is 3 (include-any, exclude-any, include-all) times 2^32 possible values. However, the number of possible values used in real networks is quite small.

Included Resources

A path computation request can be associated to an ordered set of network resources (links, nodes) to be included along the computed path. This constraint would have a huge cardinality as in principle any combination of network resources is possible. However, as far as the Orchestrator doesn't know details about the internal topology of the domain, it shouldn't include this type of constraint at all (see more details below).

Excluded Resources

A path computation request can be associated to a set of network resources (links, nodes, SRLGs) to be excluded from the computed path. Like for included resources, this constraint has a potentially very high cardinality, but, once again, it can't be actually used by the Orchestrator, if it's not aware of the domain topology (see more details below).

As discussed above, the Orchestrator can specify include or exclude resources depending on the abstract topology information that the domain controller exposes:

- In case the domain controller exposes the entire domain as a single abstract TE node with his own external terminations and detailed connectivity matrix (whose size we are estimating), no other topological details are available, therefore the size of the detailed connectivity matrix only depends on the combination of the constraints that the Orchestrator can use in a path computation request to the domain controller. These constraints cannot refer to any details of the internal topology of the domain, as those details are not known to the Orchestrator and so they do not impact size of the detailed connectivity matrix exported.

- Instead in case the domain controller exposes a topology including more than one abstract TE nodes and TE links, and their attributes (e.g. SRLGs, affinities for the links), the Orchestrator knows these details and therefore could compute a path across the domain referring to them in the constraints. The detailed connectivity matrixes, whose size need to be estimated here, are the ones relevant to the abstract TE nodes exported to the Orchestrator. These detailed connectivity matrixes and therefore theirs sizes, while cannot depend on the other abstract TE nodes and TE links, which are external to the given abstract node, could depend to SRLGs (and other attributes, like affinities) which could be present also in the portion of the topology represented by the abstract nodes, and therefore contribute to the size of the related detailed connectivity matrix.

We also don’t consider here the possibility to ask for more than one path in diversity or for point-to-multi-point paths, which are for further study.

Considering for example an IP domain without considering SRLG and affinities, we have an estimated number of paths depending on these estimated cardinalities:

Endpoints = N*(N-1), Bandwidth = 5, Metrics = 6, Bounds = 20,
Priority = 8, Local prot = 2

The number of paths to be pre-computed by each IP domain is therefore 24960 * N(N-1) where N is the number of domain access points.

This means that with just 4 access points we have nearly 300000 paths to compute, advertise and maintain (if a change happens in the domain, due to a fault, or just the deployment of new traffic, a substantial number of paths need to be recomputed and the relevant changes advertised to the upper controller).

This seems quite challenging. In fact, if we assume a mean length of 1K for the json describing a path (a quite conservative estimate), reporting 300000 paths means transferring and then parsing more than 300 Mbytes for each domain. If we assume that 20% (to be checked) of this paths change when a new deployment of traffic occurs, we have 60 Mbytes of transfer for each domain traversed by a new end-to-end path. If a network has, let say, 20 domains (we want to estimate the load for a non-trivial domain setup) in the beginning a total initial transfer of 6Gigs is needed, and eventually, assuming 4-5 domains are involved in mean during a path deployment we could have 240-300 Mbytes of changes advertised to the higher order controller.

Further bare-bone solutions can be investigated, removing some more options, if this is considered not acceptable; in conclusion, it seems that an approach based only on the information provided by the detailed connectivity matrix is hardly feasible, and could be applicable only to small networks with a limited meshing degree between domains and renouncing to a number of path computation features.

[Editor's note:] Evaluate whether to describe the bare-bone solution as another way to jointly use detailed connectivity matrix and path computation in a complimentary way. The paragraph above could be updated as follow:

We could still try and provide a bare-bone solution removing some more options:

- No local protection
- Max one bound in path computation. When asking for a path with a bound, the request towards the domain is done using the bound metric as the objective metric unregarding the original objective metric. While this provides possibly a non-optimal solution, that solution guarantees the satisfaction of the bound, if possible at all.
- Reduce the used metric to just IGP and delay, ignoring hops and TE metric (or using TE for delay).

In this case we have just 5*2*8=80*N*(N-1) paths. With 4 access points we have 12*80 = 960 paths to be computed and maintained. This is still heavy but probably feasible if the number of the domain access points is limited.

In conclusion, the connectivity matrix approach seems feasible only with a bare-bone approach with limited meshing among domains and without using most of the available path computation capabilities, unless the bare-one approach is complemented with path computation as described in section 3.2.3.

<<Check also text in section 3.3>>

Contributors

Dieter Beller
Nokia
Email: dieter.beller@nokia.com

Gianmarco Bruno
Ericsson
Email: gianmarco.bruno@ericsson.com

Francesco Lazzeri
Ericsson
Email: francesco.lazzeri@ericsson.com

Young Lee
Huawei
Email: leeyoung@huawei.com

Carlo Perocchio
Ericsson
Email: carlo.perocchio@ericsson.com

Authors' Addresses

Italo Busi (Editor)
Huawei
Email: italo.busi@huawei.com

Sergio Belotti (Editor)
Nokia
Email: sergio.belotti@nokia.com

Victor Lopez
Telefonica
Email: victor.lopezalvarez@telefonica.com

Oscar Gonzalez de Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com

Anurag Sharma
Google
Email: ansha@google.com

Yan Shi
China Unicom
Email: shiyan49@chinaunicom.cn

Ricard Vilalta
CTTC
Email: ricard.vilalta@cttc.es

Karthik Sethuraman
NEC
Email: karthik.sethuraman@necam.com

Michael Scharf
Nokia
Email: michael.scharf@gmail.com

Daniele Ceccarelli
Ericsson
Email: daniele.ceccarelli@ericsson.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 23, 2019

T. Saad
R. Gandhi
Cisco Systems Inc
X. Liu
Volta Networks
V. Beeram
Juniper Networks
H. Shah
Ciena
I. Bryskin
Huawei Technologies

October 20, 2018

A YANG Data Model for Traffic Engineering Tunnels and Interfaces
draft-ietf-teas-yang-te-17

Abstract

This document defines a YANG data model for the configuration and management of Traffic Engineering (TE) interfaces, tunnels and Label Switched Paths (LSPs). The model is divided into YANG modules that classify data into generic, device-specific, technology agnostic, and technology-specific elements.

This model covers data for configuration, operational state, remote procedural calls, and event notifications.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

YANG [RFC6020] and [RFC7950] is a data modeling language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG has proved relevant beyond its initial confines, as bindings to other interfaces (e.g. RESTCONF [RFC8040]) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document describes YANG data model for TE Tunnels, Label Switched Paths (LSPs) and TE interfaces and covers data applicable to generic or device-independent, device-specific, and Multiprotocol Label Switching (MPLS) technology specific.

The document describes a high-level relationship between the modules defined in this document, as well as other external protocol YANG modules. The TE generic YANG data model does not include any data specific to a signaling protocol. It is expected other data plane technology model(s) will augment the TE generic YANG data model.

Also, it is expected other YANG module(s) that model TE signaling protocols, such as RSVP-TE ([RFC3209], [RFC3473]), or Segment-Routing TE (SR-TE) will augment the TE generic YANG module.

1.1. Terminology

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

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

1.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
te	ietf-te	this document
te-dev	ietf-te-device	this document
te-types	ietf-te-types	[I-D.ietf-teas-yang-te-types]
te-mpls-types	ietf-te-mpls-types	[I-D.ietf-teas-yang-te-types]

Table 1: Prefixes and corresponding YANG modules

1.3. TE Technology Models

This document describes the TE generic YANG data model that is independent of any dataplane technology. One of the design objectives is to allow specific data plane technologies models to reuse the TE generic data model and possibly augment it with technology specific data. There are multiple options that were considered:

- Elements of the TE generic YANG data model, including TE tunnels, LSPs, and interfaces can be augmented with leaf(s) to identify the specific technology layer. This approach implies a single list of elements (e.g. TE tunnel(s)) in the model can carry elements of different technology layers.

- An instance of the TE generic YANG model can be mounted in the YANG tree once for each TE technology layer(s). This approach provides separation of elements belonging to different technology layers into separate lists per layer in the data model.

The model defined in this document leverages the first apprach by relying on the LSP encoding type to identify the specific technology associated with a specific TE interface, tunnel or LSP. For example, for an MPLS TE LSP, the LSP encoding type is assumed to be of "te-types:lsp-encoding-packet".

Finally, the TE generic YANG data model does not include any signaling protocol data. It is expected TE signaling protocol module(s) will be defined in other document(s) to cover protocols such as RSVP-TE ([RFC3209], [RFC3473]), and Segment-Routing TE (SR-TE) model and that augment the TE generic YANG data model.

1.4. State Data Organization

The Network Management Datastore Architecture (NMDA) [RFC8342] addresses modeling state data for ephemeral objects. This draft adopts the NMDA proposal for configuration and state data representation as per IETF guidelines for new IETF YANG models.

2. Model Overview

The data model(s) defined in this document cover core TE features that are commonly supported across different vendor implementations. The support of extended or vendor specific TE feature(s) is expected to be in augmentations to the base models defined in this document.

2.1. Module(s) Relationship

The TE generic YANG data model defined in "ietf-te.yang" covers the building blocks that are device independent and agnostic of any specific technology or control plane instances. The TE device model defined in "ietf-te-device.yang" augments the TE generic YANG data model and covers data that is specific to a device - for example, attributes of TE interfaces, or TE timers that are local to a TE node.

The TE data model for specific instances of data plane technology exist in a separate YANG module(s) that augment the TE generic YANG data model. For example, the MPLS-TE module "ietf-te-mpls.yang" defined in another document can augment the TE generic model as shown in Figure 1.

The TE data model for specific instances of signaling protocol are outside the scope of this document and defined in separate documents. For example, the RSVP-TE [RFC3209] YANG model augmentation of the TE model is covered in [I-D.ietf-teas-yang-rsvp], and other signaling protocol model(s) (e.g. for Segment-Routing TE) are expected to also augment the TE generic YANG data model.

The TE generic YANG module "ietf-te" imports the following modules:

- ietf-yang-types and ietf-inet-types defined in [RFC6991]

- ietf-te-types defined in [I-D.ietf-teas-yang-te-types]

The TE device YANG module "ietf-te-device" imports the following module(s): - ietf-yang-types and ietf-inet-types defined in [RFC6991] - ietf-routing-types defined in [RFC8294] - ietf-te-types defined in [I-D.ietf-teas-yang-te-types] - ietf-te defined in this document

```
     TE generic     +---------+         o: augment
     module         | ietf-te |o-------------+
                    +---------+               \
                           o                   \
                           |\                   \
                           | \                   \
                           |  +----------------+  \
                           |  | ietf-te-device | TE device module
                           |  +----------------+    \
                           |       o        o        \
                           |     /           \        \
                           |   /              \        \
                    +--------------+           +---------------+
     RSVP-TE module | ietf-rsvp-te |o .        | ietf-te-mpls^ |
                    +--------------+   \       +---------------+
                       |                \
                       |                 \
                       |                  \
                       |                   \
                       |                    \
                       o                 +-------------------+
                    +-----------+        | ietf-rsvp-otn-te^ |
     RSVP module    | ietf-rsvp |        +-------------------+
                    +-----------+           RSVP-TE with OTN
                                            extensions

                           ^ shown for illustration
                             (not in this document)
```

Figure 1: Relationship of TE module(s) with other signaling protocol modules

```
     +---------+
     | ietf-te |       ^: import
     +---------+
       ^  ^  ^
       |  |  +--------------------------------------+
       |  +----------------------+                  |
       |                         |                  |
     +----------------+   +---------------------+ +--------------------+
     | ietf-te-types* |   | ietf-te-mpls-types* | | ietf-te-otn-types* |
     +----------------+   +---------------------+ +--------------------+

     *: not in this document, shown for illustration only
```

Figure 2: Relationship between generic and technology specific TE types modules

2.2. Design Considerations

The following considerations are taken into account with respect data organization:

- o reusable TE data types that are data plane independent are grouped in the TE generic types module "ietf-te-types.yang" defined in [I-D.ietf-teas-yang-te-types]

- o reusable TE data types that are data plane specific (e.g. MPLS [RFC3473]) are defined in a data plane type module, e.g. "ietf-te-mpls-types.yang" as defined in [I-D.ietf-teas-yang-te-types]. Other data plane technology types are expected to be defined in separate module(s) as shown in Figure 2

- o The TE generic YANG data model "ietf-te" contains device independent data and can be used to model data off a device (e.g. on a controller). The device-specific TE data is defined in a separate module "ietf-te-device" as shown in Figure 1.

- o In general, minimal elements in the model are designated as "mandatory" to allow freedom to vendors to adapt the data model to their specific product implementation.

- o This model declares a number of TE functions as features that can be optionally supported.

2.3. Model Tree Diagram

Figure 3 shows the tree diagram of the TE YANG model defined in modules: ietf-te.yang, and ietf-te-device.yang.

```
module: ietf-te
  +--rw te!
     +--rw globals
     |  +--rw named-admin-groups
     |  |  +--rw named-admin-group* [name]
   te-types:extended-admin-groups,te-types:named-extended-admin-groups}?
     |  |     +--rw name            string
     |  |     +--rw bit-position?   uint32
     |  +--rw named-srlgs
     |  |  +--rw named-srlg* [name] {te-types:named-srlg-groups}?
     |  |     +--rw name     string
     |  |     +--rw group?   te-types:srlg
     |  |     +--rw cost?    uint32
     |  +--rw named-path-constraints
     |  |  +--rw named-path-constraint* [name]
     {te-types:named-path-constraints}?
     |  |     +--rw name                        string
     |  |     +--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-values
     |  |     |  +--rw usage?    identityref
     |  |     |  +--rw values*   srlg
     |  |     +--rw path-srlgs-names
     |  |     |  +--rw path-srlgs-name* [usage]
     |  |     |     +--rw usage        identityref
     |  |     |     +--rw srlg-name* [name]
     |  |     |        +--rw name    string
     |  |     +--rw disjointness?
     te-types:te-path-disjointness
     |  |     +--rw explicit-route-objects
     |  |     |  +--rw route-object-exclude-always* [index]
     |  |     |  |  +--rw index                  uint32
     |  |     |  |  +--rw (type)?
     |  |     |  |     +--:(num-unnum-hop)
     |  |     |  |     |  +--rw num-unnum-hop
     |  |     |  |     |     +--rw node-id?      te-types:te-node-id
     |  |     |  |     |     +--rw link-tp-id?   te-types:te-tp-id
     |  |     |  |     |     +--rw hop-type?     te-hop-type
     |  |     |  |     |     +--rw direction?    te-link-direction
     |  |     |  |     +--:(as-number)
     |  |     |  |     |  +--rw as-number-hop
     |  |     |  |     |     +--rw as-number?   binary
     |  |     |  |     |     +--rw hop-type?    te-hop-type
     |  |     |  |     +--:(label)
     |  |     |  |        +--rw label-hop
     |  |     |  |           +--rw te-label
     |  |     |  |              +--rw (technology)?
     |  |     |  |              |  +--:(generic)
     |  |     |  |              |     +--rw generic?
     rt-types:generalized-label
     |  |     |  |              +--rw direction?
     te-label-direction
     |  |     |  +--rw route-object-include-exclude* [index]
     |  |     |     +--rw explicit-route-usage?   identityref
     |  |     |     +--rw index                   uint32
     |  |     |     +--rw (type)?
     |  |     |        +--:(num-unnum-hop)
     |  |     |        |  +--rw num-unnum-hop
     |  |     |        |     +--rw node-id?      te-types:te-node-id
     |  |     |        |     +--rw link-tp-id?   te-types:te-tp-id
     |  |     |        |     +--rw hop-type?     te-hop-type
     |  |     |        |     +--rw direction?    te-link-direction
     |  |     |        +--:(as-number)
     |  |     |        |  +--rw as-number-hop
     |  |     |        |     +--rw as-number?   binary
     |  |     |        |     +--rw hop-type?    te-hop-type
     |  |     |        +--:(label)
     |  |     |        |  +--rw label-hop
     |  |     |        |     +--rw te-label
     |  |     |        |        +--rw (technology)?
     |  |     |        |        |  +--:(generic)
     |  |     |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |     |        |        +--rw direction?
     te-label-direction
     |  |     |        +--:(srlg)
     |  |     |           +--rw srlg
     |  |     |              +--rw srlg?   uint32
     |  |     +--rw shared-resources-tunnels
     |  |     |  +--rw lsp-shared-resources-tunnel*   tunnel-ref
     |  |     +--rw path-in-segment!
     |  |     |  +--rw label-restrictions
     |  |     |     +--rw label-restriction* [index]
     |  |     |        +--rw restriction?    enumeration
     |  |     |        +--rw index           uint32
     |  |     |        +--rw label-start
     |  |     |        |  +--rw te-label
     |  |     |        |     +--rw (technology)?
     |  |     |        |     |  +--:(generic)
     |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |     |        |     +--rw direction?       te-label-direction
     |  |     |        +--rw label-end
     |  |     |        |  +--rw te-label
     |  |     |        |     +--rw (technology)?
     |  |     |        |     |  +--:(generic)
     |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |     |        |     +--rw direction?       te-label-direction
     |  |     |        +--rw label-step
     |  |     |        |  +--rw (technology)?
     |  |     |        |     +--:(generic)
     |  |     |        |        +--rw generic?   int32
     |  |     |        +--rw range-bitmap?   binary
     |  |     +--rw path-out-segment!
     |  |     |  +--rw label-restrictions
     |  |     |     +--rw label-restriction* [index]
     |  |     |        +--rw restriction?    enumeration
     |  |     |        +--rw index           uint32
     |  |     |        +--rw label-start
     |  |     |        |  +--rw te-label
     |  |     |        |     +--rw (technology)?
     |  |     |        |     |  +--:(generic)
     |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |     |        |     +--rw direction?       te-label-direction
     |  |     |        +--rw label-end
     |  |     |        |  +--rw te-label
     |  |     |        |     +--rw (technology)?
     |  |     |        |     |  +--:(generic)
     |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |     |        |     +--rw direction?       te-label-direction
     |  |     |        +--rw label-step
     |  |     |        |  +--rw (technology)?
     |  |     |        |     +--:(generic)
     |  |     |        |        +--rw generic?   int32
     |  |     |        +--rw range-bitmap?   binary
     |  |     +--ro state
     |  |     |  +--ro bandwidth-generic_state?   te-types:te-bandwidth
     |  |     |  +--ro disjointness_state?
     te-types:te-path-disjointness
     |  |     +--rw te-mpls:bandwidth
     |  |        +--rw te-mpls:specification-type?
     te-mpls-types:te-bandwidth-requested-type
     |  |        +--rw te-mpls:set-bandwidth?
     te-mpls-types:bandwidth-kbps
     |  |        +--rw te-mpls:class-type?
     te-types:te-ds-class
     |  |        +--ro te-mpls:state
     |  |           +--ro te-mpls:signaled-bandwidth?
     te-mpls-types:bandwidth-kbps
     |  +--rw te-dev:lsp-install-interval?        uint32
     |  +--rw te-dev:lsp-cleanup-interval?        uint32
     |  +--rw te-dev:lsp-invalidation-interval?   uint32
     +--rw tunnels
     |  +--rw tunnel* [name]
     |  |  +--rw name                                string
     |  |  +--rw identifier?                         uint16
     |  |  +--rw description?                        string
     |  |  +--rw encoding?                           identityref
     |  |  +--rw switching-type?                     identityref
     |  |  +--rw provisioning-state?                 identityref
     |  |  +--rw preference?                         uint8
     |  |  +--rw reoptimize-timer?                   uint16
     |  |  +--rw source?                             te-types:te-node-id
     |  |  +--rw destination?                        te-types:te-node-id
     |  |  +--rw src-tp-id?                          binary
     |  |  +--rw dst-tp-id?                          binary
     |  |  +--rw bidirectional?                      boolean
     |  |  +--rw association-objects
     |  |  |  +--rw association-object* [type ID source global-source]
     |  |  |  |  +--rw type             identityref
     |  |  |  |  +--rw ID               uint16
     |  |  |  |  +--rw source           inet:ip-address
     |  |  |  |  +--rw global-source    inet:ip-address
     |  |  |  +--rw association-object-extended* [type ID source
     global-source extended-ID]
     |  |  |     +--rw type             identityref
     |  |  |     +--rw ID               uint16
     |  |  |     +--rw source           inet:ip-address
     |  |  |     +--rw global-source    inet:ip-address
     |  |  |     +--rw extended-ID      binary
     |  |  +--rw protection
     |  |  |  +--rw enable?                         boolean
     |  |  |  +--rw protection-type?                identityref
     |  |  |  +--rw protection-reversion-disable?   boolean
     |  |  |  +--rw hold-off-time?                  uint32
     |  |  |  +--rw wait-to-revert?                 uint16
     |  |  |  +--rw aps-signal-id?                  uint8
     |  |  +--rw restoration
     |  |  |  +--rw enable?                          boolean
     |  |  |  +--rw restoration-type?                identityref
     |  |  |  +--rw restoration-scheme?              identityref
     |  |  |  +--rw restoration-reversion-disable?   boolean
     |  |  |  +--rw hold-off-time?                   uint32
     |  |  |  +--rw wait-to-restore?                 uint16
     |  |  |  +--rw wait-to-revert?                  uint16
     |  |  +--rw te-topology-identifier
     |  |  |  +--rw provider-id?   te-types:te-global-id
     |  |  |  +--rw client-id?     te-types:te-global-id
     |  |  |  +--rw topology-id?   te-types:te-topology-id
     |  |  +--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 dependency-tunnels
     |  |  |  +--rw dependency-tunnel* [name]
     |  |  |     +--rw name              ->
     ../../../../../tunnels/tunnel/name
     |  |  |     +--rw encoding?         identityref
     |  |  |     +--rw switching-type?   identityref
     |  |  +--rw hierarchical-link
     |  |  |  +--rw local-te-node-id?         te-types:te-node-id
     |  |  |  +--rw local-te-link-tp-id?      te-types:te-tp-id
     |  |  |  +--rw remote-te-node-id?        te-types:te-node-id
     |  |  |  +--rw te-topology-identifier
     |  |  |     +--rw provider-id?   te-types:te-global-id
     |  |  |     +--rw client-id?     te-types:te-global-id
     |  |  |     +--rw topology-id?   te-types:te-topology-id
     |  |  +--ro state
     |  |  |  +--ro operational-state?                  identityref
     |  |  |  +--ro te-dev:lsp-install-interval?        uint32
     |  |  |  +--ro te-dev:lsp-cleanup-interval?        uint32
     |  |  |  +--ro te-dev:lsp-invalidation-interval?   uint32
     |  |  +--rw p2p-primary-paths
     |  |  |  +--rw p2p-primary-path* [name]
     |  |  |     +--rw name                             string
     |  |  |     +--rw path-setup-protocol?             identityref
     |  |  |     +--rw path-computation-method?         identityref
     |  |  |     +--rw path-computation-server?         inet:ip-address
     |  |  |     +--rw compute-only?                    empty
     |  |  |     +--rw use-path-computation?            boolean
     |  |  |     +--rw lockdown?                        empty
     |  |  |     +--rw path-scope?                      identityref
     |  |  |     +--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 route-object-exclude-object*
     [index]
     |  |  |     |     |  |  |     +--rw index                  uint32
     |  |  |     |     |  |  |     +--rw (type)?
     |  |  |     |     |  |  |        +--:(num-unnum-hop)
     |  |  |     |     |  |  |        |  +--rw num-unnum-hop
     |  |  |     |     |  |  |        |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |     |  |  |        |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |     |  |  |        |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |  |        |     +--rw direction?
     te-link-direction
     |  |  |     |     |  |  |        +--:(as-number)
     |  |  |     |     |  |  |        |  +--rw as-number-hop
     |  |  |     |     |  |  |        |     +--rw as-number?   binary
     |  |  |     |     |  |  |        |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |  |        +--:(label)
     |  |  |     |     |  |  |        |  +--rw label-hop
     |  |  |     |     |  |  |        |     +--rw te-label
     |  |  |     |     |  |  |        |        +--rw (technology)?
     |  |  |     |     |  |  |        |        |  +--:(generic)
     |  |  |     |     |  |  |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |     |  |  |        |        +--rw direction?
     te-label-direction
     |  |  |     |     |  |  |        +--:(srlg)
     |  |  |     |     |  |  |           +--rw srlg
     |  |  |     |     |  |  |              +--rw srlg?   uint32
     |  |  |     |     |  |  +--rw explicit-route-include-objects
     |  |  |     |     |  |     +--rw route-object-include-object*
     [index]
     |  |  |     |     |  |        +--rw index                  uint32
     |  |  |     |     |  |        +--rw (type)?
     |  |  |     |     |  |           +--:(num-unnum-hop)
     |  |  |     |     |  |           |  +--rw num-unnum-hop
     |  |  |     |     |  |           |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |     |  |           |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |           |     +--rw direction?
     te-link-direction
     |  |  |     |     |  |           +--:(as-number)
     |  |  |     |     |  |           |  +--rw as-number-hop
     |  |  |     |     |  |           |     +--rw as-number?   binary
     |  |  |     |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |           +--:(label)
     |  |  |     |     |  |              +--rw label-hop
     |  |  |     |     |  |                 +--rw te-label
     |  |  |     |     |  |                    +--rw (technology)?
     |  |  |     |     |  |                    |  +--:(generic)
     |  |  |     |     |  |                    |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |     |  |                    +--rw direction?
     te-label-direction
     |  |  |     |     |  +--rw tiebreakers
     |  |  |     |     |     +--rw tiebreaker* [tiebreaker-type]
     |  |  |     |     |        +--rw tiebreaker-type    identityref
     |  |  |     |     +--:(objective-function)
     {path-optimization-objective-function}?
     |  |  |     |        +--rw objective-function
     |  |  |     |           +--rw objective-function-type?
     identityref
     |  |  |     +--rw preference?                      uint8
     |  |  |     +--rw k-requested-paths?               uint8
     |  |  |     +--rw named-path-constraint?           ->
../../../../../globals/named-path-constraints/named-path-constraint/name
     {te-types:named-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-values
     |  |  |     |  +--rw usage?    identityref
     |  |  |     |  +--rw values*   srlg
     |  |  |     +--rw path-srlgs-names
     |  |  |     |  +--rw path-srlgs-name* [usage]
     |  |  |     |     +--rw usage        identityref
     |  |  |     |     +--rw srlg-name* [name]
     |  |  |     |        +--rw name    string
     |  |  |     +--rw disjointness?
     te-types:te-path-disjointness
     |  |  |     +--rw explicit-route-objects
     |  |  |     |  +--rw route-object-exclude-always* [index]
     |  |  |     |  |  +--rw index                  uint32
     |  |  |     |  |  +--rw (type)?
     |  |  |     |  |     +--:(num-unnum-hop)
     |  |  |     |  |     |  +--rw num-unnum-hop
     |  |  |     |  |     |     +--rw node-id?      te-types:te-node-id
     |  |  |     |  |     |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |  |     |     +--rw hop-type?     te-hop-type
     |  |  |     |  |     |     +--rw direction?    te-link-direction
     |  |  |     |  |     +--:(as-number)
     |  |  |     |  |     |  +--rw as-number-hop
     |  |  |     |  |     |     +--rw as-number?   binary
     |  |  |     |  |     |     +--rw hop-type?    te-hop-type
     |  |  |     |  |     +--:(label)
     |  |  |     |  |        +--rw label-hop
     |  |  |     |  |           +--rw te-label
     |  |  |     |  |              +--rw (technology)?
     |  |  |     |  |              |  +--:(generic)
     |  |  |     |  |              |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |              +--rw direction?
     te-label-direction
     |  |  |     |  +--rw route-object-include-exclude* [index]
     |  |  |     |     +--rw explicit-route-usage?   identityref
     |  |  |     |     +--rw index                   uint32
     |  |  |     |     +--rw (type)?
     |  |  |     |        +--:(num-unnum-hop)
     |  |  |     |        |  +--rw num-unnum-hop
     |  |  |     |        |     +--rw node-id?      te-types:te-node-id
     |  |  |     |        |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |        |     +--rw hop-type?     te-hop-type
     |  |  |     |        |     +--rw direction?    te-link-direction
     |  |  |     |        +--:(as-number)
     |  |  |     |        |  +--rw as-number-hop
     |  |  |     |        |     +--rw as-number?   binary
     |  |  |     |        |     +--rw hop-type?    te-hop-type
     |  |  |     |        +--:(label)
     |  |  |     |        |  +--rw label-hop
     |  |  |     |        |     +--rw te-label
     |  |  |     |        |        +--rw (technology)?
     |  |  |     |        |        |  +--:(generic)
     |  |  |     |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |        +--rw direction?
     te-label-direction
     |  |  |     |        +--:(srlg)
     |  |  |     |           +--rw srlg
     |  |  |     |              +--rw srlg?   uint32
     |  |  |     +--rw shared-resources-tunnels
     |  |  |     |  +--rw lsp-shared-resources-tunnel*   tunnel-ref
     |  |  |     +--rw path-in-segment!
     |  |  |     |  +--rw label-restrictions
     |  |  |     |     +--rw label-restriction* [index]
     |  |  |     |        +--rw restriction?    enumeration
     |  |  |     |        +--rw index           uint32
     |  |  |     |        +--rw label-start
     |  |  |     |        |  +--rw te-label
     |  |  |     |        |     +--rw (technology)?
     |  |  |     |        |     |  +--:(generic)
     |  |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-end
     |  |  |     |        |  +--rw te-label
     |  |  |     |        |     +--rw (technology)?
     |  |  |     |        |     |  +--:(generic)
     |  |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-step
     |  |  |     |        |  +--rw (technology)?
     |  |  |     |        |     +--:(generic)
     |  |  |     |        |        +--rw generic?   int32
     |  |  |     |        +--rw range-bitmap?   binary
     |  |  |     +--rw path-out-segment!
     |  |  |     |  +--rw label-restrictions
     |  |  |     |     +--rw label-restriction* [index]
     |  |  |     |        +--rw restriction?    enumeration
     |  |  |     |        +--rw index           uint32
     |  |  |     |        +--rw label-start
     |  |  |     |        |  +--rw te-label
     |  |  |     |        |     +--rw (technology)?
     |  |  |     |        |     |  +--:(generic)
     |  |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-end
     |  |  |     |        |  +--rw te-label
     |  |  |     |        |     +--rw (technology)?
     |  |  |     |        |     |  +--:(generic)
     |  |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-step
     |  |  |     |        |  +--rw (technology)?
     |  |  |     |        |     +--:(generic)
     |  |  |     |        |        +--rw generic?   int32
     |  |  |     |        +--rw range-bitmap?   binary
     |  |  |     +--ro state
     |  |  |     |  +--ro computed-paths-properties
     |  |  |     |  |  +--ro computed-path-properties* [k-index]
     |  |  |     |  |     +--ro k-index            uint8
     |  |  |     |  |     +--ro path-properties
     |  |  |     |  |        +--ro path-metric* [metric-type]
     |  |  |     |  |        |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |        |  +--ro state
     |  |  |     |  |        |     +--ro metric-type?
     identityref
     |  |  |     |  |        |     +--ro accumulative-value?   uint64
     |  |  |     |  |        +--ro path-affinities-values
     |  |  |     |  |        |  +--ro path-affinities-value* [usage]
     |  |  |     |  |        |     +--ro usage    identityref
     |  |  |     |  |        |     +--ro value?   admin-groups
     |  |  |     |  |        +--ro path-affinity-names
     |  |  |     |  |        |  +--ro path-affinity-name* [usage]
     |  |  |     |  |        |     +--ro usage            identityref
     |  |  |     |  |        |     +--ro affinity-name* [name]
     |  |  |     |  |        |        +--ro name    string
     |  |  |     |  |        +--ro path-srlgs-values
     |  |  |     |  |        |  +--ro usage?    identityref
     |  |  |     |  |        |  +--ro values*   srlg
     |  |  |     |  |        +--ro path-srlgs-names
     |  |  |     |  |        |  +--ro path-srlgs-name* [usage]
     |  |  |     |  |        |     +--ro usage        identityref
     |  |  |     |  |        |     +--ro srlg-name* [name]
     |  |  |     |  |        |        +--ro name    string
     |  |  |     |  |        +--ro path-route-objects
     |  |  |     |  |        |  +--ro path-computed-route-object*
     [index]
     |  |  |     |  |        |     +--ro index    -> ../state/index
     |  |  |     |  |        |     +--ro state
     |  |  |     |  |        |        +--ro index?
     uint32
     |  |  |     |  |        |        +--ro (type)?
     |  |  |     |  |        |           +--:(num-unnum-hop)
     |  |  |     |  |        |           |  +--ro num-unnum-hop
     |  |  |     |  |        |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |        |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |        |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |        |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |        |           +--:(as-number)
     |  |  |     |  |        |           |  +--ro as-number-hop
     |  |  |     |  |        |           |     +--ro as-number?   binary
     |  |  |     |  |        |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |        |           +--:(label)
     |  |  |     |  |        |              +--ro label-hop
     |  |  |     |  |        |                 +--ro te-label
     |  |  |     |  |        |                    +--ro (technology)?
     |  |  |     |  |        |                    |  +--:(generic)
     |  |  |     |  |        |                    |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |        |                    +--ro direction?
     te-label-direction
     |  |  |     |  |        +--ro shared-resources-tunnels
     |  |  |     |  |           +--ro lsp-shared-resources-tunnel*
     tunnel-ref
     |  |  |     |  +--ro lsps
     |  |  |     |  |  +--ro lsp* [source destination tunnel-id lsp-id
     extended-tunnel-id]
     |  |  |     |  |     +--ro source
     inet:ip-address
     |  |  |     |  |     +--ro destination
     inet:ip-address
     |  |  |     |  |     +--ro tunnel-id
     uint16
     |  |  |     |  |     +--ro lsp-id
     uint16
     |  |  |     |  |     +--ro extended-tunnel-id
     inet:ip-address
     |  |  |     |  |     +--ro operational-state?
     identityref
     |  |  |     |  |     +--ro path-setup-protocol?
     identityref
     |  |  |     |  |     +--ro origin-type?
     enumeration
     |  |  |     |  |     +--ro lsp-resource-status?
     enumeration
     |  |  |     |  |     +--ro lockout-of-normal?
     boolean
     |  |  |     |  |     +--ro freeze?
     boolean
     |  |  |     |  |     +--ro lsp-protection-role?
     enumeration
     |  |  |     |  |     +--ro lsp-protection-state?
     identityref
     |  |  |     |  |     +--ro protection-group-ingress-node-id?
     te-types:te-node-id
     |  |  |     |  |     +--ro protection-group-egress-node-id?
     te-types:te-node-id
     |  |  |     |  |     +--ro lsp-shared-resources-tunnel?
     tunnel-ref
     |  |  |     |  |     +--ro lsp-record-route-subobjects
     |  |  |     |  |     |  +--ro record-route-subobject* [index]
     |  |  |     |  |     |     +--ro index               uint32
     |  |  |     |  |     |     +--ro (type)?
     |  |  |     |  |     |        +--:(numbered)
     |  |  |     |  |     |        |  +--ro address?
     te-types:te-tp-id
     |  |  |     |  |     |        |  +--ro ip-flags?     binary
     |  |  |     |  |     |        +--:(unnumbered)
     |  |  |     |  |     |        |  +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |     |        |  +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |        +--:(label)
     |  |  |     |  |     |           +--ro label-hop
     |  |  |     |  |     |              +--ro te-label
     |  |  |     |  |     |              |  +--ro (technology)?
     |  |  |     |  |     |              |  |  +--:(generic)
     |  |  |     |  |     |              |  |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |     |              |  +--ro direction?
     te-label-direction
     |  |  |     |  |     |              +--ro label-flags?   binary
     |  |  |     |  |     +--ro path-properties
     |  |  |     |  |     |  +--ro path-metric* [metric-type]
     |  |  |     |  |     |  |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |     |  |  +--ro state
     |  |  |     |  |     |  |     +--ro metric-type?
     identityref
     |  |  |     |  |     |  |     +--ro accumulative-value?   uint64
     |  |  |     |  |     |  +--ro path-affinities-values
     |  |  |     |  |     |  |  +--ro path-affinities-value* [usage]
     |  |  |     |  |     |  |     +--ro usage    identityref
     |  |  |     |  |     |  |     +--ro value?   admin-groups
     |  |  |     |  |     |  +--ro path-affinity-names
     |  |  |     |  |     |  |  +--ro path-affinity-name* [usage]
     |  |  |     |  |     |  |     +--ro usage            identityref
     |  |  |     |  |     |  |     +--ro affinity-name* [name]
     |  |  |     |  |     |  |        +--ro name    string
     |  |  |     |  |     |  +--ro path-srlgs-values
     |  |  |     |  |     |  |  +--ro usage?    identityref
     |  |  |     |  |     |  |  +--ro values*   srlg
     |  |  |     |  |     |  +--ro path-srlgs-names
     |  |  |     |  |     |  |  +--ro path-srlgs-name* [usage]
     |  |  |     |  |     |  |     +--ro usage        identityref
     |  |  |     |  |     |  |     +--ro srlg-name* [name]
     |  |  |     |  |     |  |        +--ro name    string
     |  |  |     |  |     |  +--ro path-route-objects
     |  |  |     |  |     |  |  +--ro path-computed-route-object*
     [index]
     |  |  |     |  |     |  |     +--ro index    -> ../state/index
     |  |  |     |  |     |  |     +--ro state
     |  |  |     |  |     |  |        +--ro index?
     uint32
     |  |  |     |  |     |  |        +--ro (type)?
     |  |  |     |  |     |  |           +--:(num-unnum-hop)
     |  |  |     |  |     |  |           |  +--ro num-unnum-hop
     |  |  |     |  |     |  |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |     |  |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |     |  |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |     |  |           +--:(as-number)
     |  |  |     |  |     |  |           |  +--ro as-number-hop
     |  |  |     |  |     |  |           |     +--ro as-number?   binary
     |  |  |     |  |     |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |     |  |           +--:(label)
     |  |  |     |  |     |  |              +--ro label-hop
     |  |  |     |  |     |  |                 +--ro te-label
     |  |  |     |  |     |  |                    +--ro (technology)?
     |  |  |     |  |     |  |                    |  +--:(generic)
     |  |  |     |  |     |  |                    |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |     |  |                    +--ro direction?
     te-label-direction
     |  |  |     |  |     |  +--ro shared-resources-tunnels
     |  |  |     |  |     |     +--ro lsp-shared-resources-tunnel*
     tunnel-ref
     |  |  |     |  |     +--ro te-dev:lsp-timers
     |  |  |     |  |     |  +--ro te-dev:life-time?         uint32
     |  |  |     |  |     |  +--ro te-dev:time-to-install?   uint32
     |  |  |     |  |     |  +--ro te-dev:time-to-destroy?   uint32
     |  |  |     |  |     +--ro te-dev:downstream-info
     |  |  |     |  |     |  +--ro te-dev:nhop?
     inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:outgoing-interface?
     if:interface-ref
     |  |  |     |  |     |  +--ro te-dev:neighbor?
     inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:label?
     rt-types:generalized-label
     |  |  |     |  |     +--ro te-dev:upstream-info
     |  |  |     |  |     |  +--ro te-dev:phop?       inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:neighbor?   inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:label?
     rt-types:generalized-label
     |  |  |     |  |     +--ro te-mpls:performance-metric-one-way
     |  |  |     |  |     |  +--ro te-mpls:one-way-delay?
     uint32
     |  |  |     |  |     |  +--ro te-mpls:one-way-min-delay?
     uint32
     |  |  |     |  |     |  +--ro te-mpls:one-way-max-delay?
     uint32
     |  |  |     |  |     |  +--ro te-mpls:one-way-delay-variation?
     uint32
     |  |  |     |  |     |  +--ro te-mpls:one-way-packet-loss?
     decimal64
     |  |  |     |  |     |  +--ro te-mpls:one-way-residual-bandwidth?
     rt-types:bandwidth-ieee-float32
     |  |  |     |  |     |  +--ro te-mpls:one-way-available-bandwidth?
     rt-types:bandwidth-ieee-float32
     |  |  |     |  |     |  +--ro te-mpls:one-way-utilized-bandwidth?
     rt-types:bandwidth-ieee-float32
     |  |  |     |  |     +--ro te-mpls:performance-metric-two-way
     |  |  |     |  |        +--ro te-mpls:two-way-delay?
     uint32
     |  |  |     |  |        +--ro te-mpls:two-way-min-delay?
     uint32
     |  |  |     |  |        +--ro te-mpls:two-way-max-delay?
     uint32
     |  |  |     |  |        +--ro te-mpls:two-way-delay-variation?
     uint32
     |  |  |     |  |        +--ro te-mpls:two-way-packet-loss?
     decimal64
     |  |  |     |  +--ro te-mpls:static-lsp-name?
     mpls-static:static-lsp-ref
     |  |  |     +--rw p2p-reverse-primary-path
     |  |  |     |  +--rw name?                         string
     |  |  |     |  +--rw path-setup-protocol?          identityref
     |  |  |     |  +--rw path-computation-method?      identityref
     |  |  |     |  +--rw path-computation-server?      inet:ip-address
     |  |  |     |  +--rw compute-only?                 empty
     |  |  |     |  +--rw use-path-computation?         boolean
     |  |  |     |  +--rw lockdown?                     empty
     |  |  |     |  +--rw path-scope?                   identityref
     |  |  |     |  +--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 route-object-exclude-object*
     [index]
     |  |  |     |  |     |  |  |     +--rw index
     uint32
     |  |  |     |  |     |  |  |     +--rw (type)?
     |  |  |     |  |     |  |  |        +--:(num-unnum-hop)
     |  |  |     |  |     |  |  |        |  +--rw num-unnum-hop
     |  |  |     |  |     |  |  |        |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |  |     |  |  |        |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |  |  |        |     +--rw hop-type?
     te-hop-type
     |  |  |     |  |     |  |  |        |     +--rw direction?
     te-link-direction
     |  |  |     |  |     |  |  |        +--:(as-number)
     |  |  |     |  |     |  |  |        |  +--rw as-number-hop
     |  |  |     |  |     |  |  |        |     +--rw as-number?   binary
     |  |  |     |  |     |  |  |        |     +--rw hop-type?
     te-hop-type
     |  |  |     |  |     |  |  |        +--:(label)
     |  |  |     |  |     |  |  |        |  +--rw label-hop
     |  |  |     |  |     |  |  |        |     +--rw te-label
     |  |  |     |  |     |  |  |        |        +--rw (technology)?
     |  |  |     |  |     |  |  |        |        |  +--:(generic)
     |  |  |     |  |     |  |  |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |     |  |  |        |        +--rw direction?
     te-label-direction
     |  |  |     |  |     |  |  |        +--:(srlg)
     |  |  |     |  |     |  |  |           +--rw srlg
     |  |  |     |  |     |  |  |              +--rw srlg?   uint32
     |  |  |     |  |     |  |  +--rw explicit-route-include-objects
     |  |  |     |  |     |  |     +--rw route-object-include-object*
     [index]
     |  |  |     |  |     |  |        +--rw index
     uint32
     |  |  |     |  |     |  |        +--rw (type)?
     |  |  |     |  |     |  |           +--:(num-unnum-hop)
     |  |  |     |  |     |  |           |  +--rw num-unnum-hop
     |  |  |     |  |     |  |           |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |  |     |  |           |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |  |     |  |           |     +--rw direction?
     te-link-direction
     |  |  |     |  |     |  |           +--:(as-number)
     |  |  |     |  |     |  |           |  +--rw as-number-hop
     |  |  |     |  |     |  |           |     +--rw as-number?   binary
     |  |  |     |  |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |  |     |  |           +--:(label)
     |  |  |     |  |     |  |              +--rw label-hop
     |  |  |     |  |     |  |                 +--rw te-label
     |  |  |     |  |     |  |                    +--rw (technology)?
     |  |  |     |  |     |  |                    |  +--:(generic)
     |  |  |     |  |     |  |                    |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |     |  |                    +--rw direction?
     te-label-direction
     |  |  |     |  |     |  +--rw tiebreakers
     |  |  |     |  |     |     +--rw tiebreaker* [tiebreaker-type]
     |  |  |     |  |     |        +--rw tiebreaker-type    identityref
     |  |  |     |  |     +--:(objective-function)
     {path-optimization-objective-function}?
     |  |  |     |  |        +--rw objective-function
     |  |  |     |  |           +--rw objective-function-type?
     identityref
     |  |  |     |  +--rw named-path-constraint?        ->
./../../../../globals/named-path-constraints/named-path-constraint/name
```

```
     {te-types:named-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-values
     |  |  |     |  |  +--rw usage?    identityref
     |  |  |     |  |  +--rw values*   srlg
     |  |  |     |  +--rw path-srlgs-names
     |  |  |     |  |  +--rw path-srlgs-name* [usage]
     |  |  |     |  |     +--rw usage        identityref
     |  |  |     |  |     +--rw srlg-name* [name]
     |  |  |     |  |        +--rw name    string
     |  |  |     |  +--rw disjointness?
     te-types:te-path-disjointness
     |  |  |     |  +--rw explicit-route-objects
     |  |  |     |  |  +--rw route-object-exclude-always* [index]
     |  |  |     |  |  |  +--rw index                  uint32
     |  |  |     |  |  |  +--rw (type)?
     |  |  |     |  |  |     +--:(num-unnum-hop)
     |  |  |     |  |  |     |  +--rw num-unnum-hop
     |  |  |     |  |  |     |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |  |  |     |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |  |  |     |     +--rw hop-type?     te-hop-type
     |  |  |     |  |  |     |     +--rw direction?    te-link-direction
     |  |  |     |  |  |     +--:(as-number)
     |  |  |     |  |  |     |  +--rw as-number-hop
     |  |  |     |  |  |     |     +--rw as-number?   binary
     |  |  |     |  |  |     |     +--rw hop-type?    te-hop-type
     |  |  |     |  |  |     +--:(label)
     |  |  |     |  |  |        +--rw label-hop
     |  |  |     |  |  |           +--rw te-label
     |  |  |     |  |  |              +--rw (technology)?
     |  |  |     |  |  |              |  +--:(generic)
     |  |  |     |  |  |              |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |  |              +--rw direction?
     te-label-direction
     |  |  |     |  |  +--rw route-object-include-exclude* [index]
     |  |  |     |  |     +--rw explicit-route-usage?   identityref
     |  |  |     |  |     +--rw index                   uint32
     |  |  |     |  |     +--rw (type)?
     |  |  |     |  |        +--:(num-unnum-hop)
     |  |  |     |  |        |  +--rw num-unnum-hop
     |  |  |     |  |        |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |  |        |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |  |        |     +--rw hop-type?     te-hop-type
     |  |  |     |  |        |     +--rw direction?    te-link-direction
     |  |  |     |  |        +--:(as-number)
     |  |  |     |  |        |  +--rw as-number-hop
     |  |  |     |  |        |     +--rw as-number?   binary
     |  |  |     |  |        |     +--rw hop-type?    te-hop-type
     |  |  |     |  |        +--:(label)
     |  |  |     |  |        |  +--rw label-hop
     |  |  |     |  |        |     +--rw te-label
     |  |  |     |  |        |        +--rw (technology)?
     |  |  |     |  |        |        |  +--:(generic)
     |  |  |     |  |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |        |        +--rw direction?
     te-label-direction
     |  |  |     |  |        +--:(srlg)
     |  |  |     |  |           +--rw srlg
     |  |  |     |  |              +--rw srlg?   uint32
     |  |  |     |  +--rw shared-resources-tunnels
     |  |  |     |  |  +--rw lsp-shared-resources-tunnel*   tunnel-ref
     |  |  |     |  +--rw path-in-segment!
     |  |  |     |  |  +--rw label-restrictions
     |  |  |     |  |     +--rw label-restriction* [index]
     |  |  |     |  |        +--rw restriction?    enumeration
     |  |  |     |  |        +--rw index           uint32
     |  |  |     |  |        +--rw label-start
     |  |  |     |  |        |  +--rw te-label
     |  |  |     |  |        |     +--rw (technology)?
     |  |  |     |  |        |     |  +--:(generic)
     |  |  |     |  |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |        |     +--rw direction?
     te-label-direction
     |  |  |     |  |        +--rw label-end
     |  |  |     |  |        |  +--rw te-label
     |  |  |     |  |        |     +--rw (technology)?
     |  |  |     |  |        |     |  +--:(generic)
     |  |  |     |  |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |        |     +--rw direction?
     te-label-direction
     |  |  |     |  |        +--rw label-step
     |  |  |     |  |        |  +--rw (technology)?
     |  |  |     |  |        |     +--:(generic)
     |  |  |     |  |        |        +--rw generic?   int32
     |  |  |     |  |        +--rw range-bitmap?   binary
     |  |  |     |  +--rw path-out-segment!
     |  |  |     |  |  +--rw label-restrictions
     |  |  |     |  |     +--rw label-restriction* [index]
     |  |  |     |  |        +--rw restriction?    enumeration
     |  |  |     |  |        +--rw index           uint32
     |  |  |     |  |        +--rw label-start
     |  |  |     |  |        |  +--rw te-label
     |  |  |     |  |        |     +--rw (technology)?
     |  |  |     |  |        |     |  +--:(generic)
     |  |  |     |  |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |        |     +--rw direction?
     te-label-direction
     |  |  |     |  |        +--rw label-end
     |  |  |     |  |        |  +--rw te-label
     |  |  |     |  |        |     +--rw (technology)?
     |  |  |     |  |        |     |  +--:(generic)
     |  |  |     |  |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |        |     +--rw direction?
     te-label-direction
     |  |  |     |  |        +--rw label-step
     |  |  |     |  |        |  +--rw (technology)?
     |  |  |     |  |        |     +--:(generic)
     |  |  |     |  |        |        +--rw generic?   int32
     |  |  |     |  |        +--rw range-bitmap?   binary
     |  |  |     |  +--ro state
     |  |  |     |  |  +--ro computed-paths-properties
     |  |  |     |  |  |  +--ro computed-path-properties* [k-index]
     |  |  |     |  |  |     +--ro k-index            uint8
     |  |  |     |  |  |     +--ro path-properties
     |  |  |     |  |  |        +--ro path-metric* [metric-type]
     |  |  |     |  |  |        |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |  |        |  +--ro state
     |  |  |     |  |  |        |     +--ro metric-type?
     identityref
     |  |  |     |  |  |        |     +--ro accumulative-value?   uint64
     |  |  |     |  |  |        +--ro path-affinities-values
     |  |  |     |  |  |        |  +--ro path-affinities-value* [usage]
     |  |  |     |  |  |        |     +--ro usage    identityref
     |  |  |     |  |  |        |     +--ro value?   admin-groups
     |  |  |     |  |  |        +--ro path-affinity-names
     |  |  |     |  |  |        |  +--ro path-affinity-name* [usage]
     |  |  |     |  |  |        |     +--ro usage            identityref
     |  |  |     |  |  |        |     +--ro affinity-name* [name]
     |  |  |     |  |  |        |        +--ro name    string
     |  |  |     |  |  |        +--ro path-srlgs-values
     |  |  |     |  |  |        |  +--ro usage?    identityref
     |  |  |     |  |  |        |  +--ro values*   srlg
     |  |  |     |  |  |        +--ro path-srlgs-names
     |  |  |     |  |  |        |  +--ro path-srlgs-name* [usage]
     |  |  |     |  |  |        |     +--ro usage        identityref
     |  |  |     |  |  |        |     +--ro srlg-name* [name]
     |  |  |     |  |  |        |        +--ro name    string
     |  |  |     |  |  |        +--ro path-route-objects
     |  |  |     |  |  |        |  +--ro path-computed-route-object*
     [index]
     |  |  |     |  |  |        |     +--ro index    -> ../state/index
     |  |  |     |  |  |        |     +--ro state
     |  |  |     |  |  |        |        +--ro index?
     uint32
     |  |  |     |  |  |        |        +--ro (type)?
     |  |  |     |  |  |        |           +--:(num-unnum-hop)
     |  |  |     |  |  |        |           |  +--ro num-unnum-hop
     |  |  |     |  |  |        |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |  |        |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |  |        |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |  |        |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |  |        |           +--:(as-number)
     |  |  |     |  |  |        |           |  +--ro as-number-hop
     |  |  |     |  |  |        |           |     +--ro as-number?
     binary
     |  |  |     |  |  |        |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |  |        |           +--:(label)
     |  |  |     |  |  |        |              +--ro label-hop
     |  |  |     |  |  |        |                 +--ro te-label
     |  |  |     |  |  |        |                    +--ro (technology)?
     |  |  |     |  |  |        |                    |  +--:(generic)
     |  |  |     |  |  |        |                    |     +--ro
     generic?   rt-types:generalized-label
     |  |  |     |  |  |        |                    +--ro direction?
     te-label-direction
     |  |  |     |  |  |        +--ro shared-resources-tunnels
     |  |  |     |  |  |           +--ro lsp-shared-resources-tunnel*
     tunnel-ref
     |  |  |     |  |  +--ro lsps
     |  |  |     |  |     +--ro lsp* [source destination tunnel-id
     lsp-id extended-tunnel-id]
     |  |  |     |  |        +--ro source
     inet:ip-address
     |  |  |     |  |        +--ro destination
     inet:ip-address
     |  |  |     |  |        +--ro tunnel-id
     uint16
     |  |  |     |  |        +--ro lsp-id
     uint16
     |  |  |     |  |        +--ro extended-tunnel-id
     inet:ip-address
     |  |  |     |  |        +--ro operational-state?
     identityref
     |  |  |     |  |        +--ro path-setup-protocol?
     identityref
     |  |  |     |  |        +--ro origin-type?
     enumeration
     |  |  |     |  |        +--ro lsp-resource-status?
     enumeration
     |  |  |     |  |        +--ro lockout-of-normal?
     boolean
     |  |  |     |  |        +--ro freeze?
     boolean
     |  |  |     |  |        +--ro lsp-protection-role?
     enumeration
     |  |  |     |  |        +--ro lsp-protection-state?
     identityref
     |  |  |     |  |        +--ro protection-group-ingress-node-id?
     te-types:te-node-id
     |  |  |     |  |        +--ro protection-group-egress-node-id?
     te-types:te-node-id
     |  |  |     |  |        +--ro lsp-shared-resources-tunnel?
     tunnel-ref
     |  |  |     |  |        +--ro lsp-record-route-subobjects
     |  |  |     |  |        |  +--ro record-route-subobject* [index]
     |  |  |     |  |        |     +--ro index                 uint32
     |  |  |     |  |        |     +--ro (type)?
     |  |  |     |  |        |        +--:(numbered)
     |  |  |     |  |        |        |  +--ro address?
     te-types:te-tp-id
     |  |  |     |  |        |        |  +--ro ip-flags?     binary
     |  |  |     |  |        |        +--:(unnumbered)
     |  |  |     |  |        |        |  +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |        |        |  +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |        |        +--:(label)
     |  |  |     |  |        |           +--ro label-hop
     |  |  |     |  |        |              +--ro te-label
     |  |  |     |  |        |              |  +--ro (technology)?
     |  |  |     |  |        |              |  |  +--:(generic)
     |  |  |     |  |        |              |  |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |        |              |  +--ro direction?
     te-label-direction
     |  |  |     |  |        |              +--ro label-flags?   binary
     |  |  |     |  |        +--ro path-properties
     |  |  |     |  |           +--ro path-metric* [metric-type]
     |  |  |     |  |           |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |           |  +--ro state
     |  |  |     |  |           |     +--ro metric-type?
     identityref
     |  |  |     |  |           |     +--ro accumulative-value?   uint64
     |  |  |     |  |           +--ro path-affinities-values
     |  |  |     |  |           |  +--ro path-affinities-value* [usage]
     |  |  |     |  |           |     +--ro usage    identityref
     |  |  |     |  |           |     +--ro value?   admin-groups
     |  |  |     |  |           +--ro path-affinity-names
     |  |  |     |  |           |  +--ro path-affinity-name* [usage]
     |  |  |     |  |           |     +--ro usage            identityref
     |  |  |     |  |           |     +--ro affinity-name* [name]
     |  |  |     |  |           |        +--ro name    string
     |  |  |     |  |           +--ro path-srlgs-values
     |  |  |     |  |           |  +--ro usage?    identityref
     |  |  |     |  |           |  +--ro values*   srlg
     |  |  |     |  |           +--ro path-srlgs-names
     |  |  |     |  |           |  +--ro path-srlgs-name* [usage]
     |  |  |     |  |           |     +--ro usage        identityref
     |  |  |     |  |           |     +--ro srlg-name* [name]
     |  |  |     |  |           |        +--ro name    string
     |  |  |     |  |           +--ro path-route-objects
     |  |  |     |  |           |  +--ro path-computed-route-object*
     [index]
     |  |  |     |  |           |     +--ro index    -> ../state/index
     |  |  |     |  |           |     +--ro state
     |  |  |     |  |           |        +--ro index?
     uint32
     |  |  |     |  |           |        +--ro (type)?
     |  |  |     |  |           |           +--:(num-unnum-hop)
     |  |  |     |  |           |           |  +--ro num-unnum-hop
     |  |  |     |  |           |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |           |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |           |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |           |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |           |           +--:(as-number)
     |  |  |     |  |           |           |  +--ro as-number-hop
     |  |  |     |  |           |           |     +--ro as-number?
     binary
     |  |  |     |  |           |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |           |           +--:(label)
     |  |  |     |  |           |              +--ro label-hop
     |  |  |     |  |           |                 +--ro te-label
     |  |  |     |  |           |                    +--ro (technology)?
     |  |  |     |  |           |                    |  +--:(generic)
     |  |  |     |  |           |                    |     +--ro
     generic?   rt-types:generalized-label
     |  |  |     |  |           |                    +--ro direction?
     te-label-direction
     |  |  |     |  |           +--ro shared-resources-tunnels
     |  |  |     |  |              +--ro lsp-shared-resources-tunnel*
     tunnel-ref
     |  |  |     |  +--rw p2p-reverse-secondary-path
     |  |  |     |     +--rw secondary-path?        ->
     ../../../../../p2p-secondary-paths/p2p-secondary-path/name
     |  |  |     |     +--rw path-setup-protocol?   identityref
     |  |  |     +--rw candidate-p2p-secondary-paths
     |  |  |     |  +--rw candidate-p2p-secondary-path* [secondary-path]
     |  |  |     |     +--rw secondary-path        ->
     ../../../../../p2p-secondary-paths/p2p-secondary-path/name
     |  |  |     |     +--rw path-setup-protocol?   identityref
     |  |  |     |     +--ro state
     |  |  |     |        +--ro active?   boolean
     |  |  |     +--rw te-mpls:static-lsp-name?
     mpls-static:static-lsp-ref
     |  |  +--rw p2p-secondary-paths
     |  |  |  +--rw p2p-secondary-path* [name]
     |  |  |     +--rw name                           string
     |  |  |     +--rw path-setup-protocol?           identityref
     |  |  |     +--rw path-computation-method?       identityref
     |  |  |     +--rw path-computation-server?       inet:ip-address
     |  |  |     +--rw compute-only?                  empty
     |  |  |     +--rw use-path-computation?          boolean
     |  |  |     +--rw lockdown?                      empty
     |  |  |     +--rw path-scope?                    identityref
     |  |  |     +--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 route-object-exclude-object*
     [index]
     |  |  |     |     |  |  |     +--rw index                uint32
     |  |  |     |     |  |  |     +--rw (type)?
     |  |  |     |     |  |  |        +--:(num-unnum-hop)
     |  |  |     |     |  |  |        |  +--rw num-unnum-hop
     |  |  |     |     |  |  |        |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |     |  |  |        |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |     |  |  |        |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |  |        |     +--rw direction?
     te-link-direction
     |  |  |     |     |  |  |        +--:(as-number)
     |  |  |     |     |  |  |        |  +--rw as-number-hop
     |  |  |     |     |  |  |        |     +--rw as-number?   binary
     |  |  |     |     |  |  |        |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |  |        +--:(label)
     |  |  |     |     |  |  |        |  +--rw label-hop
     |  |  |     |     |  |  |        |     +--rw te-label
     |  |  |     |     |  |  |        |        +--rw (technology)?
     |  |  |     |     |  |  |        |        |  +--:(generic)
     |  |  |     |     |  |  |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |     |  |  |        |        +--rw direction?
     te-label-direction
     |  |  |     |     |  |  |        +--:(srlg)
     |  |  |     |     |  |  |           +--rw srlg
     |  |  |     |     |  |  |              +--rw srlg?   uint32
     |  |  |     |     |  |  +--rw explicit-route-include-objects
     |  |  |     |     |  |     +--rw route-object-include-object*
     [index]
     |  |  |     |     |  |        +--rw index                  uint32
     |  |  |     |     |  |        +--rw (type)?
     |  |  |     |     |  |           +--:(num-unnum-hop)
     |  |  |     |     |  |           |  +--rw num-unnum-hop
     |  |  |     |     |  |           |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |     |  |           |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |           |     +--rw direction?
     te-link-direction
     |  |  |     |     |  |           +--:(as-number)
     |  |  |     |     |  |           |  +--rw as-number-hop
     |  |  |     |     |  |           |     +--rw as-number?   binary
     |  |  |     |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |           +--:(label)
     |  |  |     |     |  |              +--rw label-hop
     |  |  |     |     |  |                 +--rw te-label
     |  |  |     |     |  |                    +--rw (technology)?
     |  |  |     |     |  |                    |  +--:(generic)
     |  |  |     |     |  |                    |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |     |  |                    +--rw direction?
     te-label-direction
     |  |  |     |     |  +--rw tiebreakers
     |  |  |     |     |     +--rw tiebreaker* [tiebreaker-type]
     |  |  |     |     |        +--rw tiebreaker-type    identityref
     |  |  |     |     +--:(objective-function)
     {path-optimization-objective-function}?
     |  |  |     |        +--rw objective-function
     |  |  |     |           +--rw objective-function-type?
     identityref
     |  |  |     +--rw preference?                 uint8
     |  |  |     +--rw k-requested-paths?          uint8
     |  |  |     +--rw named-path-constraint?      ->
../../../../../globals/named-path-constraints/named-path-constraint/name
     {te-types:named-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-values
     |  |  |     |  +--rw usage?    identityref
     |  |  |     |  +--rw values*   srlg
     |  |  |     +--rw path-srlgs-names
     |  |  |     |  +--rw path-srlgs-name* [usage]
     |  |  |     |     +--rw usage        identityref
     |  |  |     |     +--rw srlg-name* [name]
     |  |  |     |        +--rw name    string
     |  |  |     +--rw disjointness?
     te-types:te-path-disjointness
     |  |  |     +--rw explicit-route-objects
     |  |  |     |  +--rw route-object-exclude-always* [index]
     |  |  |     |  |  +--rw index                   uint32
     |  |  |     |  |  +--rw (type)?
     |  |  |     |  |     +--:(num-unnum-hop)
     |  |  |     |  |     |  +--rw num-unnum-hop
     |  |  |     |  |     |     +--rw node-id?      te-types:te-node-id
     |  |  |     |  |     |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |  |     |     +--rw hop-type?     te-hop-type
     |  |  |     |  |     |     +--rw direction?    te-link-direction
     |  |  |     |  |     +--:(as-number)
     |  |  |     |  |     |  +--rw as-number-hop
     |  |  |     |  |     |     +--rw as-number?   binary
     |  |  |     |  |     |     +--rw hop-type?    te-hop-type
     |  |  |     |  |     +--:(label)
     |  |  |     |  |        +--rw label-hop
     |  |  |     |  |           +--rw te-label
     |  |  |     |  |              +--rw (technology)?
     |  |  |     |  |              |  +--:(generic)
     |  |  |     |  |              |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |              +--rw direction?
     te-label-direction
     |  |  |     |  +--rw route-object-include-exclude* [index]
     |  |  |     |     +--rw explicit-route-usage?   identityref
     |  |  |     |     +--rw index                   uint32
     |  |  |     |     +--rw (type)?
     |  |  |     |        +--:(num-unnum-hop)
     |  |  |     |        |  +--rw num-unnum-hop
     |  |  |     |        |     +--rw node-id?      te-types:te-node-id
     |  |  |     |        |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |        |     +--rw hop-type?     te-hop-type
     |  |  |     |        |     +--rw direction?    te-link-direction
     |  |  |     |        +--:(as-number)
     |  |  |     |        |  +--rw as-number-hop
     |  |  |     |        |     +--rw as-number?   binary
     |  |  |     |        |     +--rw hop-type?    te-hop-type
     |  |  |     |        +--:(label)
     |  |  |     |        |  +--rw label-hop
     |  |  |     |        |     +--rw te-label
     |  |  |     |        |        +--rw (technology)?
     |  |  |     |        |        |  +--:(generic)
     |  |  |     |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |        +--rw direction?
     te-label-direction
     |  |  |     |        +--:(srlg)
     |  |  |     |           +--rw srlg
     |  |  |     |              +--rw srlg?   uint32
     |  |  |     +--rw shared-resources-tunnels
     |  |  |     |  +--rw lsp-shared-resources-tunnel*   tunnel-ref
     |  |  |     +--rw path-in-segment!
     |  |  |     |  +--rw label-restrictions
     |  |  |     |     +--rw label-restriction* [index]
     |  |  |     |        +--rw restriction?    enumeration
     |  |  |     |        +--rw index           uint32
     |  |  |     |        +--rw label-start
     |  |  |     |        |  +--rw te-label
     |  |  |     |        |     +--rw (technology)?
     |  |  |     |        |     |  +--:(generic)
     |  |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-end
     |  |  |     |        |  +--rw te-label
     |  |  |     |        |     +--rw (technology)?
     |  |  |     |        |     |  +--:(generic)
     |  |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-step
     |  |  |     |        |  +--rw (technology)?
     |  |  |     |        |     +--:(generic)
     |  |  |     |        |        +--rw generic?   int32
     |  |  |     |        +--rw range-bitmap?   binary
     |  |  |     +--rw path-out-segment!
     |  |  |     |  +--rw label-restrictions
     |  |  |     |     +--rw label-restriction* [index]
     |  |  |     |        +--rw restriction?    enumeration
     |  |  |     |        +--rw index           uint32
     |  |  |     |        +--rw label-start
     |  |  |     |        |  +--rw te-label
     |  |  |     |        |     +--rw (technology)?
     |  |  |     |        |     |  +--:(generic)
     |  |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-end
     |  |  |     |        |  +--rw te-label
     |  |  |     |        |     +--rw (technology)?
     |  |  |     |        |     |  +--:(generic)
     |  |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-step
     |  |  |     |        |  +--rw (technology)?
     |  |  |     |        |     +--:(generic)
     |  |  |     |        |        +--rw generic?   int32
     |  |  |     |        +--rw range-bitmap?   binary
     |  |  |     +--rw protection
     |  |  |     |  +--rw enable?                          boolean
     |  |  |     |  +--rw protection-type?                 identityref
     |  |  |     |  +--rw protection-reversion-disable?   boolean
     |  |  |     |  +--rw hold-off-time?                  uint32
     |  |  |     |  +--rw wait-to-revert?                 uint16
     |  |  |     |  +--rw aps-signal-id?                  uint8
     |  |  |     +--rw restoration
     |  |  |     |  +--rw enable?                           boolean
     |  |  |     |  +--rw restoration-type?                 identityref
     |  |  |     |  +--rw restoration-scheme?               identityref
     |  |  |     |  +--rw restoration-reversion-disable?   boolean
     |  |  |     |  +--rw hold-off-time?                   uint32
     |  |  |     |  +--rw wait-to-restore?                 uint16
     |  |  |     |  +--rw wait-to-revert?                  uint16
     |  |  |     +--ro state
     |  |  |     |  +--ro computed-paths-properties
     |  |  |     |  |  +--ro computed-path-properties* [k-index]
     |  |  |     |  |     +--ro k-index            uint8
     |  |  |     |  |     +--ro path-properties
     |  |  |     |  |        +--ro path-metric* [metric-type]
     |  |  |     |  |        |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |        |  +--ro state
     |  |  |     |  |        |     +--ro metric-type?
     identityref
     |  |  |     |  |        |     +--ro accumulative-value?   uint64
     |  |  |     |  |        +--ro path-affinities-values
     |  |  |     |  |        |  +--ro path-affinities-value* [usage]
     |  |  |     |  |        |     +--ro usage    identityref
     |  |  |     |  |        |     +--ro value?   admin-groups
     |  |  |     |  |        +--ro path-affinity-names
     |  |  |     |  |        |  +--ro path-affinity-name* [usage]
     |  |  |     |  |        |     +--ro usage            identityref
     |  |  |     |  |        |     +--ro affinity-name* [name]
     |  |  |     |  |        |        +--ro name    string
     |  |  |     |  |        +--ro path-srlgs-values
     |  |  |     |  |        |  +--ro usage?    identityref
     |  |  |     |  |        |  +--ro values*   srlg
     |  |  |     |  |        +--ro path-srlgs-names
     |  |  |     |  |        |  +--ro path-srlgs-name* [usage]
     |  |  |     |  |        |     +--ro usage        identityref
     |  |  |     |  |        |     +--ro srlg-name* [name]
     |  |  |     |  |        |        +--ro name    string
     |  |  |     |  |        +--ro path-route-objects
     |  |  |     |  |        |  +--ro path-computed-route-object*
     [index]
     |  |  |     |  |        |     +--ro index    -> ../state/index
     |  |  |     |  |        |     +--ro state
     |  |  |     |  |        |        +--ro index?
     uint32
     |  |  |     |  |        |        +--ro (type)?
     |  |  |     |  |        |           +--:(num-unnum-hop)
     |  |  |     |  |        |           |  +--ro num-unnum-hop
     |  |  |     |  |        |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |        |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |        |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |        |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |        |           +--:(as-number)
     |  |  |     |  |        |           |  +--ro as-number-hop
     |  |  |     |  |        |           |     +--ro as-number?   binary
     |  |  |     |  |        |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |        |           +--:(label)
     |  |  |     |  |        |              +--ro label-hop
     |  |  |     |  |        |                 +--ro te-label
     |  |  |     |  |        |                    +--ro (technology)?
     |  |  |     |  |        |                    |  +--:(generic)
     |  |  |     |  |        |                    |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |        |                    +--ro direction?
     te-label-direction
     |  |  |     |  |        +--ro shared-resources-tunnels
     |  |  |     |  |           +--ro lsp-shared-resources-tunnel*
     tunnel-ref
     |  |  |     |  +--ro lsps
     |  |  |     |  |  +--ro lsp* [source destination tunnel-id lsp-id
     extended-tunnel-id]
     |  |  |     |  |     +--ro source
     inet:ip-address
     |  |  |     |  |     +--ro destination
     inet:ip-address
     |  |  |     |  |     +--ro tunnel-id
     uint16
     |  |  |     |  |     +--ro lsp-id
     uint16
     |  |  |     |  |     +--ro extended-tunnel-id
     inet:ip-address
     |  |  |     |  |     +--ro operational-state?
     identityref
     |  |  |     |  |     +--ro path-setup-protocol?
     identityref
     |  |  |     |  |     +--ro origin-type?
     enumeration
     |  |  |     |  |     +--ro lsp-resource-status?
     enumeration
     |  |  |     |  |     +--ro lockout-of-normal?
     boolean
     |  |  |     |  |     +--ro freeze?
     boolean
     |  |  |     |  |     +--ro lsp-protection-role?
     enumeration
     |  |  |     |  |     +--ro lsp-protection-state?
     identityref
     |  |  |     |  |     +--ro protection-group-ingress-node-id?
     te-types:te-node-id
     |  |  |     |  |     +--ro protection-group-egress-node-id?
     te-types:te-node-id
     |  |  |     |  |     +--ro lsp-shared-resources-tunnel?
     tunnel-ref
     |  |  |     |  |     +--ro lsp-record-route-subobjects
     |  |  |     |  |     |  +--ro record-route-subobject* [index]
     |  |  |     |  |     |     +--ro index                uint32
     |  |  |     |  |     |     +--ro (type)?
     |  |  |     |  |     |        +--:(numbered)
     |  |  |     |  |     |        |  +--ro address?
     te-types:te-tp-id
     |  |  |     |  |     |        |  +--ro ip-flags?     binary
     |  |  |     |  |     |        +--:(unnumbered)
     |  |  |     |  |     |        |  +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |     |        |  +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |        +--:(label)
     |  |  |     |  |     |           +--ro label-hop
     |  |  |     |  |     |              +--ro te-label
     |  |  |     |  |     |              |  +--ro (technology)?
     |  |  |     |  |     |              |  |  +--:(generic)
     |  |  |     |  |     |              |  |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |     |              |  +--ro direction?
     te-label-direction
     |  |  |     |  |     |              +--ro label-flags?   binary
     |  |  |     |  |     +--ro path-properties
     |  |  |     |  |     |  +--ro path-metric* [metric-type]
     |  |  |     |  |     |  |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |     |  |  +--ro state
     |  |  |     |  |     |  |     +--ro metric-type?
     identityref
     |  |  |     |  |     |  |     +--ro accumulative-value?   uint64
     |  |  |     |  |     |  +--ro path-affinities-values
     |  |  |     |  |     |  |  +--ro path-affinities-value* [usage]
     |  |  |     |  |     |  |     +--ro usage    identityref
     |  |  |     |  |     |  |     +--ro value?   admin-groups
     |  |  |     |  |     |  +--ro path-affinity-names
     |  |  |     |  |     |  |  +--ro path-affinity-name* [usage]
     |  |  |     |  |     |  |     +--ro usage            identityref
     |  |  |     |  |     |  |     +--ro affinity-name* [name]
     |  |  |     |  |     |  |        +--ro name    string
     |  |  |     |  |     |  +--ro path-srlgs-values
     |  |  |     |  |     |  |  +--ro usage?    identityref
     |  |  |     |  |     |  |  +--ro values*   srlg
     |  |  |     |  |     |  +--ro path-srlgs-names
     |  |  |     |  |     |  |  +--ro path-srlgs-name* [usage]
     |  |  |     |  |     |  |     +--ro usage        identityref
     |  |  |     |  |     |  |     +--ro srlg-name* [name]
     |  |  |     |  |     |  |        +--ro name    string
     |  |  |     |  |     |  +--ro path-route-objects
     |  |  |     |  |     |  |  +--ro path-computed-route-object*
     [index]
     |  |  |     |  |     |  |     +--ro index    -> ../state/index
     |  |  |     |  |     |  |     +--ro state
     |  |  |     |  |     |  |        +--ro index?
     uint32
     |  |  |     |  |     |  |        +--ro (type)?
     |  |  |     |  |     |  |           +--:(num-unnum-hop)
     |  |  |     |  |     |  |           |  +--ro num-unnum-hop
     |  |  |     |  |     |  |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |     |  |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |     |  |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |     |  |           +--:(as-number)
     |  |  |     |  |     |  |           |  +--ro as-number-hop
     |  |  |     |  |     |  |           |     +--ro as-number?   binary
     |  |  |     |  |     |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |     |  |           +--:(label)
     |  |  |     |  |     |  |              +--ro label-hop
     |  |  |     |  |     |  |                 +--ro te-label
     |  |  |     |  |     |  |                    +--ro (technology)?
     |  |  |     |  |     |  |                    |  +--:(generic)
     |  |  |     |  |     |  |                    |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |     |  |                    +--ro direction?
     te-label-direction
     |  |  |     |  |     |  +--ro shared-resources-tunnels
     |  |  |     |  |     |     +--ro lsp-shared-resources-tunnel*
     tunnel-ref
     |  |  |     |  |     +--ro te-dev:lsp-timers
     |  |  |     |  |     |  +--ro te-dev:life-time?         uint32
     |  |  |     |  |     |  +--ro te-dev:time-to-install?   uint32
     |  |  |     |  |     |  +--ro te-dev:time-to-destroy?   uint32
     |  |  |     |  |     +--ro te-dev:downstream-info
     |  |  |     |  |     |  +--ro te-dev:nhop?
     inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:outgoing-interface?
     if:interface-ref
     |  |  |     |  |     |  +--ro te-dev:neighbor?
     inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:label?
     rt-types:generalized-label
     |  |  |     |  |     +--ro te-dev:upstream-info
     |  |  |     |  |        +--ro te-dev:phop?       inet:ip-address
     |  |  |     |  |        +--ro te-dev:neighbor?   inet:ip-address
     |  |  |     |  |        +--ro te-dev:label?
     rt-types:generalized-label
     |  |  |     |  +--ro te-mpls:static-lsp-name?
     mpls-static:static-lsp-ref
     |  |  |     +--rw te-mpls:static-lsp-name?
     mpls-static:static-lsp-ref
     |  |  +---x tunnel-action
     |  |  |  +---w input
     |  |  |  |  +---w action-type?   identityref
     |  |  |  +--ro output
     |  |  |     +--ro action-result?   identityref
     |  |  +---x protection-external-commands
     |  |  |  +---w input
     |  |  |     +---w protection-external-command?        identityref
     |  |  |     +---w protection-group-ingress-node-id?
     te-types:te-node-id
     |  |  |     +---w protection-group-egress-node-id?
     te-types:te-node-id
     |  |  |     +---w path-ref?                           path-ref
     |  |  |     +---w traffic-type?                       enumeration
     |  |  |     +---w extra-traffic-tunnel-ref?           tunnel-ref
     |  |  +--rw te-dev:lsp-install-interval?        uint32
     |  |  +--rw te-dev:lsp-cleanup-interval?        uint32
     |  |  +--rw te-dev:lsp-invalidation-interval?   uint32
     |  |  +--rw te-mpls:tunnel-igp-shortcut
     |  |  |  +--rw te-mpls:shortcut-eligible?   boolean
     |  |  |  +--rw te-mpls:metric-type?         identityref
     |  |  |  +--rw te-mpls:metric?              int32
     |  |  |  +--rw te-mpls:routing-afs*         inet:ip-version
     |  |  +--rw te-mpls:forwarding
     |  |  |  +--rw te-mpls:binding-label?   rt-types:mpls-label
     |  |  |  +--rw te-mpls:load-share?      uint32
     |  |  |  +--rw te-mpls:policy-class?    uint8
     |  |  +--rw te-mpls:bandwidth-mpls
     |  |     +--rw te-mpls:specification-type?
     te-mpls-types:te-bandwidth-requested-type
     |  |     +--rw te-mpls:set-bandwidth?
     te-mpls-types:bandwidth-kbps
     |  |     +--rw te-mpls:class-type?           te-types:te-ds-class
     |  |     +--ro te-mpls:state
     |  |     |  +--ro te-mpls:signaled-bandwidth?
     te-mpls-types:bandwidth-kbps
     |  |     +--rw te-mpls:auto-bandwidth
     |  |        +--rw te-mpls:enabled?            boolean
     |  |        +--rw te-mpls:min-bw?
     te-mpls-types:bandwidth-kbps
     |  |        +--rw te-mpls:max-bw?
     te-mpls-types:bandwidth-kbps
     |  |        +--rw te-mpls:adjust-interval?    uint32
     |  |        +--rw te-mpls:adjust-threshold?   rt-types:percentage
     |  |        +--rw te-mpls:overflow
     |  |        |  +--rw te-mpls:enabled?               boolean
     |  |        |  +--rw te-mpls:overflow-threshold?
     rt-types:percentage
     |  |        |  +--rw te-mpls:trigger-event-count?   uint16
     |  |        +--rw te-mpls:underflow
     |  |           +--rw te-mpls:enabled?               boolean
     |  |           +--rw te-mpls:underflow-threshold?
     rt-types:percentage
     |  |           +--rw te-mpls:trigger-event-count?   uint16
     |  +--rw tunnel-p2mp* [name]
     |     +--rw name           string
     |     +--rw identifier?    uint16
     |     +--rw description?   string
     |     +--ro state
     |        +--ro operational-state?   identityref
     +--ro lsps-state
     |  +--ro lsp* [source destination tunnel-id lsp-id
     extended-tunnel-id]
     |     +--ro source                               inet:ip-address
     |     +--ro destination                          inet:ip-address
     |     +--ro tunnel-id                            uint16
     |     +--ro lsp-id                               uint16
     |     +--ro extended-tunnel-id                   inet:ip-address
     |     +--ro operational-state?                   identityref
     |     +--ro path-setup-protocol?                 identityref
     |     +--ro origin-type?                         enumeration
     |     +--ro lsp-resource-status?                 enumeration
     |     +--ro lockout-of-normal?                   boolean
     |     +--ro freeze?                              boolean
     |     +--ro lsp-protection-role?                 enumeration
     |     +--ro lsp-protection-state?                identityref
     |     +--ro protection-group-ingress-node-id?   te-types:te-node-id
     |     +--ro protection-group-egress-node-id?    te-types:te-node-id
     |     +--ro lsp-record-route-subobjects
     |     |  +--ro record-route-subobject* [index]
     |     |     +--ro index              uint32
     |     |     +--ro (type)?
     |     |        +--:(numbered)
     |     |        |  +--ro address?      te-types:te-tp-id
     |     |        |  +--ro ip-flags?     binary
     |     |        +--:(unnumbered)
     |     |        |  +--ro node-id?      te-types:te-node-id
     |     |        |  +--ro link-tp-id?   te-types:te-tp-id
     |     |        +--:(label)
     |     |           +--ro label-hop
     |     |              +--ro te-label
     |     |              |  +--ro (technology)?
     |     |              |  |  +--:(generic)
     |     |              |  |     +--ro generic?
     rt-types:generalized-label
     |     |              |  +--ro direction?       te-label-direction
     |     |              +--ro label-flags?   binary
     |     +--ro te-dev:lsp-timers
     |     |  +--ro te-dev:life-time?         uint32
     |     |  +--ro te-dev:time-to-install?   uint32
     |     |  +--ro te-dev:time-to-destroy?   uint32
     |     +--ro te-dev:downstream-info
     |     |  +--ro te-dev:nhop?                 inet:ip-address
     |     |  +--ro te-dev:outgoing-interface?   if:interface-ref
     |     |  +--ro te-dev:neighbor?             inet:ip-address
     |     |  +--ro te-dev:label?
     rt-types:generalized-label
     |     +--ro te-dev:upstream-info
     |        +--ro te-dev:phop?       inet:ip-address
     |        +--ro te-dev:neighbor?   inet:ip-address
     |        +--ro te-dev:label?      rt-types:generalized-label
     +--rw te-dev:interfaces
        +--rw te-dev:threshold-type?            enumeration
        +--rw te-dev:delta-percentage?          rt-types:percentage
        +--rw te-dev:threshold-specification?   enumeration
        +--rw te-dev:up-thresholds*             rt-types:percentage
        +--rw te-dev:down-thresholds*           rt-types:percentage
        +--rw te-dev:up-down-thresholds*        rt-types:percentage
        +--rw te-dev:interface* [interface]
           +--rw te-dev:interface
           if:interface-ref
           +--rw te-dev:te-metric?
           te-types:te-metric
           +--rw (te-dev:admin-group-type)?
           |  +--:(te-dev:value-admin-groups)
           |  |  +--rw (te-dev:value-admin-group-type)?
           |  |     +--:(te-dev:admin-groups)
           |  |     |  +--rw te-dev:admin-group?
           te-types:admin-group
           |  |     +--:(te-dev:extended-admin-groups)
           {te-types:extended-admin-groups}?
           |  |        +--rw te-dev:extended-admin-group?
           te-types:extended-admin-group
           |  +--:(te-dev:named-admin-groups)
           |     +--rw te-dev:named-admin-groups* [named-admin-group]
{te-types:extended-admin-groups,te-types:named-extended-admin-groups}?
           |        +--rw te-dev:named-admin-group    ->
        ../../../../te:globals/named-admin-groups/named-admin-group/name
           +--rw (te-dev:srlg-type)?
           |  +--:(te-dev:value-srlgs)
           |  |  +--rw te-dev:values* [value]
           |  |     +--rw te-dev:value    uint32
           |  +--:(te-dev:named-srlgs)
           |     +--rw te-dev:named-srlgs* [named-srlg]
           {te-types:named-srlg-groups}?
           |        +--rw te-dev:named-srlg    ->
           ../../../../te:globals/named-srlgs/named-srlg/name
           +--rw te-dev:threshold-type?                     enumeration
           +--rw te-dev:delta-percentage?
           rt-types:percentage
           +--rw te-dev:threshold-specification?            enumeration
           +--rw te-dev:up-thresholds*
           rt-types:percentage
           +--rw te-dev:down-thresholds*
           rt-types:percentage
           +--rw te-dev:up-down-thresholds*
           rt-types:percentage
           +--rw te-dev:switching-capabilities* [switching-capability]
           |  +--rw te-dev:switching-capability    identityref
           |  +--rw te-dev:encoding?             identityref
           +--ro te-dev:state
              +--ro te-dev:te-advertisements_state
                 +--ro te-dev:flood-interval?           uint32
                 +--ro te-dev:last-flooded-time?        uint32
                 +--ro te-dev:next-flooded-time?        uint32
                 +--ro te-dev:last-flooded-trigger?     enumeration
                 +--ro te-dev:advertized-level-areas* [level-area]
                    +--ro te-dev:level-area    uint32

  rpcs:
    +---x globals-rpc
    +---x interfaces-rpc
    +---x tunnels-rpc
       +---w input
       |  +---w tunnel-info
       |     +---w (type)?
       |        +--:(tunnel-p2p)
       |        |  +---w p2p-id?    tunnel-ref
       |        +--:(tunnel-p2mp)
       |           +---w p2mp-id?   tunnel-p2mp-ref
       +--ro output
```

```
          +--ro result
             +--ro result?   enumeration

  notifications:
    +---n globals-notif
    +---n tunnels-notif
module: ietf-te-device

  rpcs:
    +---x interfaces-rpc

  notifications:
    +---n interfaces-notif
```

Figure 3: TE generic model configuration and state tree

3. Model Organization

The TE generic YANG data module "ietf-te" covers configuration, state, RPC and notifications data pertaining to TE global parameters, interfaces, tunnels and LSPs parameters that are device independent.

The container "te" is the top level container in the data model. The presence of this container enables TE function system wide.

The model top level organization is shown below in Figure 4:

```
   module: ietf-te
      +--rw te!
         +--rw globals
            .
            .

         +--rw tunnels
            .
            .

         +-- lsps-state

   rpcs:
      +---x globals-rpc
      +---x tunnels-rpc
   notifications:
      +---n globals-notif
      +---n tunnels-notif
```

Figure 4: TE generic highlevel model view

3.1. Global Configuration and State Data

The global TE branch of the data model covers configurations that control TE features behavior system-wide, and its respective state. Examples of such configuration data are:

- Table of named SRLG mappings

- Table of named (extended) administrative groups mappings

- Table of named explicit paths to be referenced by TE tunnels

- Table of named path-constraints sets

- Auto-bandwidth global parameters

- TE diff-serve TE-class maps

- System-wide capabilities for LSP reoptimization (included in the TE device model)

  * Reoptimization timers (periodic interval, LSP installation and cleanup)

- System-wide capabilities for TE state flooding (included in the TE device model)

  * Periodic flooding interval

- Global capabilities that affect the originating, traversing and terminating LSPs. For example:

  * Path selection parameters (e.g. metric to optimize, etc.)

  * Path or segment protection parameters

3.2. Interfaces Configuration and State Data

This branch of the model covers configuration and state data corresponding to TE interfaces present on a device. The module "ietf-te-device" is introduced to hold TE device specific properties.

Examples of TE interface properties are: * Maximum reservable bandwidth, bandwidth constraints (BC) * Flooding parameters * Flooding intervals and threshold values * interface attributes * (Extended) administrative groups * SRLG values * TE metric value * Fast reroute backup tunnel properties (such as static, auto-tunnel)

The state corresponding to the TE interfaces applied configuration, protocol derived state, and stats and counters all fall under the interface "state" sub-container as shown in Figure 5 below:

```
module: ietf-te-device
  augment /te:te:
      +--rw interfaces
         .
         +-- rw te-dev:te-attributes
                <<intended configuration>>
             .
             +-- ro state
                <<derived state associated with the TE interface>>
```

Figure 5: TE interface state

This covers state data for TE interfaces such as:

- Bandwidth information: maximum bandwidth, available bandwidth at different priorities and for each class-type (CT)

- List of admitted LSPs

  - Name, bandwidth value and pool, time, priority

- Statistics: state counters, flooding counters, admission counters (accepted/rejected), preemption counters

- Adjacency information

  - Neighbor address

  - Metric value

3.3. Tunnels Configuration and State Data

This branch covers data related to TE tunnels configuration and state. Data that is device independent is defined in the TE generic YANG module "ietf-te", where as the device dependent data is defined in the device module "ietf-te-device". The derived state associated with tunnels is grouped under a state container as shown in Figure 6.

```
   module: ietf-te
      +--rw te!
         +--rw tunnels
               <<intended configuration>>
            .
            +-- ro state
               <<derived state associated with the tunnel>>
```

Figure 6: TE interface state tree

Examples of tunnel configuration data for TE tunnels:

- Name and type (e.g. P2P, P2MP) of the TE tunnel
- Administrative and operational state of the TE tunnel
- Set of primary and corresponding secondary paths and corresponding path attributes
- Bidirectional path attribute(s) including forwarding and reverse path properties
- Protection and restoration path parameters

3.3.1. Tunnel Compute-Only Mode

A configured TE tunnel, by default, is provisioned so it can carry traffic as soon as a valid path is computed and an LSP instantiated. In some cases, however, a TE tunnel may be provisioned for the only purpose of computing a path and reporting it without the need to instantiate the LSP or commit any resources. In such a case, the tunnel is configured in "compute-only" mode to distinguish it from default tunnel behavior.

A "compute-only" TE tunnel is configured as a usual TE tunnel with associated per path constraint(s) and properties on a device or controller. The device or controller computes the feasible path(s) subject to configured constraints and reflects the computed path(s) in the LSP(s) Record-Route Object (RRO) list. At any time, a client may query "on-demand" the "compute-only" TE tunnel computed path(s) properties by querying the state of the tunnel. Alternatively, the client can subscribe on the "compute-only" TE tunnel to be notified of computed path(s) and whenever it changes.

3.3.2. Tunnel Hierarchical Link Endpoint

TE LSPs can be set up in MPLS or Generalized MPLS (GMPLS) networks to be used to form links to carry traffic in in other (client) networks [RFC6107]. In this case, the model introduces the TE tunnel hierarchical link endpoint parameters to identify the specific link in the client layer that the TE tunnel is associated with.

3.4. TE LSPs State Data

TE LSPs are derived state data that is usually instantiated via signaling protocols. TE LSPs exists on routers as ingress (starting point of LSP), transit (mid-point of LSP), or egress (termination point of the LSP). TE LSPs are distinguished by the 5 tuple, and LSP type (P2P or P2MP). In the model, the nodes holding LSPs data exist in the read-only lsps-state list as show in Figure 3.

3.5. Global RPC Data

This branch of the model covers system-wide RPC execution data to trigger actions and optionally expect responses. Examples of such TE commands are to:

- Clear global TE statistics of various features

3.6. Interface RPC Data

This collection of data in the model defines TE interface RPC execution commands. Examples of these are to:

- Clear TE statistics for all or for individual TE interfaces

- Trigger immediate flooding for one or all TE interfaces

3.7. Tunnel RPC Data

This branch of the model covers TE tunnel RPC execution data to trigger actions and expect responses. The TE generic YANG data model defines target containers that an external module in [I-D.ietf-teas-yang-path-computation] augments with RPCs that allow the invocation of certain TE functions (e.g. path computations).

4. TE Generic and Helper YANG Modules

```
<CODE BEGINS> file "ietf-te@2018-10-10.yang"
module ietf-te {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te";

  /* Replace with IANA when assigned */
  prefix "te";

  /* Import TE generic types */
  import ietf-te-types {
    prefix te-types;
    reference "draft-ietf-teas-yang-te-types: A YANG Data Model for
               Common Traffic Engineering Types";
  }

  import ietf-inet-types {
    prefix inet;
    reference "RFC6991: Common YANG Data Types";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group";

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

     WG Chair: Lou Berger
               <mailto:lberger@labn.net>

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

     Editor:   Tarek Saad
               <mailto:tsaad@cisco.com>

     Editor:   Rakesh Gandhi
               <mailto:rgandhi@cisco.com>

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

     Editor:   Himanshu Shah
               <mailto:hshah@ciena.com>

     Editor:   Xufeng Liu
               <mailto:Xufeng_Liu@jabil.com>

     Editor:   Igor Bryskin
               <mailto:Igor.Bryskin@huawei.com>";
  description
    "YANG data module for TE configuration,
    state, RPC and notifications.";

  revision "2018-10-10" {
    description "Latest update to TE generic YANG module.";
    reference "TBA";
  }

  typedef tunnel-ref {
    type leafref {
      path "/te:te/te:tunnels/te:tunnel/te:name";
    }
    description
      "This type is used by data models that need to reference
       configured TE tunnel.";
  }

  typedef tunnel-p2mp-ref {
    type leafref {
      path "/te:te/te:tunnels/te:tunnel-p2mp/te:name";
    }
    description
      "This type is used by data models that need to reference
       configured P2MP TE tunnel.";
    reference "RFC4875";
  }

  typedef path-ref {
    type union {
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/" +
              "te:p2p-primary-paths/te:p2p-primary-path/te:name";
      }
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/" +
             "te:p2p-secondary-paths/te:p2p-secondary-path/te:name";
      }
    }
    description
      "This type is used by data models that need to reference
       configured primary or secondary path of a TE tunnel.";
  }

  /**
   * TE tunnel generic groupings
   */
  grouping p2p-reverse-primary-path-properties {
    description "tunnel path properties.";
    reference "RFC7551";
    container p2p-reverse-primary-path {
      description "Tunnel reverse primary path properties";
      uses p2p-path-reverse-properties_config;
      uses path-constraints-common_config;
      container state {
        config false;
        description
          "Configuration applied parameters and state";
        uses p2p-path-properties_state;
      }
      container p2p-reverse-secondary-path {
        description "Tunnel reverse secondary path properties";
        uses p2p-reverse-path-candidate-secondary-path-config;
      }
    }
  }

  grouping p2p-secondary-path-properties {
    description "tunnel path properties.";
    uses p2p-path-properties_config;
    uses path-constraints-common_config;
    uses protection-restoration-params_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      uses p2p-path-properties_state;
    }
  }

  grouping p2p-primary-path-properties {
    description
      "TE tunnel primary path properties grouping";
    uses p2p-path-properties_config;
    uses path-constraints-common_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      uses p2p-path-properties_state;
    }
  }

  grouping path-properties_state {
    description "Computed path properties grouping";
    leaf metric-type {
      type identityref {
        base te-types:path-metric-type;
      }
      description "TE path metric type";
    }
    leaf accumulative-value {
      type uint64;
      description "TE path metric accumulative value";
    }
  }

  grouping path-properties {
    description "TE computed path properties grouping";
    container path-properties {
      description "The TE path computed properties";
      list path-metric {
        key metric-type;
        description "TE path metric type";
        leaf metric-type {
          type leafref {
            path "../state/metric-type";
          }
          description "TE path metric type";
        }
        container state {
          config false;
          description
            "Configuration applied parameters and state";
          uses path-properties_state;
        }
      }
      uses te-types:generic-path-affinities;
      uses te-types:generic-path-srlgs;
      container path-route-objects {
        description
          "Container for the list of computed route objects
           as returned by the computation engine";
        list path-computed-route-object {
          key index;
          description
            "List of computed route objects returned by the
             computation engine";
          leaf index {
            type leafref {
              path "../state/index";
            }
            description "Index of computed route object";
          }
          container state {
            config false;
            description
              "Configuration applied parameters and state";
            leaf index {
              type uint32;
              description "ERO subobject index";
            }
            uses te-types:explicit-route-hop;
          }
        }
      }
      uses shared-resources-tunnels;
    }
  }

  grouping p2p-path-properties_state {
    description "TE per path state parameters";
    container computed-paths-properties {
      description "Computed path properties container";
      list computed-path-properties {
        key k-index;
        description "List of computed paths";
        leaf k-index {
          type uint8;
          description
            "The k-th path returned from the computation server.";
        }
        uses path-properties {
          description "The TE path computed properties";
        }
      }
    }
    container lsps {
      description "TE LSPs container";
      list lsp {
        key
          "source destination tunnel-id lsp-id "+
          "extended-tunnel-id";
        description "List of LSPs associated with the tunnel.";
        uses lsp-properties_state;
        uses shared-resources-tunnels_state;
        uses lsp-record-route-information_state;
        uses path-properties {
          description "The TE path actual properties";
        }
      }
    }
  }

  grouping p2p-path-properties-common_config {
    description
      "TE tunnel common path properties configuration grouping";
    leaf name {
      type string;
      description "TE path name";
    }
    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "Signaling protocol used to set up this tunnel";
    }
    leaf path-computation-method {
      type identityref {
        base te-types:path-computation-method;
      }
      default te-types:path-locally-computed;
      description
        "The method used for computing the path, either
        locally computed, queried from a server or not
        computed at all (explicitly configured).";
    }
    leaf path-computation-server {
      when "../path-computation-method = "+
      "'te-types:path-externally-queried'" {
        description
          "The path-computation server when the path is
           externally queried";
      }
      type inet:ip-address;
      description
        "Address of the external path computation
         server";
    }
    leaf compute-only {
      type empty;
      description
        "When set, the path is computed and updated whenever
         the topology is updated. No resources are committed
         or reserved in the network.";
    }
    leaf use-path-computation {
      when "../path-computation-method =" +
      " 'te-types:path-locally-computed'";
```

```
        type boolean;
        description "A CSPF dynamically computed path";
      }
      leaf lockdown {
        type empty;
        description
          "Indicates no reoptimization to be attempted for
           this path.";
      }
      leaf path-scope {
        type identityref {
          base te-types:path-scope-type;
        }
        default te-types:path-scope-end-to-end;
        description "Path scope if segment or an end-to-end path";
      }
    }

    grouping p2p-path-reverse-properties_config {
      description
        "TE tunnel reverse path properties configuration
         grouping";
      uses p2p-path-properties-common_config;
      uses te-types:generic-path-optimization;
      leaf named-path-constraint {
        if-feature te-types:named-path-constraints;
        type leafref {
          path "../../../../../../globals/"
          + "named-path-constraints/named-path-constraint/"
          + "name";
        }
        description
          "Reference to a globally defined named path
          constraint set";
      }
    }

    grouping p2p-path-properties_config {
      description
        "TE tunnel path properties configuration grouping";
      uses p2p-path-properties-common_config;
      uses te-types:generic-path-optimization;
      leaf preference {
        type uint8 {
          range "1..255";
        }
        description
          "Specifies a preference for this path. The lower the
          number higher the preference";
      }
      leaf k-requested-paths {
        type uint8;
        description
          "The number of k-shortest-paths  requested from the path
           computation server and returned sorted by its optimization
           objective";
      }
      leaf named-path-constraint {
        if-feature te-types:named-path-constraints;
        type leafref {
          path "../../../../../globals/"
          + "named-path-constraints/named-path-constraint/"
          + "name";
        }
        description
          "Reference to a globally defined named path
          constraint set";
      }
    }

    /* TE tunnel configuration data */
    grouping tunnel-p2mp-params_config {
      description
        "Configuration parameters relating to TE tunnel";
      leaf name {
        type string;
        description "TE tunnel name.";
      }
      leaf identifier {
        type uint16;
        description
          "TE tunnel Identifier.";
        reference "RFC 3209";
      }
      leaf description {
        type string;
        description
          "Textual description for this TE tunnel";
      }
    }

    grouping hierarchical-link_config {
      description
        "Hierarchical link configuration grouping";
      reference "RFC4206";
      leaf local-te-node-id {
        type te-types:te-node-id;
        description
          "Local TE node identifier";
      }
      leaf local-te-link-tp-id {
        type te-types:te-tp-id;
        description
          "Local TE link termination point identifier";
      }
      leaf remote-te-node-id {
        type te-types:te-node-id;
        description
          "Remote TE node identifier";
      }
      uses te-types:te-topology-identifier;
    }

    grouping hierarchical-link {
      description
        "Hierarchical link grouping";
      reference "RFC4206";
      container hierarchical-link {
        description
          "Identifies a hierarchical link (in client layer)
           that this tunnel is associated with.";
        uses hierarchical-link_config;
      }
    }

    grouping protection-restoration-params_state {
      description
        "Protection parameters grouping";
      leaf lockout-of-normal {
        type boolean;
        description
          "
            When set to ’True’, it represents a lockout of normal
            traffic external command. When set to ’False’, it
            represents a clear lockout of normal traffic external
            command. The lockout of normal traffic command applies
            to this Tunnel.
          ";
        reference
          "ITU-T G.808, RFC 4427";
      }
      leaf freeze {
        type boolean;
        description
```

```
        "
          When set to ’True’, it represents a freeze external
          command. When set to ’False’, it represents a clear
          freeze external command. The freeze command command
          applies to all the Tunnels which are sharing the
          protection resources with this Tunnel.
        ";
      reference
        "ITU-T G.808, RFC 4427";
    }
    leaf lsp-protection-role {
      type enumeration {
        enum working {
          description
            "A working LSP must be a primary LSP whilst a protecting
             LSP can be either a primary or a secondary LSP. Also,
             known as protected LSPs when working LSPs are associated
             with protecting LSPs.";
        }
        enum protecting {
          description
            "A secondary LSP is an LSP that has been provisioned
             in the control plane only; e.g. resource allocation
             has not been committed at the data plane";
        }
      }
      description "LSP role type";
      reference "rfc4872, section 4.2.1";
    }

    leaf lsp-protection-state {
      type identityref {
        base te-types:lsp-protection-state;
      }
      description
        "The state of the APS state machine controlling which
         tunnels is using the resources of the protecting LSP.";
    }
    leaf protection-group-ingress-node-id {
      type te-types:te-node-id;
      description
        "Indicates the te-node-id of the protection group
        ingress node when the APS state represents an extenal
        command (LoP, SF, MS) applied to it or a WTR timer
        running on it. If the external command is not applied to
        the ingress node or the WTR timer is not running on it,
        this attribute is not specified. If value 0.0.0.0 is used
        when the te-node-id of the protection group ingress node is
        unknown (e.g., because the ingress node is outside the scope
        of control of the server)";
    }
    leaf protection-group-egress-node-id {
      type te-types:te-node-id;
      description
        "Indicates the te-node-id of the protection group egress node
        when the APS state represents an extenal command (LoP, SF,
        MS) applied to it or a WTR timer running on it. If the
        external command is not applied to the ingress node or
        the WTR timer is not running on it, this attribute is not
        specified. If value 0.0.0.0 is used when the te-node-id of
        the protection group ingress node is unknown (e.g., because
        the ingress node is outside the scope of control of the
        server)";
    }
  }

  grouping protection-restoration-params_config {
    description "Protection and restoration parameters";
    container protection {
      description "Protection parameters";
      leaf enable {
        type boolean;
        default 'false';
        description
          "A flag to specify if LSP protection is enabled";
        reference "rfc4427";
      }
      leaf protection-type {
        type identityref {
          base te-types:lsp-protection-type;
        }
        description "LSP protection type.";
      }
      leaf protection-reversion-disable {
        type boolean;
        description "Disable protection reversion to working path";
      }
      leaf hold-off-time {
        type uint32;
        units "milli-seconds";
        default 0;
        description
          "The time between the declaration of an SF or SD condition
           and the initialization of the protection switching
           algorithm.";
        reference "rfc4427";
      }
      leaf wait-to-revert {
        type uint16;
        units seconds;
        description
         "Time to wait before attempting LSP reversion";
        reference "rfc4427";
      }
      leaf aps-signal-id {
        type uint8 {
          range "1..255";
        }
        description
          "The APS signal number used to reference the traffic of this
           tunnel. The default value for normal traffic is 1.
           The default value for extra-traffic is 255. If not specified,
           non-default values can be assigned by the server,
           if and only if, the server controls both endpoints.";
        reference
          "ITU-T G.808.1";
      }
    }
    container restoration {
      description "Restoration parameters";
      leaf enable {
        type boolean;
        default ’false’;
        description
          "A flag to specify if LSP restoration is enabled";
        reference "rfc4427";
      }
      leaf restoration-type {
        type identityref {
          base te-types:lsp-restoration-type;
        }
        description "LSP restoration type.";
      }
      leaf restoration-scheme {
        type identityref {
          base te-types:restoration-scheme-type;
        }
        description "LSP restoration scheme.";
      }
      leaf restoration-reversion-disable {
        type boolean;
        description "Disable restoration reversion to working path";
      }
      leaf hold-off-time {
        type uint32;
        units "milli-seconds";
        description
          "The time between the declaration of an SF or SD condition
           and the initialization of the protection switching
           algorithm.";
        reference "rfc4427";
      }
      leaf wait-to-restore {
        type uint16;
        units seconds;
        description
         "Time to wait before attempting LSP restoration";
        reference "rfc4427";
      }
      leaf wait-to-revert {
        type uint16;
        units seconds;
        description
         "Time to wait before attempting LSP reversion";
        reference "rfc4427";
      }
    }
  }

  grouping p2p-dependency-tunnels_config {
    description
      "Groupong for tunnel dependency list of tunnels";
    container dependency-tunnels {
      description "Dependency tunnels list";
      list dependency-tunnel {
        key "name";
        description "Dependency tunnel entry";
        leaf name {
          type leafref {
            path "../../../../../tunnels/tunnel/name";
            require-instance false;
          }
          description "Dependency tunnel name";
        }
        leaf encoding {
          type identityref {
            base te-types:lsp-encoding-types;
          }
          description "LSP encoding type";
          reference "RFC3945";
        }
        leaf switching-type {
          type identityref {
            base te-types:switching-capabilities;
          }
          description "LSP switching type";
          reference "RFC3945";
        }
      }
    }
  }

  grouping tunnel-p2p-params_config {
    description
      "Configuration parameters relating to TE tunnel";
    leaf name {
      type string;
      description "TE tunnel name.";
    }
    leaf identifier {
      type uint16;
      description
        "TE tunnel Identifier.";
      reference "RFC3209";
    }
    leaf description {
      type string;
      description
        "Textual description for this TE tunnel";
    }
    leaf encoding {
      type identityref {
        base te-types:lsp-encoding-types;
      }
      description "LSP encoding type";
      reference "RFC3945";
    }
    leaf switching-type {
      type identityref {
        base te-types:switching-capabilities;
      }
      description "LSP switching type";
      reference "RFC3945";
    }
    leaf provisioning-state {
      type identityref {
        base te-types:tunnel-state-type;
      }
      default te-types:tunnel-state-up;
      description "TE tunnel administrative state.";
    }
    leaf preference {
      type uint8 {
        range "1..255";
      }
      description
        "Specifies a preference for this tunnel.
        A lower number signifies a better preference";
    }
    leaf reoptimize-timer {
      type uint16;
      units seconds;
      description
       "frequency of reoptimization of
        a traffic engineered LSP";
    }
    leaf source {
      type te-types:te-node-id;
      description
        "TE tunnel source node ID.";
    }
    leaf destination {
      type te-types:te-node-id;
      description
        "TE tunnel destination node ID";
    }
    leaf src-tp-id {
      type binary;
      description
        "TE tunnel source termination point identifier.";
    }
    leaf dst-tp-id {
      type binary;
      description
        "TE tunnel destination termination point identifier.";
    }
    leaf bidirectional {
      type boolean;
      default ’false’;
      description "TE tunnel bidirectional";
    }
    uses tunnel-p2p-associations_config;
    uses protection-restoration-params_config;
    uses te-types:tunnel-constraints_config;
    uses p2p-dependency-tunnels_config;
    uses hierarchical-link;
  }
  grouping tunnel-p2p-associations_config {
    description "TE tunnel association grouping";
    container association-objects {
      description "TE tunnel associations";
      list association-object {
        key "type ID source global-source";
        description "List of association base objects";
        reference "RFC4872";
        leaf type {
          type identityref {
            base te-types:association-type;
          }
          description "Association type";
          reference "RFC4872";
        }
        leaf ID {
          type uint16;
          description "Association ID";
          reference "RFC4872";
        }
        leaf source {
          type inet:ip-address;
          description "Association source";
          reference "RFC4872";
        }
        leaf global-source {
          type inet:ip-address;
          description "Association global source";
          reference "RFC4872";
        }
      }
      list association-object-extended {
        key "type ID source global-source extended-ID";
        description "List of extended association objects";
        reference "RFC6780";
        leaf type {
          type identityref {
            base te-types:association-type;
          }
          description "Association type";
        }
        leaf ID {
          type uint16;
          description "Association ID";
          reference "RFC4872";
        }
        leaf source {
          type inet:ip-address;
          description "Association source";
        }
        leaf global-source {
          type inet:ip-address;
          description "Association global source";
          reference "RFC4872";
        }
        leaf extended-ID {
          type binary;
          description "Association extended ID";
          reference "RFC4872";
        }
      }
    }
  }

  grouping tunnel-p2p-params_state {
    description
      "State parameters relating to TE tunnel";
    leaf operational-state {
      type identityref {
        base te-types:tunnel-state-type;
      }
      default te-types:tunnel-state-up;
      description "TE tunnel administrative state.";
    }
  }

  grouping path-access-segment-info {
    description
      "If an end-to-end tunnel crosses multiple domains using
       the same technology, some additional constraints have to be
       taken in consideration in each domain";
    container path-in-segment {
      presence
        "The end-to-end tunnel starts in a previous domain;
         this tunnel is a segment in the current domain.";
      description
        "This tunnel is a segment that needs to be coordinated
         with previous segment stitched on head-end side.";
      uses te-types:label-set-info;
    }
    container path-out-segment {
      presence
        "The end-to-end tunnel is not terminated in this domain;
         this tunnel is a segment in the current domain.";
      description
        "This tunnel is a segment that needs to be coordinated
```

```
         with previous segment stitched on head-end side.";
      uses te-types:label-set-info;
    }
  }

  /* TE tunnel configuration/state grouping */
  grouping tunnel-p2mp-properties {
    description
      "Top level grouping for P2MP tunnel properties.";
    uses tunnel-p2mp-params_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      leaf operational-state {
        type identityref {
          base te-types:tunnel-state-type;
        }
        default te-types:tunnel-state-up;
        description "TE tunnel administrative state.";
      }
    }
  }

  grouping p2p-path-candidate-secondary-path-config {
    description
      "Configuration parameters relating to a secondary path which
      is a candidate for a particular primary path";

    leaf secondary-path {
      type leafref {
        path "../../../../../p2p-secondary-paths/" +
             "p2p-secondary-path/name";
      }
      description
        "A reference to the secondary path that should be utilised
        when the containing primary path option is in use";
    }

    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "Signaling protocol used to set up this tunnel";
    }
  }
  grouping p2p-reverse-path-candidate-secondary-path-config {
    description
      "Configuration parameters relating to a secondary path which
      is a candidate for a particular primary path";

    leaf secondary-path {
      type leafref {
        path "../../../../../p2p-secondary-paths/" +
             "p2p-secondary-path/name";
      }
      description
        "A reference to the secondary path that should be utilised
        when the containing primary path option is in use";
    }

    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "Signaling protocol used to set up this tunnel";
    }
  }

  grouping p2p-path-candidate-secondary-path-state {
    description
      "Operational state parameters relating to a secondary path
      which is a candidate for a particular primary path";

    leaf active {
      type boolean;
      description
        "Indicates the current active path option that has
        been selected of the candidate secondary paths";
    }
  }

  grouping tunnel-p2p-properties {
    description
      "Top level grouping for tunnel properties.";
    uses tunnel-p2p-params_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      uses tunnel-p2p-params_state;
    }
    container p2p-primary-paths {
      description "Set of P2P primary aths container";
      list p2p-primary-path {
        key "name";
        description
          "List of primary paths for this tunnel.";
        uses p2p-primary-path-properties;
        uses p2p-reverse-primary-path-properties;
        container candidate-p2p-secondary-paths {
          description
            "The set of candidate secondary paths which may be used
            for this primary path. When secondary paths are specified
            in the list the path of the secondary LSP in use must be
            restricted to those path options referenced. The
            priority of the secondary paths is specified within the
            list. Higher priority values are less preferred - that is
            to say that a path with priority 0 is the most preferred
            path. In the case that the list is empty, any secondary
            path option may be utilised when the current primary path
            is in use.";
          list candidate-p2p-secondary-path {
            key "secondary-path";
            description
              "List of secondary paths for this tunnel.";
            uses p2p-path-candidate-secondary-path-config;

            container state {
              config false;
              description
                "Configuration applied parameters and state";
              uses p2p-path-candidate-secondary-path-state;
            }
          }
        }
      }
    }
    container p2p-secondary-paths {
      description "Set of P2P secondary paths container";
      list p2p-secondary-path {
        key "name";
        description
          "List of secondary paths for this tunnel.";
        uses p2p-secondary-path-properties;
      }
    }
  }

  grouping shared-resources-tunnels_state {
    description
```

```
      "The specific tunnel that is using the shared secondary path
       resources";
    leaf lsp-shared-resources-tunnel {
      type tunnel-ref;
      description
        "Reference to the tunnel that sharing secondary path
        resources with this tunnel";
    }
  }
  grouping shared-resources-tunnels {
    description
      "Set of tunnels that share secondary path resources with
      this tunnnel";
    container shared-resources-tunnels {
      description
        "Set of tunnels that share secondary path resources with
        this tunnnel";
      leaf-list lsp-shared-resources-tunnel {
        type tunnel-ref;
        description
          "Reference to the tunnel that sharing secondary path
          resources with this tunnel";
      }
    }
  }

  grouping tunnel-actions {
    description "Tunnel actions";
    action tunnel-action {
      description "Tunnel action";
      input {
        leaf action-type {
          type identityref {
            base te-types:tunnel-action-type;
          }
          description "Tunnel action type";
        }
      }
      output {
        leaf action-result {
          type identityref {
            base te-types:te-action-result;
          }
          description "The result of the RPC operation";
        }
      }
    }
  }
  grouping tunnel-protection-actions {
    description
      "Protection external command actions";
    action protection-external-commands {
      input {
        leaf protection-external-command {
          type identityref {
            base te-types:protection-external-commands;
          }
          description
            "Protection external command";
        }
        leaf protection-group-ingress-node-id {
          type te-types:te-node-id;
          description
            "When specified, indicates whether the action is
             applied on ingress node.
             By default, if neither ingress nor egress node-id
             is set, the the action applies to ingress node only.";
        }
        leaf protection-group-egress-node-id {
          type te-types:te-node-id;
          description
            "When specified, indicates whether the action is
             applied on egress node.
             By default, if neither ingress nor egress node-id
             is set, the the action applies to ingress node only.";
        }
        leaf path-ref {
          type path-ref;
          description
            "Indicates to which path the external command applies to.";
        }
        leaf traffic-type {
          type enumeration {
            enum normal-traffic {
              description
                "The manual-switch or forced-switch command applies to
                 the normal traffic (this Tunnel).";
            }
            enum null-traffic {
              description
                "The manual-switch or forced-switch command applies to
                 the null traffic.";
            }
            enum extra-traffic {
              description
                "The manual-switch or forced-switch command applies to
```

```
                 the extra traffic (the extra-traffic Tunnel sharing
                 protection bandwidth with this Tunnel).";
            }
          }
          description
            "Indicates whether the manual-switch or forced-switch
             commands applies to the normal traffic, the null traffic
             or the extra-traffic.";
          reference
            "ITU-T G.808, RFC 4427";
        }
        leaf extra-traffic-tunnel-ref {
          type tunnel-ref;
          description
            "In case there are multiple extra-traffic tunnels sharing
             protection bandwidth with this Tunnel (m:n protection),
             represents which extra-traffic Tunnel the manual-switch or
             forced-switch to extra-traffic command applies to.";
        }
      }
    }
  }

  /*** End of TE tunnel groupings ***/

  /**
   * LSP related generic groupings
   */
  grouping lsp-record-route-information_state {
    description "recorded route information grouping";
    container lsp-record-route-subobjects {
      description "RSVP recorded route object information";
      list record-route-subobject {
        when "../../origin-type = ’ingress’" {
          description "Applicable on non-ingress LSPs only";
        }
        key "index";
        description "Record route sub-object list";
        uses te-types:record-route-subobject_state;
      }
    }
  }

  grouping lsps-state-grouping {
    description
      "LSPs state operational data grouping";
    container lsps-state {
      config false;
      description "TE LSPs state container";
      list lsp {
        key
          "source destination tunnel-id lsp-id "+
          "extended-tunnel-id";
        description "List of LSPs associated with the tunnel.";
        uses lsp-properties_state;
        uses lsp-record-route-information_state;
      }
    }
  }

  /*** End of TE LSP groupings ***/

  /**
   * TE global generic groupings
   */

  /* Global named admin-groups configuration data */
  grouping named-admin-groups_config {
    description
      "Global named administrative groups configuration
      grouping";
    leaf name {
      type string;
      description
        "A string name that uniquely identifies a TE
        interface named admin-group";
    }
    leaf bit-position {
      type uint32;
      description
        "Bit position representing the administrative group";
      reference "RFC3209 and RFC7308";
    }
  }
  grouping named-admin-groups {
    description
      "Global named administrative groups configuration
      grouping";
    container named-admin-groups {
      description "TE named admin groups container";
      list named-admin-group {
        if-feature te-types:extended-admin-groups;
        if-feature te-types:named-extended-admin-groups;
        key "name";
        description
          "List of named TE admin-groups";
        uses named-admin-groups_config;
      }
    }
  }

  /* Global named admin-srlgs configuration data */
  grouping named-srlgs_config {
    description
      "Global named SRLGs configuration grouping";
    leaf name {
      type string;
      description
        "A string name that uniquely identifies a TE
        interface named srlg";
    }
    leaf group {
      type te-types:srlg;
      description "An SRLG value";
    }
    leaf cost {
      type uint32;
      description
        "SRLG associated cost. Used during path to append
         the path cost when traversing a link with this SRLG";
    }
  }

  grouping named-srlgs {
    description
      "Global named SRLGs configuration grouping";
    container named-srlgs {
      description "TE named SRLGs container";
      list named-srlg {
        if-feature te-types:named-srlg-groups;
        key "name";
        description
          "A list of named SRLG groups";
        uses named-srlgs_config;
      }
    }
  }

  /* Global named paths constraints configuration data */
  grouping path-constraints_state {
    description
      "TE path constraints state";
    leaf bandwidth-generic_state {
      type te-types:te-bandwidth;
      description
        "A technology agnostic requested bandwidth to use
         for path computation";
    }
    leaf disjointness_state {
      type te-types:te-path-disjointness;
      description
        "The type of resource disjointness.";
    }
  }

  grouping path-constraints-common_config {
    description
      "Global named path constraints configuration
      grouping";
    uses te-types:common-path-constraints-attributes;
    uses te-types:generic-path-disjointness;
    uses te-types:path-route-objects;
    uses shared-resources-tunnels {
      description
        "Set of tunnels that are allowed to share secondary path
         resources of this tunnel";
    }
    uses path-access-segment-info {
      description
        "Tunnel constraints induced by other segments.";
    }
  }

  grouping path-constraints {
    description "Per path constraints";
    uses path-constraints-common_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      uses path-constraints_state;
    }
  }

  grouping named-path-constraints {
    description
      "Global named path constraints configuration
      grouping";
    container named-path-constraints {
      description "TE named path constraints container";
      list named-path-constraint {
        if-feature te-types:named-path-constraints;
        key "name";
        leaf name {
          type string;
          description
            "A string name that uniquely identifies a
            path constraint set";
        }
        uses path-constraints;
        description
          "A list of named path constraints";
      }
    }
  }

  /* TE globals container data */
  grouping globals-grouping {
    description
      "Globals TE system-wide configuration data grouping";
    container globals {
      description
        "Globals TE system-wide configuration data container";
      uses named-admin-groups;
      uses named-srlgs;
      uses named-path-constraints;
    }
  }

  /* TE tunnels container data */
  grouping tunnels-grouping {
    description
      "Tunnels TE configuration data grouping";
    container tunnels {
      description
        "Tunnels TE configuration data container";

      list tunnel {
        key "name";
        description "P2P TE tunnels list.";
        uses tunnel-p2p-properties;
        uses tunnel-actions;
        uses tunnel-protection-actions;
      }
      list tunnel-p2mp {
        key "name";
        unique "identifier";
        description "P2MP TE tunnels list.";
        uses tunnel-p2mp-properties;
      }
    }
  }

  /* TE LSPs ephemeral state container data */
  grouping lsp-properties_state {
    description
      "LSPs state operational data grouping";
    leaf source {
      type inet:ip-address;
      description
        "Tunnel sender address extracted from
        SENDER_TEMPLATE  object";
      reference "RFC3209";
    }
    leaf destination {
      type inet:ip-address;
      description
        "Tunnel endpoint address extracted from
        SESSION object";
      reference "RFC3209";
    }
    leaf tunnel-id {
      type uint16;
      description
        "Tunnel identifier used in the SESSION
        that remains constant over the life
        of the tunnel.";
      reference "RFC3209";
    }
    leaf lsp-id {
      type uint16;
      description
        "Identifier used in the SENDER_TEMPLATE
        and the FILTER_SPEC that can be changed
        to allow a sender to share resources with
        itself.";
      reference "RFC3209";
    }
    leaf extended-tunnel-id {
      type inet:ip-address;
       description
        "Extended Tunnel ID of the LSP.";
      reference "RFC3209";
    }
    leaf operational-state {
      type identityref {
        base te-types:lsp-state-type;
      }
      description "LSP operational state.";
    }
    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "Signaling protocol used to set up this tunnel";
    }
    leaf origin-type {
      type enumeration {
        enum ingress {
          description
            "Origin ingress";
        }
        enum egress {
          description
            "Origin egress";
        }
        enum transit {
          description
            "transit";
        }
      }
      description
        "Origin type of LSP relative to the location
        of the local switch in the path.";
    }

    leaf lsp-resource-status {
      type enumeration {
        enum primary {
          description
            "A primary LSP is a fully established LSP for
             which the resource allocation has been committed
             at the data plane";
        }
        enum secondary {
          description
            "A secondary LSP is an LSP that has been provisioned
             in the control plane only; e.g. resource allocation
             has not been committed at the data plane";
        }
      }
      description "LSP resource allocation type";
      reference "rfc4872, section 4.2.1";
    }
```

```
    uses protection-restoration-params_state;
  }
  /*** End of TE global groupings ***/

  /**
   * TE configurations container
   */
  container te {
    presence "Enable TE feature.";
    description
       "TE global container.";

    /* TE Global Configuration Data */
    uses globals-grouping;

    /* TE Tunnel Configuration Data */
    uses tunnels-grouping;

    /* TE LSPs State Data */
    uses lsps-state-grouping;

  }

  /* TE Global RPCs/execution Data */
  rpc globals-rpc {
    description
      "Execution data for TE global.";
  }

  /* TE interfaces RPCs/execution Data */
  rpc interfaces-rpc {
    description
      "Execution data for TE interfaces.";
  }

  /* TE Tunnel RPCs/execution Data */
  rpc tunnels-rpc {
    description "TE tunnels RPC nodes";
    input {
      container tunnel-info {
        description "Tunnel Identification";
        choice type {
          description "Tunnel information type";
          case tunnel-p2p {
            leaf p2p-id {
              type tunnel-ref;
              description "P2P TE tunnel";
            }
          }
          case tunnel-p2mp {
            leaf p2mp-id {
              type tunnel-p2mp-ref;
              description "P2MP TE tunnel";
            }
          }
        }
      }
    }
    output {
      container result {
        description
          "The container result of the RPC operation";
        leaf result {
          type enumeration {
            enum success {
              description "Origin ingress";
            }
            enum in-progress {
              description "Origin egress";
            }
            enum fail {
              description "transit";
            }
          }
          description "The result of the RPC operation";
        }
      }
    }
  }

  /* TE Global Notification Data */
  notification globals-notif {
    description
      "Notification messages for Global TE.";
  }

  /* TE Tunnel Notification Data */
  notification tunnels-notif {
    description
      "Notification messages for TE tunnels.";
  }
}
<CODE ENDS>
```

Figure 7: TE generic YANG module

```
   <CODE BEGINS> file "ietf-te-device@2018-10-10.yang"
   module ietf-te-device {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-te-device";

     /* Replace with IANA when assigned */
     prefix "te-dev";

     /* Import TE generic types */
     import ietf-te {
       prefix te;
       reference "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
                  Engineering Tunnels and Interfaces";
     }

     /* Import TE generic types */
     import ietf-te-types {
       prefix te-types;
       reference "draft-ietf-teas-yang-te-types: A YANG Data Model for
                  Common Traffic Engineering Types";
     }

     import ietf-interfaces {
       prefix if;
       reference "RFC7223: A YANG Data Model for Interface Management";
     }

     import ietf-inet-types {
       prefix inet;
       reference "RFC6991: Common YANG Data Types";
     }

     import ietf-routing-types {
       prefix "rt-types";
       reference "RFC6991: Common YANG Data Types";
     }

     organization
       "IETF Traffic Engineering Architecture and Signaling (TEAS)
        Working Group";

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

        WG Chair: Lou Berger
                  <mailto:lberger@labn.net>
        WG Chair: Vishnu Pavan Beeram
                  <mailto:vbeeram@juniper.net>

        Editor:   Tarek Saad
                  <mailto:tsaad@cisco.com>

        Editor:   Rakesh Gandhi
                  <mailto:rgandhi@cisco.com>

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

        Editor:   Himanshu Shah
                  <mailto:hshah@ciena.com>

        Editor:   Xufeng Liu
                  <mailto:xufeng.liu@ericsson.com>

        Editor:   Xia Chen
                  <mailto:jescia.chenxia@huawei.com>

        Editor:   Raqib Jones
                  <mailto:raqib@Brocade.com>

        Editor:   Bin Wen
                  <mailto:Bin_Wen@cable.comcast.com>";

     description
       "YANG data module for TE device configurations,
       state, RPC and notifications.";

     revision "2018-10-10" {
       description "Latest update to TE device YANG module.";
       reference "TBA";
     }


     /**
      * TE LSP device state grouping
      */
     grouping lsps-device_state {
       description "TE LSP device state grouping";
       container lsp-timers {
         when "../te:origin-type = ’ingress’" {
           description "Applicable to ingress LSPs only";
         }
         description "Ingress LSP timers";
         leaf life-time {
           type uint32;
           units seconds;
           description
             "lsp life time";
         }

         leaf time-to-install {
           type uint32;
           units seconds;
           description
             "lsp installation delay time";
         }

         leaf time-to-destroy {
           type uint32;
           units seconds;
           description
             "lsp expiration delay time";
         }
       }

       container downstream-info {
         when "../te:origin-type != ’egress’" {
           description "Applicable to ingress LSPs only";
         }
         description
           "downstream information";

         leaf nhop {
           type inet:ip-address;
           description
             "downstream nexthop.";
         }

         leaf outgoing-interface {
           type if:interface-ref;
           description
             "downstream interface.";
         }

         leaf neighbor {
           type inet:ip-address;
           description
             "downstream neighbor.";
         }

         leaf label {
           type rt-types:generalized-label;
```

```
           description
             "downstream label.";
         }
       }

       container upstream-info {
         when "../te:origin-type != ’ingress’" {
           description "Applicable to non-ingress LSPs only";
         }
         description
           "upstream information";

         leaf phop {
           type inet:ip-address;
           description
             "upstream nexthop or previous-hop.";
         }

         leaf neighbor {
           type inet:ip-address;
           description
             "upstream neighbor.";
         }

         leaf label {
           type rt-types:generalized-label;
           description
             "upstream label.";
         }
       }
     }

     /**
      * Device general groupings.
      */
     grouping tunnel-device_config {
       description "Device TE tunnel configs";
       leaf path-invalidation-action {
         type identityref {
           base te-types:path-invalidation-action-type;
         }
         description "Tunnel path invalidtion action";
       }
     }

     grouping lsp-device-timers_config {
       description "Device TE LSP timers configs";
       leaf lsp-install-interval {
         type uint32;
         units seconds;
         description
           "lsp installation delay time";
       }
       leaf lsp-cleanup-interval {
         type uint32;
         units seconds;
         description
           "lsp cleanup delay time";
       }
       leaf lsp-invalidation-interval {
         type uint32;
         units seconds;
         description
           "lsp path invalidation before taking action delay time";
       }
     }
     grouping lsp-device-timers {
       description "TE LSP timers configuration";
       uses lsp-device-timers_config;
     }

     /**
      * TE global device generic groupings
      */

     /* TE interface container data */
     grouping interfaces-grouping {
       description
         "Interface TE configuration data grouping";
       container interfaces {
         description
           "Configuration data model for TE interfaces.";
         uses te-all-attributes;
         list interface {
           key "interface";
           description "TE interfaces.";
           leaf interface {
             type if:interface-ref;
             description
               "TE interface name.";
           }
           /* TE interface parameters */
           uses te-attributes;
         }
       }
     }
     /**
      * TE interface device generic groupings
      */
     grouping te-admin-groups_config {
       description
         "TE interface affinities grouping";
       choice admin-group-type {
         description
           "TE interface administrative groups
           representation type";
         case value-admin-groups {
           choice value-admin-group-type {
             description "choice of admin-groups";
             case admin-groups {
               description
                 "Administrative group/Resource
                 class/Color.";
               leaf admin-group {
                 type te-types:admin-group;
                 description
                   "TE interface administrative group";
               }
             }
             case extended-admin-groups {
               if-feature te-types:extended-admin-groups;
               description
                 "Extended administrative group/Resource
                 class/Color.";
               leaf extended-admin-group {
                 type te-types:extended-admin-group;
                 description
                   "TE interface extended administrativei
                   group";
               }
             }
           }
         }
         case named-admin-groups {
           list named-admin-groups {
             if-feature te-types:extended-admin-groups;
             if-feature te-types:named-extended-admin-groups;
             key named-admin-group;
             description
               "A list of named admin-group entries";
             leaf named-admin-group {
               type leafref {
                 path "../../../../te:globals/" +
                   "te:named-admin-groups/te:named-admin-group/" +
                   "te:name";
               }
               description "A named admin-group entry";
             }
           }
         }
       }
     }

     /* TE interface SRLGs */
     grouping te-srlgs_config {
       description "TE interface SRLG grouping";
       choice srlg-type {
         description "Choice of SRLG configuration";
         case value-srlgs {
           list values {
             key "value";
             description "List of SRLG values that
               this link is part of.";
             leaf value {
               type uint32 {
                 range "0..4294967295";
               }
               description
                 "Value of the SRLG";
             }
           }
         }
         case named-srlgs {
           list named-srlgs {
             if-feature te-types:named-srlg-groups;
             key named-srlg;
             description
               "A list of named SRLG entries";
             leaf named-srlg {
               type leafref {
                 path "../../../../te:globals/" +
                   "te:named-srlgs/te:named-srlg/te:name";
               }
               description
                 "A named SRLG entry";
             }
           }
         }
       }
     }

     grouping te-igp-flooding-bandwidth_config {
       description
         "Configurable items for igp flooding bandwidth
         threshold configuration.";
       leaf threshold-type {
         type enumeration {
           enum DELTA {
             description
               "DELTA indicates that the local
               system should flood IGP updates when a
               change in reserved bandwidth >= the specified
               delta occurs on the interface.";
           }
           enum THRESHOLD_CROSSED {
             description
               "THRESHOLD-CROSSED indicates that
               the local system should trigger an update (and
               hence flood) the reserved bandwidth when the
               reserved bandwidth changes such that it crosses,
               or becomes equal to one of the threshold values.";
           }
         }
         description
           "The type of threshold that should be used to specify the
           values at which bandwidth is flooded. DELTA indicates that
           the local system should flood IGP updates when a change in
           reserved bandwidth >= the specified delta occurs on the
           interface. Where THRESHOLD_CROSSED is specified, the local
           system should trigger an update (and hence flood) the
           reserved bandwidth when the reserved bandwidth changes such
           that it crosses, or becomes equal to one of the threshold
           values";
       }

       leaf delta-percentage {
         when "../threshold-type = 'DELTA'" {
           description
             "The percentage delta can only be specified when the
             threshold type is specified to be a percentage delta of
             the reserved bandwidth";
         }
         type rt-types:percentage;
         description
           "The percentage of the maximum-reservable-bandwidth
           considered as the delta that results in an IGP update
           being flooded";
       }
       leaf threshold-specification {
         when "../threshold-type = 'THRESHOLD_CROSSED'" {
           description
             "The selection of whether mirrored or separate threshold
             values are to be used requires user specified thresholds to
             be set";
         }
         type enumeration {
           enum MIRRORED_UP_DOWN {
             description
               "MIRRORED_UP_DOWN indicates that a single set of
               threshold values should be used for both increasing
               and decreasing bandwidth when determining whether
               to trigger updated bandwidth values to be flooded
               in the IGP TE extensions.";
           }
           enum SEPARATE_UP_DOWN {
             description
               "SEPARATE_UP_DOWN indicates that a separate
               threshold values should be used for the increasing
               and decreasing bandwidth when determining whether
               to trigger updated bandwidth values to be flooded
               in the IGP TE extensions.";
           }
         }
         description
           "This value specifies whether a single set of threshold
           values should be used for both increasing and decreasing
           bandwidth when determining whether to trigger updated
           bandwidth values to be flooded in the IGP TE extensions.
           MIRRORED-UP-DOWN indicates that a single value (or set of
           values) should be used for both increasing and decreasing
           values, where SEPARATE-UP-DOWN specifies that the increasing
           and decreasing values will be separately specified";
       }

       leaf-list up-thresholds {
         when "../threshold-type = 'THRESHOLD_CROSSED'" +
           "and ../threshold-specification = 'SEPARATE_UP_DOWN'" {
             description
               "A list of up-thresholds can only be specified when the
               bandwidth update is triggered based on crossing a
               threshold and separate up and down thresholds are
               required";
         }
         type rt-types:percentage;
         description
           "The thresholds (expressed as a percentage of the maximum
           reservable bandwidth) at which bandwidth updates are to be
           triggered when the bandwidth is increasing.";
       }

       leaf-list down-thresholds {
         when "../threshold-type = 'THRESHOLD_CROSSED'" +
           "and ../threshold-specification = 'SEPARATE_UP_DOWN'" {
             description
               "A list of down-thresholds can only be specified when the
               bandwidth update is triggered based on crossing a
               threshold and separate up and down thresholds are
               required";
         }
         type rt-types:percentage;
         description
           "The thresholds (expressed as a percentage of the maximum
           reservable bandwidth) at which bandwidth updates are to be
           triggered when the bandwidth is decreasing.";
       }

       leaf-list up-down-thresholds {
         when "../threshold-type = 'THRESHOLD_CROSSED'" +
           "and ../threshold-specification = 'MIRRORED_UP_DOWN'" {
             description
               "A list of thresholds corresponding to both increasing
               and decreasing bandwidths can be specified only when an
               update is triggered based on crossing a threshold, and
               the same up and down thresholds are required.";
         }
         type rt-types:percentage;
         description
           "The thresholds (expressed as a percentage of the maximum
           reservable bandwidth of the interface) at which bandwidth
           updates are flooded - used both when the bandwidth is
           increasing and decreasing";
       }
     }

     /* TE interface metric */
     grouping te-metric_config {
       description "Interface TE metric grouping";
       leaf te-metric {
         type te-types:te-metric;
         description "Interface TE metric.";
       }
     }

     /* TE interface switching capabilities */
     grouping te-switching-cap_config {
       description
```

```
         "TE interface switching capabilities";
       list switching-capabilities {
         key "switching-capability";
         description
           "List of interface capabilities for this interface";
         leaf switching-capability {
           type identityref {
             base te-types:switching-capabilities;
           }
           description
             "Switching Capability for this interface";
         }
         leaf encoding {
           type identityref {
             base te-types:lsp-encoding-types;
           }
           description
             "Encoding supported by this interface";
         }
       }
     }

     grouping te-advertisements_state {
       description
         "TE interface advertisements state grouping";
       container te-advertisements_state {
         description
           "TE interface advertisements state container";
         leaf flood-interval {
           type uint32;
           description
             "The periodic flooding interval";
         }
         leaf last-flooded-time {
           type uint32;
           units seconds;
           description
             "Time elapsed since last flooding in seconds";
         }
         leaf next-flooded-time {
           type uint32;
           units seconds;
           description
             "Time remained for next flooding in seconds";
         }
         leaf last-flooded-trigger {
           type enumeration {
             enum link-up {
               description "Link-up flooding trigger";
             }
             enum link-down {
               description "Link-up flooding trigger";
             }
             enum threshold-up {
               description
                 "Bandwidth reservation up threshold";
             }
             enum threshold-down {
               description
                 "Bandwidth reservation down threshold";
             }
             enum bandwidth-change {
               description "Banwidth capacity change";
             }
             enum user-initiated {
               description "Initiated by user";
             }
             enum srlg-change {
               description "SRLG property change";
             }
             enum periodic-timer {
               description "Periodic timer expired";
             }
           }
           description "Trigger for the last flood";
         }
         list advertized-level-areas {
           key level-area;
           description
             "List of areas the TE interface is advertised
             in";
           leaf level-area {
             type uint32;
             description
               "The IGP area or level where the TE
               interface state is advertised in";
           }
         }
       }
     }

     /* TE interface attributes grouping */
     grouping te-attributes {
       description "TE attributes configuration grouping";
       uses te-metric_config;
       uses te-admin-groups_config;
```

```
       uses te-srlgs_config;
       uses te-igp-flooding-bandwidth_config;
       uses te-switching-cap_config;
       container state {
         config false;
         description
           "State parameters for interface TE metric";
         uses te-advertisements_state;
       }
     }

     grouping te-all-attributes {
       description
         "TE attributes configuration grouping for all
          interfaces";
       uses te-igp-flooding-bandwidth_config;
     }
     /*** End of TE interfaces device groupings ***/


     /**
      * TE device augmentations
      */
     augment "/te:te" {
       description "TE global container.";
       /* TE Interface Configuration Data */
       uses interfaces-grouping;
     }

     /* TE globals device augmentation */
     augment "/te:te/te:globals" {
       description
         "Global TE device specific configuration parameters";
       uses lsp-device-timers;
     }

     /* TE tunnels device configuration augmentation */
     augment "/te:te/te:tunnels/te:tunnel" {
       description
         "Tunnel device dependent augmentation";
       uses lsp-device-timers_config;
     }
     augment "/te:te/te:tunnels/te:tunnel/te:state" {
       description
         "Tunnel device dependent augmentation";
       uses lsp-device-timers_config;
     }
     /* TE LSPs device state augmentation */
     augment "/te:te/te:lsps-state/te:lsp" {
        description
          "LSP device dependent augmentation";
        uses lsps-device_state;
     }

     augment "/te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths" +
       "/te:p2p-secondary-path/te:state/te:lsps/te:lsp" {
       description
         "LSP device dependent augmentation";
       uses lsps-device_state;
     }

     augment "/te:te/te:tunnels/te:tunnel/te:p2p-primary-paths" +
       "/te:p2p-primary-path/te:state/te:lsps/te:lsp" {
       description
         "LSP device dependent augmentation";
       uses lsps-device_state;
     }

     /* TE interfaces RPCs/execution Data */
     rpc interfaces-rpc {
       description
         "Execution data for TE interfaces.";
     }

     /* TE Interfaces Notification Data */
     notification interfaces-notif {
       description
         "Notification messages for TE interfaces.";
     }
   }
   <CODE ENDS>
```

Figure 8: TE device specific YANG module

5. IANA Considerations

This document registers the following URIs 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-te XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-te-device 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-te namespace: urn:ietf:params:xml:ns:yang:ietf-te prefix: ietf-te reference: RFC3209

name: ietf-te-device namespace: urn:ietf:params:xml:ns:yang:ietf-te prefix: ietf-te-device reference: RFC3209

6. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC8341] provides means to restrict access for particular NETCONF

users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module 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. Following are the subtrees and data nodes and their sensitivity/vulnerability:

"/te/globals": This module specifies the global TE configurations on a device. Unauthorized access to this container could cause the device to ignore packets it should receive and process.

"/te/tunnels": This list specifies the configured TE tunnels on a device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

"/te/lsps-state": This list specifies the state derived LSPs. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

"/te/interfaces": This list specifies the configured TE interfaces on a device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

7. Acknowledgement

The authors would like to thank the members of the multi-vendor YANG design team who are involved in the definition of this model.

The authors would also like to thank Loa Andersson, Lou Berger, Sergio Belotti, Italo Busi, Carlo Perocchio, Francesco Lazzeri, Aihua Guo, Dhruv Dhody, Anurag Sharma, and Xian Zhang for their comments and providing valuable feedback on this document.

8. Contributors

Xia Chen
Huawei Technologies

Email: jescia.chenxia@huawei.com

Raqib Jones
Brocade

Email: raqib@Brocade.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

9. Normative References

[I-D.ietf-teas-yang-path-computation]
Busi, I., Belotti, S., Lopezalvarez, V., Dios, O., Sharma, A., Shi, Y., Vilata, R., and K. Sethuraman, "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation-02 (work in progress), June 2018.

[I-D.ietf-teas-yang-rsvp]
Beeram, V., Saad, T., Gandhi, R., Liu, X., Bryskin, I., and H. Shah, "A YANG Data Model for Resource Reservation Protocol (RSVP)", draft-ietf-teas-yang-rsvp-09 (work in progress), May 2018.

[I-D.ietf-teas-yang-te-types]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Traffic Engineering Common YANG Types", draft-ietf-teas-yang-te-types-01 (work in progress), October 2018.

[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>.

[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>.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <https://www.rfc-editor.org/info/rfc3473>.

[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>.

[RFC6107] Shiomoto, K., Ed. and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <https://www.rfc-editor.org/info/rfc6107>.

[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>.

[RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <https://www.rfc-editor.org/info/rfc8294>.

[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>.

Authors’ Addresses

Tarek Saad
Cisco Systems Inc

Email: tsaad@cisco.com

Rakesh Gandhi
Cisco Systems Inc

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Himanshu Shah
Ciena

Email: hshah@ciena.com

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com

TEAS Working Group D. King (Ed.)
Internet-Draft Old Dog Consulting
Intended status: Informational Y. Lee (Ed.)
Expires: April 23, 2018 Huawei

October 22, 2018

Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing
draft-king-teas-applicability-actn-slicing-04

Abstract

Network abstraction is a technique that can be applied to a network domain to select network resources by policy to obtain a view of potential connectivity

Network slicing is an approach to network operations that builds on the concept of network abstraction to provide programmability, flexibility, and modularity. It may use techniques such as Software Defined Networking (SDN) and Network Function Virtualization (NFV) to create multiple logical (virtual) networks, each tailored for a set of services that are sharing the same set of requirements, on top of a common network.

These logical networks are referred to as transport network slices. A transport network slice does not necessarily represent dedicated resources in the network, but does constitute a commitment by the network provider to provide a specific level of service.

The Abstraction and Control of Traffic Engineered Networks (ACTN) defines an SDN-based architecture that relies on the concepts of network and service abstraction to detach network and service control from the underlying data plane.

This document outlines the applicability of ACTN to transport network slicing in an IETF technology network. It also identifies the features of network slicing not currently within the scope of ACTN, and indicates where ACTN might be extended.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The principles of network resource separation are not new. For years, separated overlay and logical (virtual) networking have existed, allowing multiple connectivity services to be deployed over a single physical network comprised of single or multiple layers. However, several key differences exist that differentiate overlay and virtual networking from network slicing.

A transport network slice construct provides an end-to-end logical network, often with compute functions and utilising shared underlying (physical or virtual) network resources. This logical network is separated from other, often concurrent, logical networks each with independent control and management, and each of which can be created or modified on demand.

At one end of the spectrum, a virtual private wire or a virtual private network (VPN) may be used to build a network slice. In these cases, the network slices do not require the service provider to isolate network resources for the provision of the service - the service is "virtual".

At the other end of the spectrum there may be a detailed description of a complex service that will meet the needs of a set of applications with connectivity and service function requirements that may include compute resource, storage capability, and access to content. Such a service may be requested dynamically (that is, instantiated when an application needs it, and released when the application no longer needs it), and modified as the needs of the application change.

Each example represents a self-contained network that must be flexible enough to simultaneously accommodate diverse business-driven use cases from multiple players on a common network infrastructure.

This document outlines the application of the ACTN architecture [RFC8353] and enabling technologies to provide transport network slicing in an IETF technology network. It describes how the ACTN functional components can be used to support model-driven partitioning of variable-sized bandwidth to facilitate network sharing and virtualization. Furthermore, the use of model-based interfaces to dynamically request the instantiation of virtual networks could be extended to encompass requesting and instantiation of specific service functions (which may be both physical and/or virtual), and to partition network resources such as compute resource, storage capability, and access to content.

In an IETF context, there are works in progress that have some bearing with network slicing such as Enhanced VPN (VPN+) and DetNet. Both works are an independent work in their own scope while

This document highlights how the ACTN approach might be extended to address these other requirements of network slicing where TE is required.

1.1. Terminology

Resource: Any features that can be delivered, including connectivity, compute, storage, and content delivery.

Service Functions (SFs): Components that provide specific function within a network. SFs are often combined in a specific sequence, service function chain, to deliver services.

Infrastructure Resources: The hardware and necessary software for hosting and connecting SFs. These resources may include computing hardware, storage capacity, network resources (e.g. links and switching/routing devices enabling network connectivity), and physical assets for radio access.

Service Provider: A server network or collection of server networks.

Consumer: Any application, client network, or customer of a network provider.

Service Level Agreement (SLA): An agreement between a consumer and network provider that describes the quality with which features and functions are to be delivered. It may include measures of bandwidth, latency, and jitter; the types of service (such as the network service functions or billing) to be executed; the location, nature, and quantities of services (such as the amount and location of compute resources and the accelerators require).

Network Slice: An agreement between a consumer and a service provider to deliver network resources according to a specific service level agreement. A slice could span multiple technology (e.g., radio, transport and cloud) and administrative domains.

IETF Technology: A TE network slice or transport network slice.

2. Requirements for Network Slicing

The concept of network slicing is considered a key capability for future networks and, to serve customers with a wide variety of different service needs, in term of latency, reliability, capacity, and service function specific capabilities.

This section outlines the key capabilities required, and further discussed in [ngmn-network-slicing], [network-slice-5g], [3gpp.28.801] and [onf-tr526], to realise network slicing in an IETF technology network.

2.1. Resource Slicing

For network slicing, it is important to consider both infrastructure resources and servic functions. This allows a flexible approach to deliver a range of services both by partitioning (slicing) the available network resources to present them for use by a consumer, but also by providing instances of SFs at the right locations and in the correct chaining logic, with access to the necessary hardware, including specific compute and storage resources.

Mapping of resources to slices may 1-to-1, or resources may be shared among multiple slices.

2.2. Network and Function Virtualization

Virtualization is the abstraction of resources where the abstraction is made available for use by an operations entity, for example, by the Network Management Station (NMS) of a consumer network. The resources to be virtualized can be physical or already virtualized, supporting a recursive pattern with different abstraction layers. Therefore, Virtualization is critical for network slicing as it enables effective resource sharing between network slices.

Just as server virtualization makes virtual machines (VMs) independent of the underlying physical hardware, network Virtualization enables the creation of multiple isolated virtual networks that are completely decoupled from the underlying physical network, and can safely run on top of it.

2.3. Resource Isolation

Isolation of data and traffic is a major requirement that must be satisfied for certain applications to operate in concurrent network slices on a common shared underlying infrastructure. Therefore, isolation must be understood in terms of:

o Performance: Each slice is defined to meet specific service requirements, usually expressed in the form of Key Performance Indicators (KPIs). Performance isolation requires that service delivery on one network slice is not adversely impacted by congestion and performance levels of other slices;

o Security: Attacks or faults occurring in one slice must not have an impact on other slices, or customer flows are not only isolated on network edge, but multiple customer traffic is not mixed across the core of the network. Moreover, each slice must have independent security functions that prevent unauthorised entities to have read or write access to slice-specific configuration, management, accounting information, and able to record any of these attempts, whether authorised or not;

- Management: Each slice must be independently viewed, utilised and managed as a separate network.

2.4. Control and Orchestration

Orchestration is the overriding control method for network slicing. We may define orchestration as combining and coordinating multiple control methods to provide an operational mechanism that can deliver services and control underlying resources. In a network slicing environment, an orchestrator is needed to coordinate disparate processes and resources for creating, managing, and deploying the end-to-end service. Two scenarios are outlined below where orchestration would be required:

1. Multi-domain Orchestration: Managing connectivity setup of the transport service, across multiple administrative domains;

2. End-to-end Orchestration: Combining resources for an "end-to-end service (e.g., transport connectivity with firewalling and guaranteed bandwidth and minimum delay for premium radio users (spanning multiple domains).

In addition, 3GPP has also developed Release 14 "Study on management and orchestration of network slicing for next generation network" [3gpp.28.801], which defines an information model where the network slice as well as physical and virtualized network functions belong to the network operator domain, while the virtualized resources belong to another domain operated by a Virtualization infrastructure service provider.

3. Abstraction and Control of Traffic Engineered (TE) Networks (ACTN)

The framework for ACTN [RFC8453] includes a reference architecture that has been adapted for Figure 1 in this document, it describes the functional entities and methods for the coordination of resources across multiple domains, to provide end-to-end services, components include:

- Customer Network Controller (CNC);

- Multi-domain Service Coordinator (MDSC);

- Provisioning Network Controller (PNC).

```
      ---------                 ---------                  ---------
      | CNC-A |                 | CNC-B |                  | CNC-C |
      ---------                 ---------                  ---------
            \                       |                        /
             \__________            |-CMI I/F     __________/
                        \           |            /
                         -------------------------
                         |         MDSC          |
                         -------------------------
                          /      /    |         \
                         /      /     |-MPI I/F  \
                        /      /      |           \
                  -------   -------  -------       -------
                  | PNC |   | PNC |  | PNC |       | PNC |
                  -------   -------  -------       -------

   CMI - (CNC-MDSC Interface )
   MPI - (MDSC-PNC Interface)
```

Figure 1: ACTN Hierarchy

ACTN facilitates end-to-end connections and provides them to the user. The ACTN framework [RFC8453] highlights how:

- Abstraction of the underlying network resources 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;

- Creation of a virtualized environment allowing operators to view and control multi-domain networks as a single virtualized network;

- The 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. The capabilities that ACTN provides to enable slicing are outlined in Section 2 (Requirements for Network Slicing).

3.1. ACTN Virtual Network as a "Network Slice"

To support multiple clients each with its own view of and control of the server network, a network operator needs to partition (or "slice") the network resources. The resulting slices can be assigned to each client for guaranteed usage which is a step further than shared use of common network resources. See [actn-vn] for detailed ACTN VN and VNS.

An ACTN Virtual Network (VN) is a client view that may be considered a "network slice" of the ACTN managed infrastructure, and is presented by the ACTN provider as a set of abstracted resources.

Depending on the agreement between client and provider various VN operations and VN views are possible.

o Network Slice Creation: A VN could be pre-configured and created via static or dynamic request and negotiation between customer and provider. It must meet the specified SLA attributes which satisfy the customer's objectives.

o Network Slice Operations: The network slice may be further modified and deleted based on customer request to request changes in the network resources reserved for the customer, and used to construct the network slice. The customer can further act upon the network slice to manage traffic flow across the network slice.

o Network Slice View: The VN topology from a customer point of view. These may be a variety of tunnels, or an entire VN topology. Such connections may comprise of customer end points, access links, intra domain paths and inter-domain links.

Primitives (capabilities and messages) have been provided to support the different ACTN network control functions that will enable network slicing. These include: topology request/query, VN service request, path computation and connection control, VN service policy negotiation, enforcement, routing options. [RFC8454]

3.2. Examples of ACTN Delivering Types of Network Slices

In examples below the ACTN framework is used to provide control, management and orchestration for the network slice life-cycle, the connectivity . These dynamic and highly flexible, end-to-end and dedicated network slices utilising common physical infrastructure, and according to vertical-specific requirements.

The rest of this section provides three examples of using ACTN to achieve different scenarios of ACTN for network slicing. All three scenarios can be scaled up in capacity or be subject to topology changes as well as changes from customer requirements perspective.

3.2.1. ACTN Used for Virtual Private Line Model

ACTN Provides virtual connections between multiple customer locations, requested via Virtual Private Line (VPL) requester (CNC-A). Benefits of this model include:

- Automated: the service set-up and operation is network provider managed;
- Virtual: the private line is seamlessly extended from customers Site A (vCE1 to vCE2) and Site B (vCE2 to vCE3) across the ACTN-managed WAN to Site C;
- Agile: on-demand where the customer needs connectivity and fully adjustable bandwidth.

```
                     (Customer VPL Request)
                                |
                            ---------
                            | CNC-A |
   Boundary                 ---------
   Between  ====================|====================
   Customer &                   |
   Network Provider          --------
                             | MDSC |
                             --------
                              __|__
        Site A               ( PNC )             Site B
         ------             (       )            ------
         |vCE1|=============( Phys. )=============|vCE2|
         ------              ( Net )             ------
               \              -----              /
                \               ||              /
                 \              ||             /
            VPL 1 \__           ||          __/ VPL 2
                     \          ||         /
                      \         ||        /
                       \      ------     /
                        ------|vCE3|-----
                              ------
                              Site C
```

Figure 2: Virtual Private Line Model

3.2.2. ACTN Used for VPN Delivery Model

ACTN Provides VPN connections between multiple sites, requested via a VPN requestor (CNC-A), which is managed by the customer themselves. The CNC will then interact with the network providers MDSC. Benefits of this model include:

- o Provides edge-to-edge VPN multi-access connection;
- o Mostly network provider managed, with some flexibility delegated to the customer managed CNC.

```
          ----------------                            ----------------
          | Site-A Users |___________     ____________| Site-B Users |
          ----------------           |   |            ----------------
                                    -------
                                    |CNC-A|
   Boundary                         -------
   Between   ==========================|==========================
   Customer &                          |
   Network Provider                    |
                                       |
                                ---------------
                                |     MDSC     |
                                ---------------
                      _________/       |       \__________
                     /                 |                  \
                    /                  |                   \
               ---------           ---------           ---------
               |  PNC  |           |  PNC  |           |  PNC  |
               ---------           ---------           ---------
                  |                    |                  /
                  |                    |                 /
                -----                -----           -----
               (     )              (     )         (     )
  <Site A>----( Phys. )------------( Phys. )-------( Phys. )----<Site B>
               ( Net )              ( Net )         ( Net )
                -----                -----           -----
```

Figure 3: VPN Model

3.2.3. ACTN Used to Deliver a Virtual Customer Network

In this example ACTN provides a virtual network resource to the customer. This resource is customer managed. Empowering the tenant to control allocated slice (recursively). Benefits of this model include:

- o The MDSC provides the topology as part of the customer view so that the customer can control their network slice to fit their needs;

- o Resource isolation, each customer network slice is fixed and will not be affected by changes to other customer network slices;

- o Applications can interact with their assigned network slice directly, the customer may implement their own network control method and traffic prioritization, manage their own addressing scheme, and further slice their assigned network resource;

- o The network slice may also include specific capability nodes, delivered as Physical Network Functions (PNFs) or Virtual Network Functions (VNFs).

```
                                             ___________
                  ---------------           (  Network  )
                  |    CNC      |---------->(  Slice 2  )
                  ---------------          _(__________ )
               ---------------            (  Network  )_)
               |    CNC      |----------->(  Slice 1  ) ^
               ---------------            (           ) :
                     ^                    (___________) :
                     |                        ^    ^    :
   Boundary          |                        :    :    :
   Between ==========|========================:====:====:========
   Customer &        |                        :    :    :
   Network Provider  |                        :    :    :
                     v                        :    :    :
               ---------------                :    :....:
               |    MDSC     |                :         :
               ---------------                :         :
                     ^                     ---^------    ...
                     |                    (          )      .
                     v                   (  Physical  )      .
                 ----------------         ( Network  )        .
                 |     PNC      |<------>  (        )      ---^------
               ---------------- |           --------      (          )
               |              |--                        (  Physical  )
               |    PNC       |<------------------------->( Network  )
               ---------------                             (        )
                                                            --------
```

Figure 4: Network Slicing

3.3. Network Slice Service Mapping from TE to ACTN VN Models

The role of TE-service mapping model [te-service-mapping] is to create a binding relationship across a Layer-3 Service Model [l3sm], Layer-2 Service Model and TE Tunnel model, via a generic ACTN Virtual Network (VN) model [actn-vn].

The ACTN VN YANG model is a generic virtual network service model that allows customers (internal or external) to create a VN that meets the customer's service objective with various constraints.

The TE-service mapping model is needed to bind L3VPN specific service model with TE-specific parameters. This binding will facilitate a seamless service operation with underlay-TE network visibility. The TE-service model developed in this document can also be extended to support other services including L2SM, and L1CSM network service models.

```
+--------------+        +-------------------+         +----------+
|    LxSM      |o-------|                   | . . . . | ACTN VN  |
+--------------+ augment|                   |         +----------+
                        |                   |         +----------+
+--------------+        |  Augmented LxSM   | . . . . | TE-topo  |
| TE & Service |------->|      Model        |         +----------+
| Mapping Types| import |                   |         +----------+
+--------------+        |                   | . . . . | TE-tunnel|
                        +-------------------+ reference+----------+
```

Figure 5: TE-Service Mapping ([te-service-mapping])

Editors note - We plan to provide a list of models available and their relationships/dependencies. We will also provide a vertical hierarchy of how these models may be used between functional components in ACTN.

3.4. ACTN VN KPI telemetry Models

The role of ACTN VN KPI telemetry model [actn-pm-telemetry] is to provide YANG models so that customer can define key performance monitoring data relevant for its VN/network slicing via the YANG subscription model.

Key characteristics of [actn-pm-telemetry] include:

o an ability to provide scalable VN-level telemetry aggregation based on customer-subscription model for key performance parameters defined by the customer;

o an ability to facilitate proactive re-optimization and reconfiguration of VNs/Netork Slices based on network autonomic traffic engineering scaling configuration mechanism.

5. IANA Considerations

This document makes no requests for action by IANA.

6. Security Considerations

Network slicing involves the control of network resources in order to meet the service requirements of consumers. In some deployment models, the consumer is able to directly request modification in the behaviour of resources owned and operated by a service provider. Such changes could significantly affect the service provider’s ability to provide services to other consumers. Furthermore, the resources allocated for or consumed by a consumer will normally be billable by the service provider.

Therefore, it is crucial that the mechanisms used in any network slicing system allow for authentication of requests, security of those requests, and tracking of resource allocations.

It should also be noted that while the partitioning or slicing of resources is virtual, the consumers expect and require that there is no risk of leakage of data from one slice to another, no transfer of knowledge of the structure or even existence of other slices, and that changes to one slice (under the control of one consumer) should not have detrimental effects on the operation of other slices (whether under control of different or the same consumers) beyond the limits allowed within the SLA. Thus, slices are assumed to be private and to provide the appearance of genuine physical connectivity.

ACTN operates using the [netconf] or [restconf] protocols and assumes the security characteristics of those protocols. Deployment models for ACTN should fully explore the authentication and other security aspects before networks start to carry live traffic.

7. Acknowledgements

Thanks to Qin Wu, Andy Jones, Ramon Casellas, and Gert Grammel for their insight and useful discussions about network slicing.

8. References

8.1. Normative References

8.2. Informative References

[ngmn-network-slicing]
NGMN, "Description of Network Slicing Concept", 1 2016, <https://www.ngmn.org/uploads/media/160113_Network_Slicing_v1_0.pdf>.

[3gpp.28.801]
3GPP, "Study on management and orchestration of network slicing for next generation network", 3GPP TR 28.801 0.4.0, 1 2017, <http://www.3gpp.org/ftp/Specs/html-info/28801.htm>.

[network-slice-5g]
"Network Slicing for 5G with SDN/NFV: Concepts, Architectures and Challenges", Jose Ordonez-Lucena, Pablo Ameigeiras, Diego Lopez, Juan J. Ramos-Munoz, Javier Lorca, Jesus Folgueira, IEEE Communications Magazine 55, March 2017

[onf-tr526]
ONF TR-526, "Applying SDN Architecture to 5G Slicing", April 2016.

[RFC8453] Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework, RFC 8453, September 2018.

[te-service-mapping]
Y. Lee, D. Dhody, and D. Ceccarelli, "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang, work in progress.

[actn-vn] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

[RFC8454] Y. Lee, S. Belotti (Editors), "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC 8454, September 2018.

[actn-pm-elemetry] Y. Lee, et al, "YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics", draft-lee- teas-actn-pm-telemetry-autonomics, work in progress.

[l3sm] Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8049, February 2017

[netconf] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.

[restconf] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf, work in progress.

[sf-topology] I. Bryskin, et al, "Use Cases for SF Aware Topology Models", draft-ietf-teas-use-cases-sf-aware-topo-model, work in progress.

[vpn+] S. Bryant and J. Dong, "Enhanced Virtual Private Networks (VPN+)", draft-bryant-rtgwg-enhanced-vpn, work in progress.

9. Contributors

The following people contributed text to this document.

Adrian Farrel
Email: afarrel@juniper.net

Mohamed Boucadair
Email: mohamed.boucadair@orange.com

Sergio Belotti
Email: sergio.belotti@nokia.com

Daniele Ceccarelli
Email: daniele.ceccarelli@ericsson.com

Haomian Zheng
Email: zhenghaomian@huawei.com

Authors' Addresses

Daniel King
Email: daniel@olddog.co.uk

Young Lee
Email: leeyoung@huawei.com

TEAS Working Group
Internet Draft
Intended Status: Standard Track
Expires: April 6, 2019

Y. Lee (Editor)
Dhruv Dhody
Satish Karunanithi
Huawei
Ricard Vilalta
CTTC
Daniel King
Lancaster University
Daniele Ceccarelli
Ericsson

October 5, 2018

YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics

draft-lee-teas-actn-pm-telemetry-autonomics-08

Status of this Memo

This Internet-Draft will expire on April 6, 2019.

Copyright Notice

Abstract

Abstraction and Control of TE Networks (ACTN) refers to the set of virtual network operations needed to operate, control and manage large-scale multi-domain, multi-layer and multi-vendor TE networks, so as to facilitate network programmability, automation, efficient resource sharing.

This document provides YANG data models that describe Key Performance Indicator (KPI) telemetry and network autonomics for TE-tunnels and ACTN VNs.

Table of Contents

1. Introduction

Abstraction and Control of TE Networks (ACTN) describes a method for operating a Traffic Engineered (TE) network (such as an MPLS-TE network or a layer 1/0 transport network) to provide connectivity and virtual network services for customers of the TE network [ACTN-Frame]. The services provided can be optimized to meet the requirements (such as traffic patterns, quality, and reliability) of the applications hosted by the customers. Data models are a representation of objects that can be configured or monitored within a system. Within the IETF, YANG [RFC6020] is the language of choice for documenting data models, and YANG models have been produced to allow configuration or modeling of a variety of network devices, protocol instances, and network services. YANG data models have been classified in [Netmod-Yang-Model-Classification] and [Service-YANG].

[ACTN-VN] describes how customers or end to end orchestrators can request and/or instantiate a generic virtual network service. [ACTN-Applicability] describes a connection between IETF YANG model classifications to ACTN interfaces. In particular, it describes the customer service model can be mapped into the CMI (CNC-MDSC Interface) of the ACTN architecture.

The YANG model on the ACTN CMI is known as customer service model in [Service-YANG]. [PCEP-Service-Aware] describes key network performance data to be considered for end-to-end path computation in TE networks. Key performance indicator is a term that describes critical performance data that may affect VN/TE service.

1.1. Terminology

1.2. Tree Structure - Legend

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 [RFC8342].

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
rt	ietf-routing-types	[Routing-Types]
te	ietf-te	[TE-tunnel]
te-types	ietf-te-types	[TE-Types]
te-kpi	ietf-te-kpi-telemetry	[This I-D]
vn	ietf-actn-vn	[ACTN-VN]
actn-tel	ietf-actn-te-kpi-telemetry	{This I-D]

Table 1: Prefixes and corresponding YANG modules

2. Use-Cases

[ACTN-PERF] describes use-cases relevant to this draft. It introduces the dynamic creation, modification and optimization of services based on the performance monitoring in the Abstraction and Control of Transport Networks (ACTN) architecture. Figure 1 shows a high-level workflows for dynamic service control based on traffic monitoring.

Some of the key points from [ACTN-PERF] 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 CAPEX/OPEX.
- Customer services have various SLA requirements, such as service availability, latency, latency jitter, packet loss 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 service 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 CMI and MPI interfaces should be carefully considered.

```
      +-------------------------------------------+
      | CNC   +-----------------------------+     |
      |       | Dynamic Service Control APP |     |
      |       +-----------------------------+     |
      +-------------------------------------------+
      1.Traffic|  /|\4.Traffic            | /|\
      Monitor& |   | Monitor             |  | 8.Traffic
      Optimize |   | Result     5.Service |  | modify &
      Policy   |   |              modify& |  | optimize
              \|/  |       optimize Req.\|/ | result
      +------------------------------------------------+
      | MDSC       +-------------------------------+   |
      |            |Dynamic Service Control Agent |    |
      |            +-------------------------------+   |
      |      +---------------+ +-------------------+   |
      |      | Flow Optimize | | vConnection Agent |   |
      |      +---------------+ +-------------------+   |
      +------------------------------------------------+
      2. Path |   /|\3.Traffic            |  |
      Monitor |    | Monitor             |  |7.Path
      Request |    | Result      6.Path   |  | modify &
              |    |             modify&  |  | optimize
             \|/   |       optimize Req.\|/ | result
      +-------------------------------------------------------+
      | PNC +----------------------+ +----------------------+ |
      |     | Network Provisioning | |Abstract Topology Gen.| |
      |     +----------------------+ +----------------------+ |
      |     +------------------+ +--------------------+       |
      |     |Network Monitoring| |Physical Topology DB|       |
      |     +------------------+ +--------------------+       |
      +-------------------------------------------------------+
```

Figure 1 Workflows for dynamic service control based on traffic monitoring

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 (See Section 4 for details)

(ii) ACTN TE KPI Telemetry Model which provides the VN level of the aggregated performance monitoring mechanism (See Section 5 for details)

The models include -

(i) Performance Telemetry details as measured during the last interval, ex delay.

(ii) Scaling Intent based on with TE/VN could be scaled in/out.

[Editor’s Note - Need to decide if scaling and telemetry can be in the same model as per the current draft.]

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.

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.

```
+------------+          +--------------+
|  TE-Tunnel |          |    TE KPI    |
|   Model    |<---------|  Telemetry   |
+------------+ augments |     Model    |
                        +--------------+
```

3.2. ACTN TE KPI Telemetry Model

This module describes performance telemetry for ACTN 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.

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:

```
 +------------------------------------------------------------+
 | CNC                                                        |
 |                                                            |
 +------------------------------------------------------------+

1.CNC sets the         |   /|\   2. MDSC gets VN Telemetry
grouping op, and       |    |
subscribes to the      |    |    VN KPI TELEMETRY (VN Level)
VN level telemetry for |    |    VN Utilized-bw-percentage:
Delay and              |    |       Minimum across VN Members
Utilized-bw-pecentage  |    |    VN Delay: Maximum across VN
                      \|/   |     Members
 +------------------------------------------------------------+
 | MDSC                                                       |
 |                                                            |
 +------------------------------------------------------------+
```

The ACTN VN TE-Telemetry Model augments the basic ACTN 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 ACTN VN Telemetry YANG model.

```
+----------+          +--------------+
|  ACTN VN | augments |     ACTN     |
|   Model  |<---------| TE-Telemetry |
+----------+          |     Model    |
                      +--------------+
```

4. Notification

This model does not define specific notifications. To enable notifications, the mechanism defined in [I-D.ietf-netconf-yang-push] and [I-D.ietf-netconf-rfc5277bis] can be used. This mechanism currently allows the user to:

- Subscribe 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.

4.1. YANG Push Subscription Examples

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

```
   <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">
                           <utilized-
   percentage/>

                        </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 for the telemetry information for all VNs. The telemetry parameter that the client is interested in is one-way delay and utilized bandwidth percentage.

```
   <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">
             <actn-state xmlns="urn:ietf:params:xml:ns:yang:ietf-actn-
   vn">
                <vn>
                   <vn-list>
                     <vn-id/>
                     <vn-name/>
                     <vn-
   telemetry  xmlns="urn:ietf:params:xml:ns:yang:ietf-actn-te-kpi-
   telemetry">
                         <one-way-delay/>
                         <utilized-
   percentage/>

                     </vn-telemetry >
                   </vn-list>
                 </vn>
             </actn-state>
          </filter>
          <period>500</period>
       </establish-subscription>
    </netconf:rpc>
```

5. 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 scale-in-operation-type?    scaling-criteria-operation
    |  |  +-rw scale-out-operation-type?   scaling-criteria-operation
    |  |  +-rw scaling-condition* [performance-type]
    |  |     +-rw performance-type            identityref
    |  |     +-rw te-telemetry-tunnel-ref?    -> /te:te/tunnels/tunnel/name
    |  +-rw scale-out-intent
    |     +-rw threshold-time?              uint32
    |     +-rw cooldown-time?               uint32
    |     +-rw scale-in-operation-type?    scaling-criteria-operation
    |     +-rw scale-out-operation-type?   scaling-criteria-operation
    |     +-rw scaling-condition* [performance-type]
    |        +-rw performance-type            identityref
    |        +-rw te-telemetry-tunnel-ref?    -> /te:te/tunnels/tunnel/name
    +-ro te-telemetry
       +-ro id?                            string
       +-ro performance-metric-one-way
       |  +-ro one-way-delay?                  uint32
       |  +-ro one-way-min-delay?              uint32
       |  +-ro one-way-max-delay?              uint32
       |  +-ro one-way-delay-variation?        uint32
       |  +-ro one-way-packet-loss?            decimal64
       |  +-ro one-way-residual-bandwidth?     rt-types:bandwidth-ieee-float32
       |  +-ro one-way-available-bandwidth?    rt-types:bandwidth-ieee-float32
       |  +-ro one-way-utilized-bandwidth?     rt-types:bandwidth-ieee-float32
       +-ro performance-metric-two-way
       |  +-ro two-way-delay?              uint32
       |  +-ro two-way-min-delay?          uint32
       |  +-ro two-way-max-delay?          uint32
       |  +-ro two-way-delay-variation?    uint32
       |  +-ro two-way-packet-loss?        decimal64
       +-ro te-ref?                        -> /te:te/tunnels/tunnel/name



module: ietf-actn-te-kpi-telemetry
  augment /vn:actn/vn:vn/vn:vn-list:
    +-rw vn-scaling-intent
    |  +-rw scale-in-intent
    |  |  +-rw threshold-time?              uint32
    |  |  +-rw cooldown-time?               uint32
    |  |  +-rw scale-in-operation-type?    scaling-criteria-operation
    |  |  +-rw scale-out-operation-type?   scaling-criteria-operation
    |  |  +-rw scaling-condition* [performance-type]
    |  |     +-rw performance-type            identityref
    |  |     +-rw te-telemetry-tunnel-ref?    -> /te:te/tunnels/tunnel/name
    |  +-rw scale-out-intent
    |     +-rw threshold-time?              uint32
    |     +-rw cooldown-time?               uint32
    |     +-rw scale-in-operation-type?    scaling-criteria-operation
    |     +-rw scale-out-operation-type?   scaling-criteria-operation
    |     +-rw scaling-condition* [performance-type]
    |        +-rw performance-type           identityref
    |        +-rw te-telemetry-tunnel-ref?   -> /te:te/tunnels/tunnel/name
    +-ro vn-telemetry
       +-ro performance-metric-one-way
       |  +-ro one-way-delay?                uint32
       |  +-ro one-way-min-delay?            uint32
       |  +-ro one-way-max-delay?            uint32
       |  +-ro one-way-delay-variation?      uint32
       |  +-ro one-way-packet-loss?          decimal64
       |  +-ro one-way-residual-bandwidth?    rt-types:bandwidth-ieee-float32
       |  +-ro one-way-available-bandwidth?   rt-types:bandwidth-ieee-float32
       |  +-ro one-way-utilized-bandwidth?    rt-types:bandwidth-ieee-float32
       +-ro performance-metric-two-way
       |  +-ro two-way-delay?            uint32
       |  +-ro two-way-min-delay?        uint32
       |  +-ro two-way-max-delay?        uint32
       |  +-ro two-way-delay-variation?   uint32
       |  +-ro two-way-packet-loss?      decimal64
       +-ro grouping-operation?          grouping-operation
  augment /vn:actn/vn:vn/vn:vn-list/vn:vn-member-list:
    +-ro vn-member-telemetry
       +-ro performance-metric-one-way
       |  +-ro one-way-delay?                uint32
       |  +-ro one-way-min-delay?            uint32
       |  +-ro one-way-max-delay?            uint32
       |  +-ro one-way-delay-variation?      uint32
       |  +-ro one-way-packet-loss?          decimal64
       |  +-ro one-way-residual-bandwidth?    rt-types:bandwidth-ieee-float32
       |  +-ro one-way-available-bandwidth?   rt-types:bandwidth-ieee-float32
       |  +-ro one-way-utilized-bandwidth?    rt-types:bandwidth-ieee-float32
       +-ro performance-metric-two-way
       |  +-ro two-way-delay?            uint32
       |  +-ro two-way-min-delay?        uint32
       |  +-ro two-way-max-delay?        uint32
       |  +-ro two-way-delay-variation?   uint32
       |  +-ro two-way-packet-loss?      decimal64
       +-ro te-grouped-params*           -> /te:te/tunnels/tunnel/te-kpi:te-tel
emetry/id
       +-ro grouping-operation?          grouping-operation
```

6. Yang Data Model

6.1. ietf-te-kpi-telemetry model

The YANG code is as follows:

```
<CODE BEGINS> file "ietf-te-kpi-telemetry@2018-10-05.yang"

module ietf-te-kpi-telemetry {
    namespace "urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry";

    prefix "te-tel";

    import ietf-te {
       prefix "te";
    }

    import ietf-te-types {
       prefix "te-types";
    }

    import ietf-routing-types {
       prefix "rt-types";
    }

    organization
        "IETF Traffic Engineering Architecture and Signaling (TEAS)
        Working Group";

    contact
        "Editor: Young Lee <leeyoung@huawei.com>
         Editor: Dhruv Dhody <dhruv.ietf@gmail.com>
         Editor: Ricard Vilalta <ricard.vilalta@cttc.es>
         Editor: Satish Karunanithi <satish.karunanithi@gmail.com>";

    description
        "This module describes telemetry for teas tunnel model";

    revision 2018-10-05 {
        description
            "Initial revision. This YANG file defines
            the reusable base types for TE telemetry.";
        reference
            "Derived from earlier versions of base YANG files";
    }

    /*
     * Identities
     */

    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";
    }

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

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

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

    identity one-way-packet-loss {
        base telemetry-param-type;
        description
            "To specify packet loss in one (forward) direction.";
    }

    identity two-way-packet-loss {
        base telemetry-param-type;
        description
            "To specify packet loss in in both (forward and reverse)
             directions";
    }


    identity utilized-bandwidth {
        base telemetry-param-type;
        description
            "To specify utilized bandwidth over the specified source
             and destination.";
    }

    identity utilized-percentage {
     base telemetry-param-type;
     description
         "To specify utilization percentage of the entity
          (e.g., tunnel, link, etc.)";
    }
    /*
     * Enums
     */
    typedef scaling-criteria-operation {
        type enumeration {
            enum AND {
                description
                 "AND operation";
            }
            enum OR {
                description
                 "OR operation";
            }
        }
        description
         "Operations to analize list of scaling criterias";
    }



    /*
     * Groupings
     */


    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 te-telemetry-tunnel-ref {
            type leafref {
                path "/te:te/te:tunnels/te:tunnel/te:name";
            }
            description
                "Reference to tunnel";
        }
    }

    grouping scaling-intent {
     description
          "Basic sclaing intent";

     uses scaling-duration;

        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";
        }
        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";
        }

     list scaling-condition {
          key "performance-type";
          description
               "Scaling conditions";
          uses scaling-criteria;
     }

    }

    /*
     * Augments
     */

    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
               "scaling intent";

          container scale-in-intent{
                description
                    "scale-in";
                uses scaling-intent;
            }
          container scale-out-intent{
                description
                    "scale-out";
                uses scaling-intent;
            }
         }
        container te-telemetry {
                config false;
                description
                    "telemetry params";
                    leaf id {
                         type string;
                         description "Id of telemetry param";
                    }

                    uses te-types:performance-metric-container;


                    leaf te-ref{
                         type leafref{ path
'/te:te/te:tunnels/te:tunnel/te:name'; }
                         description "Reference to measured te tunnel";
                    }
        }

    }

}
```

<CODE ENDS>

6.2. ietf-actn-te-kpi-telemetry model

The YANG code is as follows:

<CODE BEGINS> file "ietf-actn-te-kpi-telemetry@2018-10-05.yang"

```
module ietf-actn-te-kpi-telemetry {

    namespace "urn:ietf:params:xml:ns:yang:ietf-actn-te-kpi-telemetry";

    prefix "actn-tel";

    import ietf-actn-vn {
        prefix "vn";
    }

    import ietf-te {
        prefix "te";
    }

    import ietf-te-types {
     prefix "te-types";
    }

    import ietf-te-kpi-telemetry {
        prefix "te-kpi";
    }

    organization
        "IETF Traffic Engineering Architecture and Signaling (TEAS)
        Working Group";

    contact
        "Editor: Young Lee <leeyoung@huawei.com>
         Editor: Dhruv Dhody <dhruv.ietf@gmail.com>
         Editor: Ricard Vilalta <ricard.vilalta@cttc.es>
         Editor: Satish Karunanithi <satish.karunanithi@gmail.com>";

    description
        "This module describes telemetry for actn vn model";


    revision 2018-10-05 {
        description
            "Initial revision. This YANG file defines
            the ACTN VN telemetry.";
        reference
            "Derived from earlier versions of base YANG files";
    }

    /*
     * Typedefs
     */

    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 analyze list of monitored params";
    }



    /*
     * Groupings
     */


   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";
        }

    }

    /*
     * Augments
     */
    augment "/vn:actn/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-intent;
            }
            container scale-out-intent{
                description
                    "VN scale-out";
                uses te-kpi:scaling-intent;
            }
        }
        container vn-telemetry {
            config false;
            description
                "VN telemetry params";

            uses te-types:performance-metric-container;

            leaf grouping-operation {
                type grouping-operation;
                description "describes the operation to apply to the VN-members"
;
            }
        }
    }

    /*
     * VN-member augment
     */
    augment "/vn:actn/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-metric-container;

         uses vn-telemetry-param;
        }
    }
}
<CODE ENDS>
```

7. 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]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF 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.

8. IANA Considerations

TDB

9. Acknowledgements

10. References

10.1. Informative References

[RFC4110] R. Callon and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, July 2005.

[RFC6020] M. Bjorklund, Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[Service-YANG] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", draft-wu-opsawg-service-model-explained, work in progress.

[Netmod-Yang-Model-Classification] D. Bogdanovic, B. Claise, and C. Moberg, "YANG Module Classification", draft-ietf-netmod-yang-model-classification, work in progress.

[Netconf] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.

[Restconf] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf, work in progress.

[Routing-Types] X. Liu, et al, "Routing Area Common YANG Data Types", draft-ietf-rtgwg-routing-types, work in progress.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018,

10.2. Normative References

[ACTN-Frame] D. Cecarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework, work in progress.

[TE-Topology] X. Liu, et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.

[TE-Tunnel] T. Saad (Editor), "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.

[ACTN-VN] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

[L3SM-YANG] S. Litkowski, L.Tomotaki, and K. Ogaki, "YANG Data Model for L3VPN service delivery", draft-ietf-l3sm-l3vpn-service-model, work in progress.

[PCEP-Service-Aware] D. Dhody, et al., "Extensions to the Path Computation Element Communication Protocol (PCEP) to compute service aware Label Switched Path (LSP)", draft-ietf-pce-pcep-service-aware, work in progress.

[ACTN-PERF] Y. XU, et al., "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.

11. Contributors

Authors’ Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive Suite 173
Plano, TX 75024, USA

Email: leeyoung@huawei.com

Dhruv Dhody
Huawei Technology
Leela Palace
Bangalore, Karnataka 560008
India

Email: dhruv.dhody@huawei.com

Satish Karunanithi
Huawei Technology
Leela Palace
Bangalore, Karnataka 560008
India

Email: satish.karunanithi@gmail.com

Ricard Vilalta
Centre Tecnologic de Telecomunicacions de Catalunya (CTTC/CERCA)
Av. Carl Friedrich Gauss 7
08860 - Castelldefels
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
Expires: April 20, 2019

Y. Lee
Q. Wu
I. Busi
Huawei

D. Cecarreli
Ericsson

J. Tantsura
Apstra

October 19, 2018

Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to VPN with the Integration of Packet and Optical Networks

draft-lee-teas-actn-vpn-poi-00

Abstract

This document outlines the applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to VPN with the integration of Packet and Optical Networks (POI). It also identifies a number of scenarios where the integration of packet and optical networks is necessary to support VPN service requirements. The role of optical underlay tunnels in the POI is to support certain applications that require a hard isolation with strict deterministic latency and guaranteed constant bandwidth.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

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 so that, for example, a link in the customer's network is constructed from a path or collection of paths in the underlying networks [RFC8453].

This document outlines the applicability of ACTN to VPN with the integration of packet and optical networks which is known as the Packet and Optical Integration (POI).

It also identifies a number of scenarios where the integration of packet and optical networks is necessary to support VPN service requirements. The role of optical underlay tunnels in the POI is to support certain applications that require a hard isolation with strict deterministic latency and guaranteed constant bandwidth.

Note that there may be other transport technologies that can support the aforementioned service requirements such as TSN or Detnet to name a few. In this particular document, we are focusing on the currently available network settings where packet networks are a client layer to optical transport networks as a server layer. The principle discussed in this document can be applied to other transport technologies when they are available.

As ACTN [RFC8453] introduces the role of controllers that facilitate network operations, the scope of this document is how controllers can facilitate L2/3VPN service provisioning in the packet and optical transport networks.

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.

2. Background and Scope

One of the important functions the MDSC performs is to identify which TE Tunnels should carry the L3VPN traffic and to relay this information to the domain-level controllers to ensure proper Virtual routing and forwarding (VRF) table be populated according to the TE binding requirement for the L3VPN. This function is referred to as TE & service mapping function. The YANG model to provide TE & service mapping function is provided in [TSM]. The role of the TE-service Mapping model [TSM] 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.

The TE-Service Mapping model also provides service-TE binding information for each service instance so that proper TE tunnel should be created.

The TE binding requirement types defined in [TSM] are:

a) 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.

Under this mode, the following sub-categories can be supported:

i. 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.

ii. Hard Isolation: This is similar to the above case but without the deterministic characteristics requirements.

iii. Soft Isolation: The customer requests a VPN service using a set of TE tunnels which can be shared with other VPN services.

b) 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.

c) VN/Tunnel Modify - This mode allows the modification of the properties of the existing VN/tunnel (e.g., bandwidth).

This document addresses cases a)-i (hard isolation with deterministic latency) and a)-ii (hard isolation with non-deterministic latency). Both cases warrant consideration of optical undelay bypass tunnels to meet the service requirement.

The optical bypass tunnel could be setup via RSVP-TE signaling and thus tunnel label allocation could be done during signaling. It is also possible that PNC and MDSC coordinates to exchange the TE tunnel label information to setup this optical bypass tunnel. This document focuses on the latter case.

The multi-hop e-BGP session between ingress and egress for multi-domain case would be setup to exchange VPN routes. The rest of the forwarding action is as per the usual BGP L3VPN handling including the use of TE tunnel.

3. POI with L2/L3VPN Service Under Single Network Operator Control

This section provides a set of specific deployment scenarios for POI under single network operator control. Specifically, the following deployment scenarios are discussed in this section:

- One optical transport domain overarched by one packet domain (see Section 3.1);
- One optical transport domain overarched by multiple packet domains (see Section 3.2);
- multiple optical transport domains overarched by multiple packet domains (see Section 3.3).

All scenarios are taking place in the context of an upper layer service configuration (e.g., L3VPN) in the packet and optical transport network.

Since this document only addresses the procedure for creating optical underlay bypass tunnels, it does not affect MP-BGP MPLS operations for inter-AS scenarios as specified in [RFC4364].

3.1. POI with single packet and single optical domain

This section provides a specific deployment scenario for POI. Specifically, it provides a deployment scenario in which hierarchical controllers (an MDSC and two PNCs, one for packet and one for optical) facilitate optical bypass tunnel across the packet domain and the optical domain.

Figure 1 shows this scenario.

```
                  +----------+
                  |   MDSC   |
                  +----------+
                       |
                   ---------
                  |         |
            +--------+    +--------+
            | P-PNC  |    | O-PNC  |
            +--------+    +--------+
                 |            |
                 |            |
            +----------------------+
       CE  / PE                 PE  \    CE
       o--/---o                  o---\---o
          \   :                  :   /
           \  :    AS Domain     :  /
            +-:------------------:-+
              :               |  :
              :               |  :
            +-:------------------:-+
           /  :                  :  \
          /   o..................o   \
          \      Optical Domain      /
           \                        /
            +----------------------+
```

Figure 1. One Packet Domain and One Optical Domain

The following control sequence describes the scenario depicted in Figure 1.

a) The MDSC translates the service instance and its requirement (hard isolation with deterministic latency).
b) The MDSC computes the path if there is any feasible path to meet the requirement based on the abstracted topology at hand. Note that there would not be any tunnel in the packet domain to meet this requirement (hard isolation with deterministic latency).
c) The MDSC finds a feasible path in the optical domain.
d) The MDSC asks the optical PNC to create a tunnel for this VPN instance whose endpoints are the ingress PE and the egress PE

of the packet domain, respectively. The MDSC and Optical PNC need to maintain an instance ID for this VPN instance.

e) The MDSC asks the Packet PNC to bind a TE-tunnel label (to be allocated by the egress PE to identify the underlay optical tunnel) with the VPN ID and the Ingress and Egress interfaces of the underlay optical tunnel.

f) The PNC in turn asks the Egress PE to allocate a TE-tunnel label. The Egress PE allocates a TE-tunnel label, populates the VRF for this VPN instance, and updates the Packet PNC with the allocated TE-tunnel label. Please refer to the note below on the details of this procedure in regard to VPN binding.

   Note: There are two cases for binding network instance with the TE tunnel label:

   1. VRF instance does not exist.
   2. VRF instance has already been created.

   For case 1, the Egress PE needs to bind the TE-tunnel label and the VPN information (e.g., VPN instance name, VPN label, RD, RT, Destination IP address, etc.) and inform this binding information to the packet PNC.

g) The packet PNC informs the MDSC the allocated TE-tunnel label for the VPN instance.

h) The MDSC informs the optical PNC to bind the TE-tunnel label with the VPN instance, which has been created previously in step d).

i) The optical PNC informs this binding information (i.e., ingress/egress interfaces from packet domain and the TE-tunnel label) to the optical ingress switch.

j) The packet PNC informs the ingress PE to use the TE-label for this VPN instance. The Ingress PE populates the VRF for the VPN with the TE-label. (Note that the TE-label would need to be PUSHed over the VPN traffic).

k) When the packet arrives at the ingress PE, it recognizes the VPN instance and PUSHes the VPN label and the TE-tunnel label and forward the traffic to optical ingress switch.

l) The optical ingress switch recognizes the TE-tunnel label and encapsulate the whole data packet including TE-tunnel label into the OTN payload.

m) The optical egress switch POPs the ODU label and forwards the data packet to the packet egress PE.

n) The packet egress PE POPs the TE-tunnel label and forwards the VPN packets to the destination CE.

Note: in steps k) - l), the assumption made was that the packet ingress PE is not OTN-capable router. If the packet ingress PE support channelized OTN interfaces, the data plane behavior in steps k) and l) would change as the following:

k’) When the packets arrives at the ingress PE, it recognizes the VPN instance and PUSHes the VPN label and the TE-tunnel label and the ODU label and forward the traffic to optical ingress switch.

l’) The optical ingress switch recognizes the incoming ODU label and swap it to outgoing ODU label.

3.2. POI with multiple packet domains and single optical domain

This section provides a specific deployment scenario for POI. Specifically, it provides a deployment scenario in which hierarchical controllers (an MDSC and two packet PNCs and one optical PNC) facilitate optical bypass tunnel across the two packet domains and the optical domain.

Figure 2 shows this scenario.

```
                                     +----------+
                                     |   MDSC   |
                                     +----------+
                                           |
                        -------------------|-------------------
                       |                   |                   |
                  +--------+          +--------+          +--------+
                  | P-PNC 1|          | O-PNC  |          | P-PNC 2|
                  +--------+          +--------+          +--------+
                       |                   |                   |
                       |                   |                   |
            +----------------------+       |       +----------------------+
       CE  / PE                ASBR \      |      / ASBR              PE   \   C
E
       o--/---o                  o---\-----|-----/---o                 o----\--o
          \   :                      /     |     \                     :    /
           \  :    AS Domain 1      /      |      \      AS Domain 2   :   /
            +-:--------------------+       |       +-------------------:--+
              :                            |                           :
              :                            |                           :
            +-:--------------------------------------------------------:--+
           /  :                                                        :   \
          /   o........................................................o    \
          \                           Optical Domain                        /
           \                                                               /
            +-------------------------------------------------------------+
```

Figure 2. Two Packet Domains and One Optical Domain

The control sequence depicted in Figure 2 is same as the control sequence a)-d) in Section 3.1 with the following differences:

e) The MDSC asks the Packet PNC 2 to bind a TE-tunnel label (to be allocated by the egress PE to identify the underlay optical tunnel) with the VPN ID and the Ingress and Egress interfaces of the underlay optical tunnel.
f) The packet PNC 2 in turn asks the Egress PE to allocate a TE-tunnel label. The Egress PE allocates a TE-tunnel label, populates the VRF for this VPN instance, and updates the packet PNC 2 with the allocated TE-tunnel label. Please refer to the note below on the details of this procedure in regard to VPN binding.

   Note: There are two cases for binding network instance with the TE tunnel label:

   1. VRF instance does not exist.
   2. VRF instance has already been created.

   For case 1, the Egress PE needs to bind the TE-tunnel label and the VPN information (e.g., VPN instance name, VPN label, RD, RT, Destination IP address, etc.) and inform this binding information to the packet PNC 2.

g) The packet PNC 2 informs the MDSC the allocated TE-tunnel label for the VPN instance.
h) The MDSC informs the packet PNC 1 the allocated TE-tunnel label for the VPN instance.
i) The MDSC informs the optical PNC to bind the TE-tunnel label with the VPN instance, which has been created previously in step d).
j) The optical PNC informs this binding information (i.e., ingress/egress interfaces from packet domain and the TE-tunnel label) to the optical ingress switch.
k) The packet PNC 1 informs the ingress PE in Domain 1 to use the TE-tunnel label for this VPN instance. The Ingress PE in Domain 2 populates the VRF for the VPN and bind with the TE-tunnel label. (Note that the TE-tunnel label would need to be PUSHed over the VPN traffic).
l) When the packets arrives at the ingress PE in Domain 1, it recognizes the VPN instance and PUSHes the VPN label and the TE-tunnel label and forward the traffic to optical ingress switch.

m) The optical ingress switch recognizes the TE-tunnel label and encapsulate the whole data packet including TE-tunnel label into the OTN payload.
n) The optical egress switch POPs the ODU label and forwards the data packet to the packet egress PE.
o) The packet egress PE in Domain 2 POPs the TE-tunnel label and forwards the VPN packets to the destination CE.

Note: in steps l) - m), the assumption made was that the packet ingress PE is not OTN-capable router. If the packet ingress PE supports channelized OTN interfaces, the data plane behavior in steps l) and m) would change as the following:

l') When the packets arrives at the ingress PE, it recognizes the VPN instance and PUSHes the VPN label and the TE-tunnel label and the ODU label and forward the traffic to optical ingress switch.

m') The optical ingress switch recognizes the incoming ODU label and swap it to outgoing ODU label.

3.3. POI with multiple packet domains and multiple optical domains

This section provides a specific deployment scenario for POI. Specifically, it provides a deployment scenario in which hierarchical controllers (an MDSC and two packet PNCs and two optical PNCs) facilitate optical bypass tunnel across two packet domains and two optical domains.

Figure 3 shows this scenario.

```
                                     +----------+
                                     |   MDSC   |
                                     +----------+
                                           |
                        -------------------|-------------------
                       |                +---------+            |
                  +---------+        +---------+ 2|       +---------+
                  | P-PNC 1 |        | O-PNC 1 |--+       | P-PNC 2 |
                  +---------+        +---------+|         +---------+
                       |               |       |               |
                       |               |       |               |
            +----------------------    |       |    +---------------------+
       CE  / PE                ASBR \  |       |   / ASBR             PE   \   C
E
       o--/---o                  o---\-|-------|--/---o                o----\--o
          \   :                      / |       |  \                    :    /
```

```
           \  :    AS Domain 1      /  |       |   \      AS Domain 2  :   /
            +-:--------------------+   |       |    +------------------:--+
              :                        |       |                       :
              :                        |       |                       :
            +-:-------------------------+     +------------------------:--+
           /  :                          \   /                         :   \
          /   o.......................o...\./...o......................o    \
          \     Optical Domain 1          / \       Optical Domain 2        /
           \                             /   \                             /
            +---------------------------+     +---------------------------+
```

Figure 3. Two Packet Domains and One Optical Domain

The control sequence depicted in Figure 3 is same as the control sequence a)-c) in Section 3.1 with the following differences:

d) The MDSC asks the optical PNC 1 to create a tunnel for this VPN instance whose endpoints are the ingress PE of the packet domain 1 and the optical inter-domain interface toward optical domain 2; and the optical PNC 2 to create a tunnel for this VPN instance whose endpoints are the optical inter-domain interface from optical domain 1 and the egress PE of the packet domain 2. The MDSC and Optical PNC 1 and PNC 2 need to maintain an instance ID for this VPN instance.

e) The MDSC asks the Packet PNC 2 to bind a TE-tunnel label with the VPN ID and the Ingress and Egress interfaces of the underlay optical tunnel.

f) The packet PNC 2 in turn asks the Egress PE to allocate a TE-tunnel label. The Egress PE allocates a TE-tunnel label, populates the VRF for this VPN instance, and updates the packet PNC 2 with the allocated TE-tunnel label. Please refer to the note below on the details of this procedure in regard to VPN binding.

Note: There are two cases for binding network instance with the TE tunnel label:

1. VRF instance does not exist.
2. VRF instance has already been created.

For case 1, the Egress PE needs to bind the TE-tunnel label and the VPN information (e.g., VPN instance name, VPN label, RD, RT, Destination IP address, etc.) and inform this binding information to the packet PNC 2.

g) The packet PNC 2 informs the MDSC the allocated TE-tunnel label for the VPN instance.

h) The MDSC informs the packet PNC 1 the allocated TE-tunnel label for the VPN instance.
i) The MDSC informs the optical PNC 1 and PNC 2 to bind the TE-tunnel label with the instance, which has been created previously in step d).
j) The optical PNC 1 informs this binding information (i.e., ingress/egress interfaces from packet domain and the TE-tunnel label) to the optical ingress switch in Domain 1. Likewise, the optical PNC 2 to the optical egress switch in Domain 2. (Note we assume that the optical border switches in Domains 1 and 2 would do the normal OTN switching).
k) The packet PNC 1 informs the ingress PE in Domain 1 to use the TE-tunnel label for this VPN instance. The Ingress PE in Domain 2 populates the VRF for the VPN with the TE-label. (Note that the TE-tunnel label would need to be PUSHed over the VPN traffic).
l) When the VPN packet arrives at the ingress PE in Domain 1, it recognizes the VPN label and PUSHes the TE-tunnel label and forward the traffic to optical ingress switch in optical domain 1.
m) The optical ingress switch in optical domain 1 recognizes the TE-tunnel label and encapsulate the whole data packets including TE-tunnel label into the OTN payload.
n) The optical egress switch in optical domain 2 POPs the OTN label and forwards the data packet to the packet egress PE.
o) The packet egress PE in Domain 2 POPs the TE-tunnel label and forwards the VPN packet to the destination CE.

Note: in steps l) - m), the assumption made was that the packet ingress PE is not OTN-capable router. If the packet ingress PE supports channelized OTN interfaces, the data plane behavior in steps l) and m) would change as the following:

l') When the packets arrives at the ingress PE, it recognizes the VPN instance and PUSHes the VPN label and the TE-tunnel label and the ODU label and forward the traffic to optical ingress switch in Domain 1.

m') The optical ingress switch in Domain 1 recognizes the incoming ODU label and swap it to outgoing ODU label.

3.4. Transport of Tunnel and VPN information

The discussions in Section 3 as to the transport mechanism of the TE-tunnel label used for the underlay bypass tunnel with the VPN instance information has the undertone of making use of the controllers. Note that other mechanisms may also be possible and that such mechanisms are not precluded when solutions are sought out.

3.5. Virtual Switching Instance (VSI) Provisioning for L2VPN

The VSI provisioning for L2VPN is similar to the VPN/VRF provision for L3VPN. L2VPN service types include:

- Point-to-point Virtual Private Wire Services (VPWSs) that use LDP-signaled Pseudowires or L2TP-signaled Pseudowires [RFC6074];

- Multipoint Virtual Private LAN Services (VPLSs) that use LDP-signaled Pseudowires or L2TP-signaled Pseudowires [RFC6074];

- Multipoint Virtual Private LAN Services (VPLSs) that use a Border Gateway Protocol (BGP) control plane as described in [RFC4761] and [RFC6624];

- IP-Only LAN-Like Services (IPLSs) that are a functional subset of VPLS services [RFC7436];

- BGP MPLS-based Ethernet VPN Services as described in [RFC7432] and [RFC7209];

- Ethernet VPN VPWS specified in [RFC8214] and [RFC7432].

3.6. Inter-domain Links Update

In order to facilitate inter-domain links for the VPN, we assume that the service/network orchestrator would know the inter-domain link status and its resource information (e.g., bandwidth available, protection/restoration policy, etc.) via some mechanisms (which are beyond the scope of this document). We also assume that the inter-domain links are pre-configured prior to service instantiation.

3.7. End-to-end Tunnel Management

It is foreseen that the MDSC should control and manage end-to-end tunnels for VPNs per VPN policy.

As discussed in [ACTN-Telemetry], the MDSC is responsible to collect domain LSP-level performance monitoring data from domain controllers and to derive and report end-to-end tunnel performance monitoring information to the customer.

4. POI with VN Recursion Under Multiple Network Operators Control

[RFC8453] briefly introduces a case for the VN supplied to a customer may be built using resources from different technology layers operated by different operators. For example, one operator may run a packet TE network and use optical connectivity provided by another operator.

Figure 4, extracted from [RFC8453], shows the case where a customer asks for end-to-end connectivity between CE A and CE B, a virtual network. The customer's CNC makes a request to Operator 1's MDSC. The MDSC works out which network resources need to be configured and sends instructions to the appropriate PNCs. However, the link between Q and R is a virtual link supplied by Operator 2: Operator 1 is a customer of Operator 2.

To support this, Operator 1 has a CNC that communicates with Operator 2's MDSC. Note that Operator 1's CNC in Figure 10 is a functional component that does not dictate implementation: it may be embedded in a PNC.

```
            Virtual     CE A o===============================o CE B
            Network

                                         -----    CNC wants to create a VN
           Customer                     | CNC |   between CE A and CE B
                                         -----
                                           :
                    *********************************************** CMI
                                           :
           Operator 1         ---------------------------
                             |           MDSC            |
                              ---------------------------
                               :           :           :
                               :           :           :
                             -----   -------------   -----
                            | PNC | |     PNC     | | PNC |
                             -----   -------------   -----
                               :     :     :     :     :
           Higher              v     v     :     v     v
           Layer      CE A o---P-----Q===========R-----S---o CE B
           Network                   |     :     |
                                     |     :     |
                                     |   -----   |
                                     |  | CNC |  | CNC wants to create a VN
                                     |   -----   | between Q and R
                                     |     :     |
                    ************************************************ CMI
                                     |     :     |
           Operator 2                |   ------   |
                                     |  | MDSC |  |
                                     |   ------   |
                                     |     :     |
                                     |   -------  |
                                     |  |  PNC  | |
                                     |   -------  |
                                       \ :  :  : /
           Lower                        \v  v  v/
           Layer                         X--Y--Z
           Network

           Where

           --- is a link
           === is a virtual link
```

Figure 4: VN Recursion with Network Layers

The CMI in Figure 4 interfaces Operator 1's CNC with Operator 2's MDSC. The functions to perform and the information carried over the inter-operator CMI are identical to those of the Customer's CNC and Operator 1's MDSC. In other words, the two CMIs depicted in Figure 4 are recursive in nature.

From a data plane perspective, the interaction between operator 1 and operator 2 is similar to the POI case discussed in section 3.2 (See Figure 2) with an exception that the packet domains belong to operator 1 while optical domain to operator 2.

The control interface depicted in Figure 4 (i.e., the CNC of operator 1 and the MDSC of operator 2) should behave similarly to

4.1. Service Request Process between Multiple Operators

As discussed previously, the reclusiveness principle applies seamlessly over the two CMIs. This implies that Operator 1's MDSC needs to pass all customer service requirements transparently to Operator 2's MDSC so that Operator 2 should provision its underlay network tunnels to meet the service requirements of the original customer. The MDSC of Operator 1 should translate/map the original customer's intent and service requirements and pass down to the corresponding PNC(s) which is(are) responsible for interfacing another operator (in this example, Operator 2) that provides transport services for the segment of the customer’s VN. The PNC in turn performs as a CNC when interfacing its southbound with Operator 2’s MDSC.

It is possible that additional recursive relationships may also exist between Operator 2 and other operators.

4.2. Service/Network Orchestration of Operator 2

Operator 2 that provides transport service for Operator 1 may also need to perform service/network orchestration function just as the case for Operator 1.

From a data plane perspective, the interaction between operator 1 and operator 2 is similar to the POI case discussed in section 3.2 (See Figure 2) with an exception that the packet domains belong to operator 1 while optical domain to operator 2.

The control interface depicted in Figure 4 (i.e., the CNC of operator 1 and the MDSC of operator 2) should behave similarly to that of the MDSC and the PNCs discussed in Section 3.

5. Security Considerations

This document defines key components and interfaces for managed traffic engineered networks. Securing the request and control of resources, confidentially of the information, and availability of function, should all be critical security considerations when deploying and operating ACTN POI platforms.

Several distributed ACTN functional components are required, and implementations should consider encrypting data that flows between components, especially when they are implemented at remote nodes, regardless these data flows are on external or internal network interfaces.

From a security and reliability perspective, ACTN POI may encounter many risks such as malicious attack and rogue elements attempting to connect to various ACTN POI components. Furthermore, some ACTN POI components represent a single point of failure and threat vector, and must also manage policy conflicts, and eavesdropping of communication between different ACTN POI components.

The conclusion is that all protocols used to realize the ACTN POI should have rich security features, and customer, application and network data should be stored in encrypted data stores. Additional security risks may still exist. Therefore, discussion and applicability of specific security functions and protocols will be better described in documents that are use case and environment specific.

6. IANA Considerations

This document has no actions for IANA.

7. References

7.1. Normative References

[RFC8453] D. Ceccarelli and Y. Lee, "Framework for Abstraction and Control of Transport Networks", RFC 8453, August 2018.

7.2. Informative References

[DHODY] D. Dhody, et al., "Packet Optical Integration (POI) Use Cases for Abstraction and Control of Transport Networks (ACTN)", draft-dhody-actn-poi-use-case, 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.

[RFC4364] E. Rosen and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[ACTN-VN] Y. Lee, et al., "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

[TSM] Y. Lee, et al., "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang, work in progress.

[TE-Topo] X. Liu, et al., "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo, work in progress.

[RFC8309] Q. Wu, W. Liu, and A. Farrel, "Service Models Explained", RFC 8309, January 2018.

[L3SM]   Q. Wu, S. Litkowski, L. Tomotaki, and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, January 2018.

[L2SM] G. Fioccola (Ed), "A YANG Data Model for L2VPN Service Delivery", draft-ietf-l2sm-l2vpn-service-model, work in progress.

[ACTN-Telemetry] Y. Lee, et al.," YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics", draft-lee-teas-actn-pm-telemetry-autonomics, work in progress.

8. Contributors

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

Dhruv Dhody
Huawei
Email: dhruv.dhody@huawei.com

Haomian Zheng
Huawei
Email: haomianzheng@hauwei.com

Authors' Addresses

Young Lee
Huawei Technologies
Email: leeyoung@huawei.com

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

Italo Busi
Huawei Technologies
Email: Italo.Busi@huawei.com

Daniele Ceccarelli
Ericsson
Email: daniele.ceccarelli@ericsson.com

Jeff Tantsura
Nuage
Email: jefftant.ietf@gmail.com

TEAS WG
Internet Draft
Intended status: standard track
Expires: April 6, 2019

Young Lee
Dhruv Dhody
Huawei

Daniele Ceccarelli
Ericsson

Jeff Tantsura
Nuage

Giuseppe Fioccola
Telecom Italia

Qin Wu
Huawei

October 5, 2018

Traffic Engineering and Service Mapping Yang Model

draft-lee-teas-te-service-mapping-yang-12

Abstract

This document provides a YANG data model to map customer service models (e.g., the L3VPM Service Model) to Traffic Engineering (TE) models (e.g., the TE Tunnel or the Abstraction and Control of Traffic Engineered Networks Virtual Network model). This model is referred to as TE Service Mapping Model and is applicable 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.

Status of this Memo

Copyright Notice

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 [RFC6020] is the language of choice for documenting data models, and YANG models have been produced to allow configuration or modeling 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. [ACTN-VN-YANG] defines a YANG model and describes how customers or end-to-end orchestrators can request and/or instantiate a generic virtual network service. [ACTN-Applicability] 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.

[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.

[L2SM] 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.

[L1CSM] 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 & service related parameters that this document addresses as new and advanced parameters that are not included in generic service models. Section 3 discusses YANG modeling approach.

1.1. Terminology

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

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

tsm-types	ietf-te-service-mapping-types	[RFCXXXX}
l1	ietf-l1csm	[L1CSM]
l2vpn-svc	ietf-l2vpn-svc	[L2SM]
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-actn-vn	[ACTN-VN]
nw	ietf-network	[RFC8345]
te-types	ietf-te-types	[TE-Types]
te-topo	ietf-te-topology	[TE-Topo]
te	ietf-te	[TE-Tunnel]

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. TE & Service Related Parameters

While L1/2/3 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 the generic 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).

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 draft.

3. YANG Modeling Approach

This section provides how the TE & Service mapping parameters are supported using augmentation of the existing service models (i.e., [L1CSM], [L2SM], and [L3SM]). Figure 1 shows the scope of the Augmented LxSM Model.

```
+--------------+        +----------------------------+           +----------+
|    LxSM      |o-------|                            | . . . . . | ACTN VN  |
+--------------+ augment|                            |           +----------+
                        |                            |           +----------+
+--------------+        |    Augmented LxSM Model    | . . . . . | TE-topo  |
| TE & Service |------->|                            |           +----------+
| Mapping Types| import |                            |           +----------+
+--------------+        |                            | . . . . . | TE-tunnel|
                        +----------------------------+ reference +----------+
```

Figure 1. Augmented LxSM Model

The Augmented LxSM model (where x=1,2,3) augments the basic LxSM model while importing the common TE & Service related parameters (defined in Section 2) grouping information from TE & Service Mapping Types. The TE & 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 grouping file.

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 behavior 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 [TE-topo], 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.

3.1. Forward Compatibility

The YANG module defined in this document supports three existing service models via augmenting while sharing the common TE & 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.

4. L3VPN Architecture in the ACTN Context

Figure 2 shows the architectural context of this document referencing the ACTN components and interfaces.

```
                              +----------------------------+
                              |  Customer Service Manager  |
                              |  +-----------------------+ |
                              |  |           CNC         + |
                              |  +-+-------------------+-+ |
                              +----|-------------------|---+
                                   |                   |
                                   |CMI(Augmented L3SM)|CMI(VN)
                                   |                   |
                  +----------------|-------------------|----+
                  | +--------------|-----------------+ |    |
                  | | MDSC         |                 | |    |
                  | |              |                 | |    |
                  | |  +-----------+--------------+  | |    |
      TE-Svc-Map<------+ Service Mapping Function |  | |    |
                  | |  +-----------+--------------+  | |    |
                  | |              |                 | |    |
                  | +-------+------|-----------------+ |    |
                  |         |      |                   |    |
                  |         |      |CMI(VN)            |    |
                  |         |      |                   |    |
                  |         |   +--|-------------------|--+ |
                  |         |   |  |        MDSC       |  | |
                  |         |   | ++-------------------++ | |
                  |         |   | +   Service Mapping   +---->TE-Svc-Map
                  |         |   | ++----------+---------+ | |
                  |         |   +--|----------|-----------+ |
                  +---------|------|----------|-------------+
                            |      |          |
                            | +----+--------+ |
                            | |             | |
        MPI(VPN / TE models)| |             | |MPI(TE / L1 models)
                            | |             | |
                      +-----|-|---+   +-----|-|----+
           IP/MPLS    |  +--+-+-+ |   |  +--+-+-+  | Optical Domain
           Domain     |  | PNC1 | |   |  | PNC2 |  | Controller
           Controller |  +--+---+ |   |  +--+---+  |
                      +-----|-----+   +-----|------+
                            |               |
                            V               | SBI
                +---------------------+     |
               /    IP/MPLS Network    \    |
              +-------------------------+   |
                                            V
                                 +---------------------+
                                /    Optical Network    \
                               +-------------------------+
```

Figure 2: L3VPN Architecture from the IP+Optical Network Perspective

There are three main entities in the ACTN architecture and shown in Figure 2.

- 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 [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 1 : Mapping Between VPN Site and VNAP

5. YANG Data Trees

```
module: ietf-l1csm-te-service-mapping
  augment /l1:l1-connectivity/l1:services/l1:service:
    +-rw te-service-mapping!
  augment /l1:l1-connectivity/l1:services/l1:service:
    +-rw te-mapping
       +-rw map-type?                identityref
       +-rw availability-type?       identityref
       +-rw (te)?
          +-:(actn-vn)
          |  +-rw actn-vn-ref?       -> /vn:actn/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
  augment /l1:l1-connectivity/l1:services/l1:service/l1:endpoint-1:
    +-rw (te)?
       +-:(actn-vn)
       |  +-rw actn-vn-ref?   -> /vn:actn/ap/access-point-list/access-point-id
       +-:(te)
          +-rw ltp?           te-types:te-tp-id
  augment /l1:l1-connectivity/l1:services/l1:service/l1:endpoint-2:
    +-rw (te)?
       +-:(actn-vn)
       |  +-rw actn-vn-ref?   -> /vn:actn/ap/access-point-list/access-point-id
       +-:(te)
          +-rw ltp?           te-types:te-tp-id



module: ietf-l2sm-te-service-mapping
  augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-service:
    +-rw te-service-mapping!
  augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-service:
    +-rw te-mapping
       +-rw map-type?               identityref
       +-rw availability-type?      identityref
       +-rw (te)?
          +-:(actn-vn)
          |  +-rw actn-vn-ref?      -> /vn:actn/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
  augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site/l2vpn-svc:site-net
work-
accesses/l2vpn-svc:site-network-access:
    +-rw (te)?
       +-:(actn-vn)
       |  +-rw actn-vn-ref?   -> /vn:actn/ap/access-point-list/access-point-id
       +-:(te)
          +-rw ltp?           te-types:te-tp-id



   module: ietf-l3sm-te-service-mapping
     augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-service:
       +-rw te-service-mapping!
     augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-service:
       +-rw te-mapping
          +-rw map-type?               identityref
          +-rw availability-type?      identityref
          +-rw (te)?
             +-:(actn-vn)
             |  +-rw actn-vn-ref?      -> /vn:actn/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
     augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site/l3vpn-svc:site-
   network-accesses/l3vpn-svc:site-network-access:
       +-rw (te)?
          +-:(actn-vn)
          |  +-rw actn-vn-ref?   -> /vn:actn/ap/access-point-list/access-point-i
d
          +-:(te)
             +-rw ltp?           te-types:te-tp-id
```

6. YANG Data Models

The YANG codes are as follows:

```
   <CODE BEGINS> file "ietf-te-service-mapping-types@2018-10-05.yang"

   module ietf-te-service-mapping-types {

          namespace "urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types";

          prefix "tsm";

          import ietf-te-types {
              prefix "te-types";
          }

          import ietf-network {
              prefix "nw";
          }

          import ietf-te {
              prefix "te";
          }

          import ietf-actn-vn {
              prefix "vn";
          }

          organization
              "IETF Traffic Engineering Architecture and Signaling (TEAS)
              Working Group";

          contact
              "Editor: Young Lee <leeyoung@huawei.com>
                       Dhruv Dhody <dhruv.ietf@gmail.com>
                       Qin Wu <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.)";

          revision 2018-10-05 {
              description
                  "initial version.";
              reference
                  "TBD";
          }


          /*
           * 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 detnet-hard-isolation {
            base new;
            description
              "Hard isolation with deterministic characteristics.";
         }

         identity hard-isolation {
            base new;
            description
              "Hard isolation.";
         }

         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 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%";
         }
```

```
          /*
           * Groupings
           */

         grouping te-ref {
            description
              "The reference to TE.";
            choice te {
               description
                  "The TE";
                  case actn-vn {
                      leaf actn-vn-ref {
                          type leafref {
                              path "/vn:actn/vn:vn/vn:vn-list/vn:vn-id";
                          }
                          description
                              "The reference to ACTN VN";
                      }
                  }
                  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";
                     }
                     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";
                     }
                  }
                  case te-tunnel {
                      leaf-list te-tunnel-list {
                          type te:tunnel-ref;
                          description
                              "Reference to TE Tunnels";
                      }

                  }

              }
          }

          grouping te-endpoint-ref {
              description
                 "The reference to TE endpoints.";
              choice te {
                 description
                    "The TE";
                 case actn-vn {
                    leaf actn-vn-ref {
                          type leafref {
                              path "/vn:actn/vn:ap/vn:access-point-list"
                              + "/vn:access-point-id";
                          }
                          description
                              "The reference to ACTN VN";
                      }
                 }
                 case te {
                    leaf ltp {
                         type te-types:te-tp-id;
                         description
                             "Reference LTP in the TE-topology";
                    }
                 }
              }

          }

          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;
              }
          }

   }
   <CODE ENDS>


   <CODE BEGINS> file "ietf-l1csm-te-service-mapping@2018-10-05.yang"



      module ietf-l1csm-te-service-mapping {

          namespace "urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping";

          prefix "tm";

          import ietf-te-service-mapping-types {
             prefix "tsm-types";
         }

         import ietf-l1csm {
              prefix "l1";
          }

          organization
              "IETF Traffic Engineering Architecture and Signaling (TEAS)
              Working Group";

          contact
              "Editor: Young Lee <leeyoung@huawei.com>
                       Dhruv Dhody <dhruv.ietf@gmail.com>
                       Qin Wu <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 ";

          revision 2018-10-05 {
              description
                  "initial version.";
              reference
                  "TBD";
          }

          /*
           * Configuration data nodes
           */
          augment "/l1:l1-connectivity/l1:services/l1: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";
             }
          }

          augment "/l1:l1-connectivity/l1:services/l1:service" {
             description
               "This augment is only valid for TE mapping --
               te mapping is added";
             uses tsm-types:te-mapping;
          }

          augment "/l1:l1-connectivity/l1:services/l1:service/l1:endpoint-1" {
            description
               "This augment is only valid for TE mapping --
                endpoint-1 te-reference is added";
            uses tsm-types:te-endpoint-ref;
          }

          augment "/l1:l1-connectivity/l1:services/l1:service/l1:endpoint-2" {
            description
              "This augment is only valid for TE mapping --
              endpoint-2 te-reference is added";
            uses tsm-types:te-endpoint-ref;
          }
      }


   <CODE ENDS>


   <CODE BEGINS> file "ietf-l2sm-te-service-mapping@2018-10-05.yang"



      module ietf-l2sm-te-service-mapping {

          namespace "urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping";

          prefix "tm";

          import ietf-te-service-mapping-types {
            prefix "tsm-types";
          }

          import ietf-l2vpn-svc {
             prefix "l2vpn-svc";
          }

          organization
              "IETF Traffic Engineering Architecture and Signaling (TEAS)
              Working Group";

          contact
              "Editor: Young Lee <leeyoung@huawei.com>
                       Dhruv Dhody <dhruv.ietf@gmail.com>
                       Qin Wu <bill.wu@huawei.com>";
          description
              "This module contains a YANG module for the mapping of
              Layer 2 Service Model (L1CSM) to the TE and VN ";

          revision 2018-10-05 {
              description
                  "initial version.";
              reference
                  "TBD";
          }

          /*
           * Configuration data nodes
           */
        augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-servi
ce" {
            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";
             }
        }

        augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-servi
ce" {
            description
              "This augment is only valid for TE mapping --
               te mapping is added";
            uses tsm-types:te-mapping;
        }

        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 --
                  network-access te-reference is added";
              uses tsm-types:te-endpoint-ref;
          }
      }
```

   <CODE ENDS>

   <CODE BEGINS> file "ietf-l3sm-te-service-mapping@2018-10-05.yang"

```
   module ietf-l3sm-te-service-mapping {

       namespace "urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping";

       prefix "tm";

       import ietf-te-service-mapping-types {
         prefix "tsm-types";
       }

       import ietf-l3vpn-svc {
          prefix "l3vpn-svc";
       }

       organization
           "IETF Traffic Engineering Architecture and Signaling (TEAS)
           Working Group";

       contact
           "Editor: Young Lee <leeyoung@huawei.com>
                    Dhruv Dhody <dhruv.ietf@gmail.com>
                    Qin Wu <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 ";

       revision 2018-10-05 {
           description
               "initial version.";
           reference
               "TBD";
       }

       /*
        * Configuration data nodes
        */
       augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-servic
e" {
         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";
         }
       }

       augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-servic
e" {
          description
            "This augment is only valid for TE mapping --
             te mapping is added";
          uses tsm-types:te-mapping;
       }

       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 --
             network-access te-reference is added";
         uses tsm-types:te-endpoint-ref;
       }
   }


   <CODE ENDS>
```

7. Security

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]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF 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.

8. IANA Considerations

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

```
--------------------------------------------------------------------
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-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-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-l3sm-te-service-mapping
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-service-mapping-types
```

```
   namespace:    urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-
   types
   reference:    RFC XXXX (TDB)
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   name:         ietf-l1csm-te-service-mapping
   namespace:    urn:ietf:params:xml:ns:yang:ietf-l1cms-te-service-
   mapping
   reference:    RFC XXXX (TDB)
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   name:         ietf-l2sm-te-service-mapping
   namespace:    urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-
   mapping
   reference:    RFC XXXX (TDB)
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   name:         ietf-l3sm-te-service-mapping
   namespace:    urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-
   mapping
   reference:    RFC XXXX (TDB)
   --------------------------------------------------------------------
```

9. Acknowledgements

We thank Diego Caviglia and Igor Bryskin for useful discussions and motivation for this work.

10. References

10.1. Informative References

[RFC4110] R. Callon and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, July 2005.

[RFC6020] M. Bjorklund, Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC8309] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", RFC 8309, January 2018.

[RFC8199] D. Bogdanovic, B. Claise, and C. Moberg, "YANG Module Classification", RFC 8199, July 2017.

[Netconf] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.

[RFC8453] D. Cecarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", RFC 8453, August 2018.

[TE-Topo] X. Liu, et. al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.

[TE-Tunnel] T. Saad (Editor), "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.

[TE-Types] T. Saad (Editor), "Traffic Engineering Common YANG Types", draft-ietf-teas-yang-te-types, work in progress.

[ACTN-VN-YANG] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

[ACTN-Applicability] Y. Lee, et al, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks, draft-ietf-teas-actn-yang, work in progress.

[RFC8299] Q. Wu, S. Litkowski, L.Tomotaki, and K. Ogaki, "YANG Data Model for L3VPN service delivery", RFC 8299, January 2018.

[L2SM] B. Wen, et al, "A YANG Data Model for L2VPN Service Delivery", draft-ietf-l2sm-l2vpn-service-model, work in progress.

[L1CSM] G. Fioccola, et al, "A Yang Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang, work in progress.

11. Contributors

Adrian Farrel
Old Dog Consulting

Email: adrian@olddog.co.uk

Italo Busi
Huawei Technologies

Email: Italo.Busi@huawei.com

Authors' Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive
Plano, TX 75023, USA
Phone: (469)277-5838

Email: leeyoung@huawei.com

Dhruv Dhody
Huawei Technologies

Email: dhruv.ietf@gmail.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Jeff Tantsura
Nuage

EMail: jefftant@gmail.com

Giuseppe Fioccola
Telecom Italia
Email: giuseppe.fioccola@telecomitalia.it

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

TEAS Working Group Z. Li
Internet-Draft D. Dhody
Intended status: Informational H. Chen
Expires: March 7, 2019 Huawei Technologies
September 3, 2018

Hierarchy of IP Controllers (HIC)
draft-li-teas-hierarchy-ip-controllers-01

Abstract

This document describes the interactions between various IP controllers in a hierarchical fashion to provide various IP services. It describes how the Abstraction and Control of Traffic Engineered Networks (ACTN) framework is applied to the Hierarchy of IP controllers (HIC) as well as document the interactions with other protocols like BGP, Path Computation Element Communication Protocol (PCEP) to provide end to end dynamic services spanning multiple domains and controllers (e.g. Layer 3 Virtual Private Network (L3VPN), Seamless MPLS etc).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Software-Defined Networking (SDN) refers to a separation between the control elements and the forwarding components so that software running in a centralized system called a controller, can act to program the devices in the network to behave in specific ways. A required element in an SDN architecture is a component that plans how the network resources will be used and how the devices will be programmed. It is possible to view this component as performing specific computations to place flows within the network given knowledge of the availability of network resources, how other forwarding devices are programmed, and the way that other flows are routed. The Application-Based Network Operation (ABNO) [RFC7491] describes how various components and technologies fit together.

A domain [RFC4655] is any collection of network elements within a common sphere of address management or path computation responsibility. Specifically within this document we mean a part of an operator’s network that is under common management. Network elements will often be grouped into domains based on technology types, vendor profiles, and geographic proximity and under a domain controller.

Multiple such domains in the network are interconnected, and a path is established through a series of connected domains to form an end-to-end path over which various services are offered. Each domain is under the control of the domain controller (or lower-level controller), and a "super controller" (or high-level controller) takes responsibility for a high-level view of the network before distributing tasks to domain controllers (or lower-level controllers). It is possible for each of the domain to use a different tunneling mechanism (eg RSVP-TE, Segment Routing (SR) etc).

[RFC8453] describes the framework for Abstraction and Control of Traffic Engineered Networks (ACTN) as well as a set of management and control functions used to operate multiple TE networks. This documents would apply the ACTN principles to Hierarchy of IP controllers (HIC) and focus on the applicability and interactions with other protocol and technologies (specific to IP packet domains).

Sometimes, service (such as Layer 3 Virtual Private Network (L3VPN), Layer 2 Virtual Private Network (L2VPN), Ethernet VPN (EVPN), Seamless MPLS) require sites attached to different domains (under the control of different domain controller) to be interconnected as part of the VPN service. This require multi-domain coordination between domain controllers to compute and setup E2E path for the VPN service.

This document describes the interactions between various IP controllers in a hierarchical fashion to provide various IP services. It describes how the Abstraction and Control of Traffic Engineered Networks (ACTN) framework is applied to the Hierarchy of IP controllers (HIC) as well as document the interactions with control plane protocols (like BGP, Path Computation Element Communication Protocol (PCEP)) and management plane aspects (Yang models) to provide end to end dynamic services spanning multiple domains and controllers (e.g. L3VPN, Seamless MPLS etc).

2. Overview

Figure 1 show examples of multi-domain IP domains under hierarchy of IP controllers.

```
                             |
                       +------------+
                       |  SuperCo   |
                       +------------+
                             |
             ----------------------------------
             |               |                |
      +------------+   +------------+   +------------+
      |   DoCo#1   |   |   DoCo#2   |   |   DoCo#3   |
      +------------+   +------------+   +------------+

      +--Domain#1--+   +--Domain#2--+   +--Domain#3--+
      |            |   |            |   |            |
      |     B------+---+---D-----E--+---+------J     |
      |    /       |   |    \   /   |   |       \    |
      |   /        |   |     \ /    |   |        \   |
      |  A         |   |      H     |   |         L  |
      |   \        |   |     / \    |   |        /   |
      |    \       |   |    /   \   |   |       /    |
      |     C------+---+---F-----G--+---+------K     |
      |            |   |            |   |            |
      +------------+   +------------+   +------------+
```

Figure 1: Example: Hierarchy of IP controllers (HIC)

The IP "Super Controller" receives request from the network/service orchestrator to setup dynamic services spanning multiple domains. The IP "Super Controller" breaks down and assigns tasks to the domain controllers, responsible for communicating to network devices in the domain. It further coordinates between the controller to provide a unified view of the multi-domain network.

2.1. Mapping to ACTN

As per [RFC8453], ACTN has following main functions -

- Multi-domain coordination

- Virtualization/Abstraction

- Customer mapping/translation

- Virtual service coordination

These functions are part of Multi Domain Service Coordinator (MDSC) and/or Provisioning Network Controller (PNC). Further these functions are part of the controller / orchestrator.

The HIC is an instantiation of ACTN framework for IP packet network. The IP domain (lower-level) controllers implements the PNC functionalities for configuring, controlling and monitoring the IP domain. The "super controller" (high-level controller) implements the MDSC functionalities for coordination between multiple domains as well as maintaining an abstracted view of multiple domains. It also takes care of the service related functionalities of customer mapping/translation and virtual service coordination.

The ACTN functions are part of the IP controllers and responsible for the TE topology and E2E path computation/setup. There are other functions along with ACTN that are needed to manage multiple IP domain networks.

2.2. Interface between Super Controller and Domain Controller in HIC

The interaction between super controller and domain controller in HIC is a combination of Control Plane and Management Plane interface as shown in Figure 2. BGP [RFC4271] and PCEP [RFC5440] are example of the control plane interface; where as NETCONF [RFC6241] and RESTCONF [RFC8040] are example of management plane interface.

```
      +----------------------------------------------+
      |                Super Controller              |
      |                                              |
      |                                              |
      +------------------*------#---------------------+
                         *      #
                         *      #
                      *************************
                      *         #             *
                ######*###############        *
                #     *              #        *
      +---------#-----*--+        +--#--------*------+
      | Domain           |        | Domain           |
      | Controller       |        | Controller       |
      +--#------------*--+        +--#------------*--+
         #            *              #            *
         #            *              #            *


       * -> Control Plane Interface
       # -> Management Plane Interface
```

Figure 2: Interface between Super Controller and Domain Controller

Note that ACTN's MDSC-PNC Interface (MPI) could be implemented via management plane interface using Yang models [I-D.ietf-teas-actn-yang] or via PCEP control plane interface [I-D.ietf-pce-applicability-actn].

3. Key Concepts

3.1. Topology

The Domain Controller is expected to be aware of the topology of the network devices in its domain. The domain controller could participate in the IGP ([RFC3630] and [RFC5305]) or use BGP-LS [RFC7752] by which link-state and TE information is collected and shared with domain controller using the BGP routing protocol.

An alternate approach would be to rely on the management plane interface which uses the YANG model for network/TE Topology as per [RFC8345] and [I-D.ietf-teas-yang-te-topo].

The domain controller is expected to share the domain topology to the Super Controller as aligned to ACTN (where PNC abstract the topology towards MDSC). A level of abstraction is usually applied while presenting the topology to a higher level controller. Topology abstraction is described in [RFC7926] as well as [RFC8453]. BGP-LS, PCEP-LS [I-D.dhodylee-pce-pcep-ls] or management plane interface based on the abstracted network/TE Topology could be used to carry the abstract topology to the super-controller. At minimum the border nodes and inter-domain links are exposed to the super-controller.

Further [RFC8453] defines three types of topology abstraction - (1) Native/White Topology; (2) Black Topology; and (3) Grey Topology. Based on the local policy, the domain controller would share the domain topology to the Super Controller based on the abstraction type. Note that any of the control plane or management plane mechanism could be used to carry abstracted domain topology. The Super Controller’s MDSC function is expected to manage a E2E topology by coordinating the abstracted domain topology received from the domain controllers.

3.2. Path Computation/Path instantiation

The Domain Controller is responsible for computing and setup of path when the source and destination is in the same domain, otherwise the Super Controller coordinates the multi-domain path computation and setup with the help of the domain controller. This is aligned to ACTN.

PCEP [RFC5440] provides mechanisms for Path Computation Elements (PCEs) [RFC4655] to perform path computations in response to Path Computation Clients (PCCs) requests. Since then, the role and function of the PCE has grown to allow delegated control [RFC8231] and PCE-initiated use of network resources [RFC8281].

Further, [RFC6805] and [I-D.ietf-pce-stateful-hpce] describes a hierarchy of PCE with Parent PCE coordinating multi-domain path computation function between Child PCE(s). This fits well with HIC as described in this document.

Note that a management plane interface which uses the YANG model for path computation/setup ([I-D.ietf-teas-yang-path-computation] and [I-D.ietf-teas-yang-te]) could be used in place of PCEP.

In case there is a need to stitch per domain tunnels into an E2E tunnel, mechanism are described in [I-D.lee-pce-lsp-stitching-hpce] and [I-D.dugeon-pce-stateful-interdomain].

3.3. BGP considerations

[RFC4456] describes the concept of route-reflection where a "route reflector" (RR) reflects the routes to avoid full mesh connection between Internal BGP (IBGP) peers. The IP domain controller can play the role of RR in its domain. The super controller can further act as RR to towards the domain controller.

BGP can provide routing policies for the traffic management, like route preference, AS-path filter policy, IP-prefix filter policy and route aggregation. The controller can distribute these BGP Policy into the routers in a single IP domain. For the scenario of multiple domains, the super controller can distribute per BGP Policy into each IP domain controller. Then the IP domain controller trickles down the BGP Policy to the network devices.

[RFC5575] describes the concept of BGP Flowspec that can be used to distribute traffic flow specifications. A flow specification is an n-tuple consisting of several matching criteria that can be applied to IP traffic. The controller can originate the flow specifications and disseminate to the routers. The flow action includes the redirection to a specific TE tunnel. Also, the IP domain controller could be responsible for collecting the flow sample in its domain and the super controller can act as the Flow Analysis Server.

[RFC7854] describes the BGP Monitoring Protocol (BMP) to monitor BGP sessions. BMP is used to obtain the route views with a flexible way. In the fashion of hierarchical architecture, the IP domain controller can be used as the domain Monitoring Station. Meanwhile, the super controller is responsible for a high-level view of the global network state.

4. VPN Service

4.1. Seamless MPLS

Seamless MPLS [I-D.ietf-mpls-seamless-mpls] describes an architecture which can be used to extend MPLS networks to integrate access and core/aggregation networks into a single MPLS domain.In the seamless MPLS for mobile backhaul, since there are multiple domains including the core network and multiple mobile backhaul networks, for each domain there is a domain controller. In order to implement the end-to-end network service provision, there should be coordination among multiple domain controllers.

```
                                |
                                |
                                |
                             +----------+
              |--------------|Super     |---------|
              |              |Controller|         |
              |              +----------+         |
              |                 |                 |
              |                 |                 |
              |                 |                 |
          +------+           +------+          +------+
     |----|DoCo  |----|  |---|DoCo  |--|  |----|DoCo  |---|
     |    |#X1   |    |  |   |#Y    |  |  |    |#X2   |   |
     |    +------+    |  |   +------+  |  |    +------+   |
     |                |  |             |  |               |
     |                |  |             |  |               |
     |                |  |             |  |               |
     |               +----+           +----+              |
     |           ....|ABR1|...........|ABR3|....          |
   +----+   .....    +----+           +----+    .....   +----+
   | PE |...                                         ...| PE |
   +----+   .....                                       +----+
                 ....+----+           +----+    .....
                     |ABR2|...........|ABR4|....
                     +----+           +----+

     |      IGP-X1    |      IGP-Y      |       IGP-X2     |
     |       (MBH)    |      (Core)     |       (MBH)      |
     |                |                 |                  |
     |-----BGP LSP-----|-----BGP LSP----|------BGP LSP-----|
     |                |                 |                  |
     |---LDP/TE LSP----|----LDP/TE LSP--|-----LDP/TE LSP---|
     |                |                 |                  |
```

Figure 3: Seamless MPLS

Super Controller is responsible for setting the seamless MPLS service. It should break down the service model to network configuration model [RFC8309] and the domain controller further break it to the device configuration model to the PE/ASBR to make the E2E seamless MPLS service. The selection of appropriate ASBRs and handling of intra-domain tunnels is coordinated by the Super Controller in the similar fashion as shown in Section 4.2.

By enabling BGP sessions between Domain Controller and Super Controller, BGP labeled routes can also be learned at Super

Controller. As Super Controller is aware of the (abstract) topology, it could make intelligent decisions regarding E2E BGP LSP to optimize based on the overall traffic information.

4.2. L3VPN

A Layer 3 IP VPN service is a collection of sites that are authorized to exchange traffic between each other over a shared IP infrastructure. [RFC4110] provides a framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs). [RFC8299] provides a L3VPN service delivery YANG model for PE-based VPNs. The Super controller is expected to implement the L3SM model and translate it to network models towards the domain controller, which in turn translate it to the device model. See [RFC8309] for more details.

```
                                     | L3SM
                                     V
                          +--------------------+
                          |  Super Controller  |
                          +--------------------+
                                     |
                     +-------------------------------+
                     |                               |
                     V                               V
                  +--------+                   +--------+
                  | DoCo#1 |                   | DoCo#2 |
                  |        |                   |        |
                  +--------+                   +--------+

            CE                                                 CE
             \     AS 100                          AS 200       /
              \                                               /
               A----B----C----ASBR1------ASBR2----D----E----F
              /    /    /       /          /     /    /    /
             /    /    /       /          /     /    /    /
      CE----G----H----I----ASBR3------ASBR4----J----K----L------CE
```

Figure 4: L3VPN

Based on the user data in L3SM model, the network configurations need to be trickle down to the network device to setup the L3VPN.

Based on the QoS or Policy requirement for the L3VPN service, the Super Controller may -

o Set the tunnel selection policy at the PE/ASBR routers so that they could select the existing tunnels

- Select an existing tunnels at the controller level and bind it to the VPN service

- Initiate the process of creating a new tunnel based on the QoS requirement and bind it the VPN service

- Initiate the process of creating a new tunnel based on the the policy

Refer [I-D.lee-teas-te-service-mapping-yang] for more details from ACTN perspective.

Apart from the Management plane interface based on respective YANG models, the control plane interface PCEP could be used for path computation and setup.

4.3. L2VPN and EVPN service

There are two fundamentally different kinds of Layer 2 VPN service that a service provider could offer to a customer: Virtual Private Wire Service (VPWS) and Virtual Private LAN Service (VPLS) [RFC4664]. A VPWS is a VPN service that supplies an L2 point-to-point service. A VPLS is an L2 service that emulates LAN service across a Wide Area Network (WAN). A BGP MPLS-based Ethernet VPN (EVPN) [RFC7432] addresses some of the limitations when it comes to multihoming and redundancy, multicast optimization, provisioning simplicity, flow-based load balancing, and multipathing etc.

The handling of L2VPN/EVPN service is done in a similar fashion as shown in Section 4.2.

5. Possible Features/Extensions

This sections list some of the possible features or protocol extensions that could be worked on to deploy HIC in a multi-domain packet network.

1. Simplify the initial configurations needed to setup the relationship between the super controller and the domain controllers. Note that this could be done via exchanges during initial session establishment, discovery via other protocols, service discovery (such as DNS) etc.

2. The (higher-level controller, lower-level controller) relationship or the the role of the controller.

3. The learning and handling of various capabilities of the Super Controller and Domain Controller.

4. Handling of multiple instances of controller at each level for high availability.

[Editor’s Note - This list is expected to be updated in next version with more details]

6. Other Considerations

6.1. Control Plane

6.1.1. PCE / PCEP

The Path Computation Element communication Protocol (PCEP) [RFC5440] provides mechanisms for Path Computation Elements (PCEs) [RFC4655] to perform path computations in response to Path Computation Clients (PCCs) requests.

The ability to compute shortest constrained TE LSPs in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains has been identified as a key motivation for PCE development.

A stateful PCE [RFC8231] is capable of considering, for the purposes of path computation, not only the network state in terms of links and nodes (referred to as the Traffic Engineering Database or TED) but also the status of active services (previously computed paths, and currently reserved resources, stored in the Label Switched Paths Database (LSPDB).

[RFC8051] describes general considerations for a stateful PCE deployment and examines its applicability and benefits, as well as its challenges and limitations through a number of use cases.

[RFC8231] describes a set of extensions to PCEP to provide stateful control. A stateful PCE has access to not only the information carried by the network’s Interior Gateway Protocol (IGP), but also the set of active paths and their reserved resources for its computations. The additional state allows the PCE to compute constrained paths while considering individual LSPs and their interactions. [RFC8281] describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model.

[RFC8231] also describes the active stateful PCE. The active PCE functionality allows a PCE to reroute an existing LSP or make changes to the attributes of an existing LSP, or a PCC to delegate control of specific LSPs to a new PCE.

Computing paths across large multi-domain environments require special computational components and cooperation between entities in different domains capable of complex path computation. The PCE provides an architecture and a set of functional components to address this problem space. A PCE may be used to compute end-to-end paths across multi-domain environments using a per-domain path computation technique [RFC5152]. The Backward recursive PCE based path computation (BRPC) mechanism [RFC5441] defines a PCE-based path computation procedure to compute inter-domain constrained MPLS and GMPLS TE networks. However, both per-domain and BRPC techniques assume that the sequence of domains to be crossed from source to destination is known, either fixed by the network operator or obtained by other means.

[RFC6805] describes a Hierarchical PCE (H-PCE) architecture which can be used for computing end-to-end paths for inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs) when the domain sequence is not known. Within the Hierarchical PCE (H-PCE) architecture, the Parent PCE (P-PCE) is used to compute a multi-domain path based on the domain connectivity information. A Child PCE (C-PCE) may be responsible for a single domain or multiple domains, it is used to compute the intra-domain path based on its domain topology information.

[I-D.ietf-pce-stateful-hpce] state the considerations for stateful PCE(s) in hierarchical PCE architecture. In particular, the behavior changes and additions to the existing stateful PCE mechanisms (including PCE- initiated LSP setup and active PCE usage) in the context of networks using the H-PCE architecture.

[I-D.ietf-pce-applicability-actn] examines the applicability of PCE/ PCEP to the ACTN framework in detail.

6.1.2. BGP

[RFC7752] describes a mechanism by which link-state and TE information can be collected from networks and shared with external components using the BGP routing protocol. This is achieved using a new BGP Network Layer Reachability Information (NLRI) encoding format and a new BGP path attribute (BGP-LS attribute) that carries link, node, and prefix parameters and attributes.

BGP-LS is a new approach to collect the network topology information. It is an extension to BGP for distribution the network's link-state (LS) topology to external entities, such as the SDN controller. Network's link-state topology consists of nodes and links and a set of attributes. The link-state topology is distributed among the IGP domain. The specific protocol used in an IGP domain could be OSPF

[RFC2238] or IS-IS [ISO10589]. Note that, the detailed link-state models of these two protocols are not identical. Therefore, BGP-LS can provide a more abstract topology model which can map the IGP models.

The domain controller acts as a consumer to collect the domain's link-state and TE information via BGP-LS. The domain controller would usually abstract the domain information towards the super-controller and further send it via BGP-LS.

BGP-Flowspec is a solution devised for preventing distributed Denial-of-service (DDoS) attack. BGP-Flowspec distributes specification rules into neighbors. [RFC5575] defines a new BGP NLRI encoding format that can be used to distribute traffic flow specifications. Additionally, it defines two applications of that encoding format: one that can be used to automate inter-domain coordination of traffic filtering, such as what is required in order to mitigate DDoS attacks; and a second application to provide traffic filtering in the context of BGP/MPLS VPN service.

The IP domain controller can act as the traffic sampling node. The super controller can act as the traffic analysis server. When the super controller finds the attack happened, the super controller should distribute the flow rules to associated IP domain controllers. And each IP domain controller should distribute the flow rules into the ingress routers. Additionally one of the actions taken could be "redirect" where flow could be redirected to the TE tunnels created by the controller.

[I-D.luo-grow-bgp-controller-based-ts] describes the traffic steering based on BGP controller. The traditional method for traffic steering depends on static configuration which is time consuming and inefficient. With the hierarchical IP controller, the IP domain controller can have the domain network topology view and routing information while the super controller can have the global network topology view and routing information. The super controller can compute the end-to-end paths to satisfy the differentiated service requirement. The IP domain controller may be used to distribute the routing policy into the routers. BGP policy varies in many aspects. Its goal is to meet the customer application and connectivity requirement, and specific service transport needs. So the super BGP controller is responsible for the coordination of multiple domain BGP Policy. And then distribute Policy to related IP domain controller. The IP domain controller is responsible for distributing the policy to its network nodes.

[I-D.ietf-idr-rtc-hierarchical-rr] describes the route target (RT) constrain mechanism in the hierarchical route reflection (RR) scenario. [RFC4684] describes the route target constrain mechanism to build a route distribution graph in order to restrict the propagation of Virtual Private Network (VPN) routes. [I-D.ietf-idr-rtc-hierarchical-rr] proposes a solution to address the RT constrain issue in the hierarchical RR scenarios. The super controller corresponding to higher level RR can receive the RT-constrain routes from the lower level RR, which is acted by the IP domain controller. The higher level RR will select one of the received routes as the best route. then it should advertise the best route to all the lower level RR to build the route distribution graph. This fits well with the HIC as described in this document.

6.2. Management Plane

6.2.1. YANG Models

This is an non-exhaustive list of possible yang models developed or in-development that could be used for HIC.

Topology: [RFC8345] defines a generic YANG data model for network topology. [I-D.ietf-teas-yang-te-topo] defines a YANG data model for representing, retrieving and manipulating Traffic Engineering (TE) Topologies.

Tunnel: [I-D.ietf-teas-yang-te] defines a YANG data model for the configuration and management of Traffic Engineering (TE) interfaces, tunnels and Label Switched Paths (LSPs).

L3VPN: The Layer 3 service model (L3SM) is defined in [RFC8299], which is a YANG data model that can be used for communication between customers and network operators and to deliver a Layer 3 provider-provisioned VPN service. [I-D.ietf-bess-l3vpn-yang] defines a YANG data model that can be used to configure and manage BGP Layer 3 VPNs at the device. Note that a network configuration model at the Domain Controller level needs to be developed.

L2VPN/EVPN: [I-D.ietf-l2sm-l2vpn-service-model] defines a YANG data model that can be used to configure a Layer 2 Provider Provisioned VPN service. This model is intended to be instantiated at management system to deliver the overall service. [I-D.ietf-bess-l2vpn-yang] and [I-D.ietf-bess-evpn-yang] defines a YANG data model to configure and manage L2VPN and EVPN service respectively. Note that a network configuration model at the Domain Controller level needs to be developed.

OAM: TBD

BGP Policy: [I-D.ietf-idr-bgp-model] defines a YANG data model that can be used to configure BGP Policy based on data center, carrier and content provider operational requirements. The model is intended to be vendor-neutral, in order to allow operators to manage BGP configuration in heterogeneous environments with routers supplied by multiple vendors. Note that a network configuration model at the Domain Controller level needs to be developed.

BGP Flowspec: [I-D.wu-idr-flowspec-yang-cfg] defines a YANG data model for Flow Specification implementations. The configuration data is described as flow specification rules that can be distributed as BGP NLRI to a network element. The rules can be used to filter Distributed Denial of Service attacks (DDoS) besides other use cases. Note that a network configuration model at the Domain Controller level needs to be developed.

[Editor's Note - the above list should be extended.]

6.2.2. Protocol Considerations

The Network Configuration Protocol (NETCONF) [RFC6241] provides mechanisms to install, manipulate, and delete the configuration of network devices. The RESTCONF [RFC8040] describes an HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG, using the data-store concepts defined in NETCONF.

Some other mechanism like gRPC/gNMI could also be used between controllers using the same YANG data models.

7. IANA Considerations

There are no IANA concerns in this document.

8. Security Considerations

There are no new security concerns in this document.

9. Acknowledgments

10. Contributing Authors

Dailongfei (Larry)
Huawei Technologies,
Beijing, China

Email: larry.dai@huawei.com

11. References

11.1. Normative References

[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>.

11.2. Informative References

[RFC2238] Clouston, B., Ed. and B. Moore, Ed., "Definitions of Managed Objects for HPR using SMIv2", RFC 2238, DOI 10.17487/RFC2238, November 1997, <https://www.rfc-editor.org/info/rfc2238>.

[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>.

[RFC4110] Callon, R. and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, DOI 10.17487/RFC4110, July 2005, <https://www.rfc-editor.org/info/rfc4110>.

[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>.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006, <https://www.rfc-editor.org/info/rfc4456>.

[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>.

[RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <https://www.rfc-editor.org/info/rfc4664>.

[RFC4684] Marques, P., Bonica, R., Fang, L., Martini, L., Raszuk, R., Patel, K., and J. Guichard, "Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)", RFC 4684, DOI 10.17487/RFC4684, November 2006, <https://www.rfc-editor.org/info/rfc4684>.

[RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008, <https://www.rfc-editor.org/info/rfc5152>.

[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>.

[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>.

[RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <https://www.rfc-editor.org/info/rfc5441>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

[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>.

[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>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[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>.

[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>.

[RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <https://www.rfc-editor.org/info/rfc7854>.

[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>.

[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>.

[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>.

[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>.

[I-D.ietf-teas-actn-yang]
Lee, Y., zhenghaomian@huawei.com, z., 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-02 (work in progress), August 2018.

[I-D.ietf-pce-applicability-actn]
Dhody, D., Lee, Y., and D. Ceccarelli, "Applicability of Path Computation Element (PCE) for Abstraction and Control of TE Networks (ACTN)", draft-ietf-pce-applicability-actn-06 (work in progress), June 2018.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-16 (work in progress), July 2018.

[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-18 (work in progress), June 2018.

[I-D.ietf-pce-stateful-hpce]
Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., King, D., and O. Dios, "Hierarchical Stateful Path Computation Element (PCE).", draft-ietf-pce-stateful-hpce-05 (work in progress), June 2018.

[I-D.ietf-teas-yang-path-computation]
Busi, I., Belotti, S., Lopezalvarez, V., Dios, O., Sharma, A., Shi, Y., Vilata, R., and K. Sethuraman, "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation-02 (work in progress), June 2018.

[I-D.ietf-mpls-seamless-mpls]
Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.

[I-D.ietf-bess-evpn-yang]
Brissette, P., Shah, H., Hussain, I., Tiruveedhula, K., and J. Rabadan, "Yang Data Model for EVPN", draft-ietf-bess-evpn-yang-05 (work in progress), February 2018.

[I-D.ietf-bess-l2vpn-yang]
Shah, H., Brissette, P., Chen, I., Hussain, I., Wen, B., and K. Tiruveedhula, "YANG Data Model for MPLS-based L2VPN", draft-ietf-bess-l2vpn-yang-08 (work in progress), February 2018.

[I-D.ietf-bess-l3vpn-yang]
Jain, D., Patel, K., Brissette, P., Li, Z., Zhuang, S., Liu, X., Haas, J., Esale, S., and B. Wen, "Yang Data Model for BGP/MPLS L3 VPNs", draft-ietf-bess-l3vpn-yang-03 (work in progress), April 2018.

[I-D.ietf-l2sm-l2vpn-service-model]
Wen, B., Fioccola, G., Xie, C., and L. Jalil, "A YANG Data Model for L2VPN Service Delivery", draft-ietf-l2sm-l2vpn-service-model-10 (work in progress), April 2018.

[I-D.dhodylee-pce-pcep-ls]
Dhody, D., Lee, Y., and D. Ceccarelli, "PCEP Extension for Distribution of Link-State and TE Information.", draft-dhodylee-pce-pcep-ls-11 (work in progress), June 2018.

[I-D.lee-teas-te-service-mapping-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Tantsura, J., Fioccola, G., and Q. Wu, "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang-10 (work in progress), August 2018.

[I-D.lee-pce-lsp-stitching-hpce]
Lee, Y., Dhody, D., and D. Ceccarelli, "PCEP Extensions for Stitching LSPs in Hierarchical Stateful PCE Model", draft-lee-pce-lsp-stitching-hpce-01 (work in progress), December 2017.

[I-D.dugeon-pce-stateful-interdomain]
Dugeon, O., Meuric, J., Lee, Y., Dhody, D., and D. Ceccarelli, "PCEP Extension for Stateful Inter-Domain Tunnels", draft-dugeon-pce-stateful-interdomain-01 (work in progress), July 2018.

[I-D.luo-grow-bgp-controller-based-ts]
Luo, Y., Ou, L., Huang, X., Zhuang, S., and Z. Li, "Traffic Steering Based on BGP Controller", draft-luo-grow-bgp-controller-based-ts-00 (work in progress), March 2018.

[I-D.ietf-idr-rtc-hierarchical-rr]
Dong, J., Chen, M., and R. Raszuk, "Extensions to RT-Constrain in Hierarchical Route Reflection Scenarios", draft-ietf-idr-rtc-hierarchical-rr-03 (work in progress), July 2017.

[I-D.ietf-idr-bgp-model]
Patel, K., Jethanandani, M., and S. Hares, "BGP Model for Service Provider Networks", draft-ietf-idr-bgp-model-03 (work in progress), May 2018.

[I-D.wu-idr-flowspec-yang-cfg]
Wu, N., Zhuang, S., and A. Choudhary, "A YANG Data Model for Flow Specification", draft-wu-idr-flowspec-yang-cfg-02 (work in progress), October 2015.

[ISO10589]
ISO, "Intermediate system to Intermediate system routing information exchange protocol for use in conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, 1992.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

EMail: lizhenbin@huawei.com

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

EMail: dhruv.ietf@gmail.com

Huaimo Chen
Huawei Technologies
Boston, MA
USA

EMail: huaimo.chen@huawei.com

TEAS Working Group
Internet Draft
Category: Informational

Haomian Zheng
Xianlong Luo
Huawei Technologies
Yang Zhao
China Mobile
Yunbin Xu
CAICT
Sergio Belotti
Dieter Beller
Nokia

Expires: April 22, 2019
October 22, 2018

Interworking of GMPLS Control and Centralized Controller System

draft-zheng-teas-gmpls-controller-inter-work-01

Abstract

Generalized Multi-Protocol Label Switching (GMPLS) control allows each network element (NE) to perform local resource discovery (e.g., LMP), routing (e.g., OSPF-TE) and signaling (e.g., RSVP-TE) in a distributed manner.

On the other hand, with the development of software-defined transport networking technology, a set of NEs can be controlled via centralized controller hierarchies to address the issue from multi-domain, multi-vendor and multi-technology. An example of such centralized architecture is ACTN controller hierarchy [RFC8453].

Instead of competing with each other, both the distributed and the centralized control plane have their own advantage, and should be complementary in the system. This document describes how the GMPLS distributed control plane can interwork with a centralized controller system in a transport network.

Status of this Memo

Copyright Notice

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].

Table of Contents

1. Introduction

Generalized Multi-Protocol Label Switching (GMPLS) [RFC3945] extends MPLS to support different classes of interfaces and switching capabilities such as Time-Division Multiplex Capable (TDM), Lambda Switch Capable (LSC), and Fiber-Switch Capable (FSC). Each network element (NE) running a GMPLS control plane collects network information from other NEs and supports service provisioning through signaling in a distributed manner. More generic description for Traffic-engineering networking information exchange can be found in [RFC7926].

On the other hand, Software-Defined Networking (SDN) technologies have been introduced to control the transport network in a centralized manner. Central controllers can collect network information from each node and provision services to corresponding nodes. One of the examples is the Abstraction and Control of Traffic Engineered Networks (ACTN) [RFC8453], which defines a hierarchical architecture with PNC, MDSC and CNC as central controllers for different network abstraction levels. A PCE-based approach has been proposed as Application-Based Network Operations (ABNO) in [RFC7491].

In such centralized controller architectures, GMPLS can be applied for the NE-level control. A central controller may support GMPLS enabled domains and may interact with a GMPLS enabled domain where the GMPLS control plane does the service provisioning from ingress to egress. In this case the centralized controller sends the request to the ingress node and does not have to configure all NEs along the

path through the domain from ingress to egress thus leveraging the GMPLS control plane. This document describes how GMPLS control interworks with centralized controller system in transport network.

2. Overview

In this section, overviews of GMPLS control plane and centralized controller system are discussed as well as the interactions between the GMPLS control plane and centralized controllers.

2.1. Overview of GMPLS Control Plane

GMPLS separates the control plane and the data plane to support time-division, wavelength, and spatial switching, which are significant in transport networks. For the NE level control in GMPLS, each node runs a GMPLS control plane instance. Functionalities such as service provisioning, protection, and restoration can be performed via GMPLS communication among multiple NEs. At the same time, the controller can also collect node and link resources in the network to construct the network topology and compute routing paths for serving service requests.

Several protocols have been designed for GMPLS control [RFC3945] including link management [RFC4204], signaling [RFC3471], and routing [RFC4202] protocols. The controllers applying these protocols communicate with each other to exchange resource information and establish LSP. In this way, controllers in different nodes in the network have the same network topology and provision services based on local policies.

2.2. Overview of Centralized Controller System

With the development of SDN technologies, a centralized controller architecture has been introduced to transport networks such as ACTN [RFC8453]. In centralized controller system, a controller is aware of the network topology and is responsible for provisioning incoming service requests. In ACTN, multiple abstraction levels are designed and controllers at different levels implement different functions. This kind of abstraction enables multi-vendor, multi-domain, and multi-technology control.

For example in ACTN, an MDSC coordinates several PNCs controlling different domains. Each PNC provides a topological view of the domain it controls, which can be abstracted, to the MDSC, so that the MDSC learns the topology of the network encompassing multiple domains. When a multi-domain service request arrives at the MDSC, the MDSC first computes an end-to-end path based on the abstracted topology view provided by the PNCs. Then, the MDSC splits this path to multiple segment according to domain boundaries and allocate each

segment to corresponding PNC for detailed path computation and LSP segment setup. When each PNC has reported the establishment of its LSP segment, the multi-domain service is established.

2.3. GMPLS Control Interwork with Centralized Controller System

The ACTN framework [RFC8453] defines a hierarchical controller architecture and describes how these controllers communicate with each other in order to control a multi-domain transport network. The controllers at the different levels in the hierarchy typically perform network abstraction of the domain they control and provide an abstracted view of their domain to the controller at the next level in the hierarchy. The controllers at the different hierarchical levels also interact with each other during end-to-end service establishment, which can span multiple domains. Within each domain, GMPLS control can be applied to each NE. The bottom-level central controller like PNC can act as a NE to collect network information and initiate LSP. Following figure shows an example of GMPLS interworking with ACTN.

```
                        +----------+
                        |   MDSC   |
                        +----------+
                          ^      ^
                          |      |
                +---------+      +---------+
                |  RESTConf / YANG models  |
                V                          V
           +---------+                +---------+
           |   PNC   |                |   PNC   |
           +---------+                +---------+
              ^   ^                      ^   ^
              |   |                      |   |
       OSPF-TE|   |PCEP           OSPF-TE|   |PCEP
       Netconf|   |               Netconf|   |
              V   V                      V   V
         .-------------.   Inter-   .-------------.
        /               \  domain  /               \
       |       LMP       |  link  |       LMP       |
      |      OSPF-TE     ==========     OSPF-TE      |
       |     RSVP-TE     |        |     RSVP-TE     |
        \               /          \               /
          `------------`             `------------`
           GMPLS domain               GMPLS domain
```

Figure 1: Example of GMPLS interworks with ACTN

In Figure 1, each domain has the GMPLS control plane enabled at the physical network level. The PNC can listen to the IGP routing protocol messages (OSPF LSAs for example) that the GMPLS control plane instances are disseminating into the network and thus learn the network topology. For path computation in the domain with PNC implementing a PCE, PCCs (e.g. NEs, other controller/PCE) use PCEP to ask the PNC for a path and get replies. The MDSC communicates with PNCs using for example REST/RESTConf based on YANG data models. As a PNC has learned its domain topology, it can report the topology to the MDSC. When a service arrives, the MDSC computes the path and coordinates PNCs to establish the corresponding LSP segment.

Alternatively, the NETCONF protocol can be used to retrieve topology information utilizing the [TE-TOPO] Yang model and the technology-specific YANG model augmentations required for the specific network technology. The PNC can retrieve topology information from any NE (the GMPLS control plane instance of each NE in the domain has the same topological view), construct the topology of the domain and export an abstracted view to the MDSC. Based on the topology retrieved from multiple PNCs, the MDSC can create topology graph of the multi-domain network, and can use it for path computation. To setup a service, the MDSC can exploit Yang tunnel model together with the technology-specific YANG model augmentations.

3. Link Management Protocol

Link management protocol (LMP) [RFC4204] runs between a pair of nodes and is used to manage TE links. In addition to setup and maintain control channels, LMP can be used to verify the data link connectivity and correlate the link property. In this way, link resources, which are fundamental resources in the network, are discovered by both ends of the link.

4. Routing Options

In GMPLS control, link state information is flooded within the network as defined in [RFC4202]. Each node in the network can build the network topology according to the flooded link state information. Routing protocols such as OSPF-TE [RFC4203] and ISIS-TE [RFC5307] have been extended to support different interfaces in GMPLS.

In centralized controller system, central controller can be placed at the GMPLS network and passively receive the information flooded in the network. In this way, the central controller can construct and update the network topology.

4.1. OSPF-TE

OSPF-TE is introduced for TE networks in [RFC3630]. OSPF extensions have been defined in [RFC4203] to enable the capability of link state information for GMPLS network. Based on this work, OSPF protocol has been extended to support technology-specific routing. The routing protocol for OTN, WSON and optical flexi-grid network are defined in [RFC7138], [RFC7688] and [RFC8363], respectively.

4.2. ISIS-TE

ISIS-TE is introduced for TE networks in [RFC5305] and is extended to support GMPLS routing functions [RFC5307], and has been updated to [RFC7074] to support the latest GMPLS switching capability and Types fields.

4.3. Netconf/RESTconf

Netconf [RFC6241] and RESTconf [RFC8040] protocols are originally used for network configuration. Besides, these protocols can also be used for topology retrieval by using topology-related YANG models, such as [RFC8345] and [TE-topo]. These protocols provide a powerful mechanism for notification that permits to notify the client about topology changes.

5. Path Computation

Once a controller learn the network topology, it can utilize the available resources to serve service requests by performing path computation. Due to abstraction, the MDSC may not have sufficient information to compute the optimal path. In this case, the MDSC can interact with different domain controllers by sending Yang Path Computation requests [PAT-COMP] to compute a set of potential optimal paths and then, based on its own constraints, policy and specific knowledge (e.g. cost of access link) can choose the more feasible path for service e2e path setup.

Path computation is one of the key objectives in various types of controllers. In the given architecture, it is possible for different components that have the capability to compute the path.

5.1. Constraint-based Path Computing in GMPLS Control

In GMPLS control, a routing path is computed by the ingress node [RFC3473] and is based on the ingress node TED. Constraint-based path computation is performed according to the local policy of the ingress node.

5.2. Path Computation Element (PCE)

PCE has been introduced in [RFC4655] as a functional component that provides services to compute path in a network. In [RFC5440], the path computation is accomplished by using the Traffic Engineering Database (TED), which maintains the link resources in the network. The emergence of PCE efficiently improve the quality of network planning and offline computation, but there is a risk that the computed path may be infeasible if there is a diversity requirement, because stateless PCE has no knowledge about the former computed paths.

To address this issue, stateful PCE has been proposed in [RFC8231]. Besides the TED, an additional LSP Database (LSP-DB) is introduced to archive each LSP computed by the PCE. In this way, PCE can easily figure out the relationship between the computing path and former computed paths. In this approach, PCE provides computed paths to PCC, and then PCC decides which path is deployed and when to be established.

In PCE Initiation [I-D.ietf-pce-pce-initiated-lsp], PCE is allowed to trigger the PCC to setup, maintenance, and teardown of the PCE-initiated LSP under the stateful PCE model. This would allow a dynamic network that is centrally controlled and deployed.

In centralized controller system, the PCE can be implement in a central controller, and the central controller performs path computation according to its local policies. On the other hand, the PCE can also be placed outside of the central controller. In this case, the central controller acts as a PCC to request path computation to the PCE through PCEP.

6. Signaling Options

Signaling mechanism is used to setup LSPs in GMPLS control. Messages are sent hop by hop between the ingress node and the egress node of the LSP to allocate labels. Once the labels are allocated along the path, the LSP setup is accomplished. Signaling protocols such as RSVP-TE [RFC3473] and CR-LDP [RFC3472] have been extended to support different interfaces in GMPLS.

6.1. RSVP-TE

RSVP-TE is introduced in [RFC3209] and extended to support GMPLS signaling in [RFC3473]. Several label formats are defined for a generalized label request, a generalized label, suggested label and label sets. Based on [RFC3473], RSVP-TE has been extended to support technology-specific signaling. The RSVP-TE extensions for OTN, WSON,

optical flexi-grid network are defined in [RFC7139], [RFC7689], and [RFC7792], respectively.

7. Interworking Scenarios

7.1. Topology Collection & Synchronization

Topology information is necessary on both network elements and controllers. The topology on network element is usually raw information, while the topology on the controller can be either raw or abstracted. Three different abstraction method has been described in [RFC8453], and different controllers can select the corresponding method depending on application.

When there are changes in the network topology, the impacted network element(s) need to report changes to all the other network elements, together with the controller, to sync up the topology information. The inter-NE synchronization can be achieved via protocols mentioned in section 3 and 4. The topology synchronization between NEs and controllers can either be achieved by routing protocols OSPF-TE/PCEP-LS in [PCEP-LS] or Netconf protocol with YANG model.

7.2. Multi-domain/layer Service Provisioning

Based on the topology information on controllers and network elements, service provisioning can be deployed. Plenty of methods have been specified for single domain service provisioning, such as using PCEP and RSVP-TE.

Multi-domain/layer service provisioning would request coordination among the controller hierarchies. Given the service request, the end-to-end delivery procedure may include interactions on MPI and SBI. The computation for a cross-domain/layer path is usually completed by MDSC, who has a global view of the topologies. Then the configuration is decomposed into lower layer controllers, including both MDSC and PNCs, to configure the network elements to set up the path.

A combination of the centralized and distributed protocols may be necessary for the interaction between network elements and controller. A typical example would be the PCE Initiation scenario, in which a PCE message (PCInitiate) is sent from the controller to the first-end node, and then trigger a RSVP procedure along the path. Similarly, the interaction between the controller and the ingress node of a domain can be achieved by Netconf protocol with corresponding YANG models, and then completed by running RSVP among the network elements.

7.3. Recovery

The GMPLS recovery functions are described in [RFC4426]. Two models, span protection and end-to-end protection and restoration, are discussed with different protection schemes and message exchange requirements. Related RSVP-TE extensions to support end-to-end recovery is described in [RFC4872]. The extensions in [RFC4872] include protection, restoration, preemption, and rerouting mechanisms for an end-to-end LSP. Besides end-to-end recovery, a GMPLS segment recovery mechanism is defined in [RFC4873]. By introducing secondary record route objects, LSP segment can be switched to another path like fast reroute [RFC4090].

For the recovery with controllers, timely interaction between controller and network elements are required. Usually the re-routing can be decomposed into path computation and delivery, the controller can take some advantage in the path computation due to the global topology view. And the delivery can be achieved by the procedure described in section 7.2.

7.4. Controller Reliability

Given the important role in the network, the reliability of controller is critical. Once a controller is shut down, the network should operate as well. It can be either achieved by controller back up or functionality back up. There are several of controller backup or federation mechanisms in the literature. It is also more reliable to have some function back up in the network element, to guarantee the performance in the network.

8. Network Management

TBD.

9. Security Considerations

TBD.

10. IANA Considerations

This document requires no IANA actions.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3471] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.

[RFC3472] Ashwood-Smith, P., Ed. and L. Berger, Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions", RFC 3472, January 2003.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.

[RFC4202] Kompella, K., Ed. and Y. Rekhter, Ed., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, October 2005.

[RFC4204] Lang, J., Ed., "Link Management Protocol (LMP)", RFC 4204, October 2005.

[RFC4426] Lang, J., Ed., Rajagopalan, B., Ed., and D. Papadimitriou, Ed., "Generalized Multi-Protocol Label witching (GMPLS) Recovery Functional Specification", RFC 4426, March 2006.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[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, May 2007.

[RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, May 2007.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, October 2008.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder J., Bierman A., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC7074] Berger, L. and J. Meuric, "Revised Definition of the GMPLS Switching Capability and Type Fields", RFC 7074, November 2013.

[RFC7138] Ceccarelli, D., Ed., Zhang, F., Belotti, S., Rao, R., and J. Drake, "Traffic Engineering Extensions to OSPF for GMPLS Control of Evolving G.709 Optical Transport Networks", RFC 7138, March 2014.

[RFC7139] Zhang, F., Ed., Zhang, G., Belotti, S., Ceccarelli, D., and K. Pithewan, "GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks", RFC 7139, March 2014.

[RFC7491] King, D., Farrel, A., "A PCE-Based Architecture for Application-Based Network Operations", RFC7491, March 2015.

[RFC7688] Lee, Y., Ed. and G. Bernstein, Ed., "GMPLS OSPF Enhancement for Signal and Network Element Compatibility for Wavelength Switched Optical Networks", RFC 7688, November 2015.

[RFC7689] Bernstein, G., Ed., Xu, S., Lee, Y., Ed., Martinelli, G., and H. Harai, "Signaling Extensions for Wavelength Switched Optical Networks", RFC 7689, November 2015.

[RFC7792] Zhang, F., Zhang, X., Farrel, A., Gonzalez de Dios, O., and D. Ceccarelli, "RSVP-TE Signaling Extensions in Support of Flexi-Grid Dense Wavelength Division Multiplexing (DWDM) Networks", RFC 7792, March 2016.

[RFC7926] Farrel, A., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D. and Zhang, X., "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC7926, July 2016.

[RFC8040] Bierman, A., Bjorklund, M., Watsen, K., "RESTCONF Protocol", RFC 8040, January 2017.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, September 2017.

[RFC8363] Zhang, X., Zheng, H., Casellas, R., Dios, O., and D. Ceccarelli, "GMPLS OSPF-TE Extensions in support of Flexi-grid DWDM networks", RFC8363, February 2017.

[RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", RFC 8281, October 2017.

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., Liu, X., "A YANG Data Model for Network Topologies", RFC 8345, March 2018.

[RFC8453] Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", RFC 8453, August 2018.

[I-D. dhodylee-pce-pcep-ls] Dhody, D., Lee, Y., Ceccarelli, D., "PCEP Extensions for Distribution of Link-State and TE Information", draft-dhodylee-pce-pcep-ls, work in progress.

[TE-topo] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., Gonzalez De Dios, O., "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-15, work in progress.

[PAT-COMP] Busi, I., Belotti, S., Lopez, V., Gonzalez de Dios, O., Sharma, A., Shi, Y., Vilalta, R., Setheraman, K., "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation-01, work in progress.

11.2. Informative References

12. Authors' Addresses

Haomian Zheng
Huawei Technologies
F3 R&D Center, Huawei Industrial Base,
Bantian, Longgang District,
Shenzhen 518129 P.R.China
Email: zhenghaomian@huawei.com

Xianlong Luo
Huawei Technologies
F3 R&D Center, Huawei Industrial Base,
Bantian, Longgang District,
Shenzhen 518129 P.R.China
Email: luoxianlong@huawei.com

Yunbin Xu
CAICT
Email: xuyunbin@ritt.cn

Yang Zhao
China Mobile
Email: zhaoyangyjy@chinamobile.com

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

Dieter Beller
Nokia
Email: Dieter.Beller@nokia.com