

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: March 27, 2020

A. Aguado
Nokia
O. Gonzalez de Dios, Ed.
V. Lopez
Telefonica
D. Voyer
Bell Canada
L. Munoz
Vodafone
September 24, 2019

Layer 3 VPN Network Model
draft-aguado-opsawg-l3sm-l3nm-02

Abstract

RFC8299 defines a L3VPN Service YANG data Model (L3SM) that can be used for communication between customers and network operators. Such model is adequate for the customer to network operator conversation and plays the role of a Customer Service Model, according to the terminology defined in RFC8309.

There is a need for a YANG model to be used in the communication between the entity that interacts directly with the customer, the service orchestrator, (either fully automated or a human operator) and the entity in charge of network orchestration and control (aka network controller / orchestrator).

This document proposes a L3VPN Network Yang Model (L3NM) to facilitate communication between a service orchestrator and a network controller / orchestrator. The resulting model is called the L3VPN Network Model (L3NM) and provides a network-centric view of the L3VPN services.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 27, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. INTRODUCTION	3
1.1. TERMINOLOGY	3
1.2. Requirements Language	3
2. REFERENCE ARCHITECTURE	4
3. YANG MODEL EXPLANATION	6
3.1. STRUCTURE OF THE MODEL	7
3.2. SITE AND BEARERS	7
3.3. BEARER AND ETHERNET ENCAPSULATION	7
3.4. MULTI-DOMAIN RESOURCE MANAGEMENT	7
3.5. REMOTE FAR-END CONFIGURATION	8
3.6. PROVIDE EDGE IDENTIFICATION POINT	8
4. DESING OF THE DATA MODEL	9
5. YANG MODULE	20
6. IANA CONSIDERATIONS	93
7. SECURITY CONSIDERATIONS	93
8. IMPLEMENTATION STATUS	93
9. ACKNOWLEDGEMENTS	94
10. CONTRIBUTORS	94
11. References	94
11.1. NORMATIVE REFERENCES	94
11.2. INFORMATIVE REFERENCES	94
Authors' Addresses	95

1. INTRODUCTION

[RFC8299] defines a L3VPN Service YANG data Model (L3SM) model that can be used for communication between customers and network operators. Such model is focused on describing the customer view of the services, and provides an abstracted view of the customer's requested services. That approach limits the usage of the L3SM to the role of a Customer Service Model, according to the terminology defined in [RFC8309].

The YANG data model proposed in this document is called the L3VPN Network Model (L3NM). The L3NM model is aimed at providing a network-centric view of L3 VPN Services. The model can be used to facilitate communication between the service orchestrator, and the network controller / orchestrator. It enables further capabilities, such as resource management or to serve as a multi-domain orchestration interface, where transport resources must be synchronized. The YANG module has been built with a prune and extend approach, taking as a starting points the YANG model described in [RFC8299].

Hence, this document does not obsolete, but complements, the definitions in [RFC8299]. It aims to provide a different scope for the L3SM, but does not attempt to address all deployment cases especially those where the L3VPN connectivity is supported through the coordination of different VPNs in different underlying networks. More complex deployment scenarios involving the coordination of different VPN instances and different technologies to provide end-to-end VPN connectivity are addressed by a complementary YANG model defined in [I-D.evenwu-opsawg-yang-composed-vpn].

1.1. TERMINOLOGY

This document assumes that the reader is familiar with the contents of [RFC6241], [RFC7950], [RFC8299], [RFC8309], and [RFC8453] and uses terminology from those documents. Tree diagrams used in this document follow the notation defined in [RFC8340].

1.2. 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. REFERENCE ARCHITECTURE

Figure 1 shows where the L3NM is used in a management stack. The figure is an expansion of the architecture presented in Section 5 of [RFC8299] and decomposes the box marked "orchestration" in that figure into three separate functional components called "Service Orchestration", "Network Orchestration", and "Domain Orchestration".

Note that some implementations may choose to construct a monolithic orchestration component, but this document assumes that there are many benefits for flexibility of implementation and deployment to separate the functional components, and that separation demands the existence of separate YANG models to be used between the components.

At the same time, terminology from [RFC8309] is introduced to show the distinction between the "Customer Service Model", the "Service Delivery Model", the "Network Configuration Model", and the "Device Configuration Model". In that context, the "Domain Orchestration" and "Config Manager" roles may be performed by "Controllers".

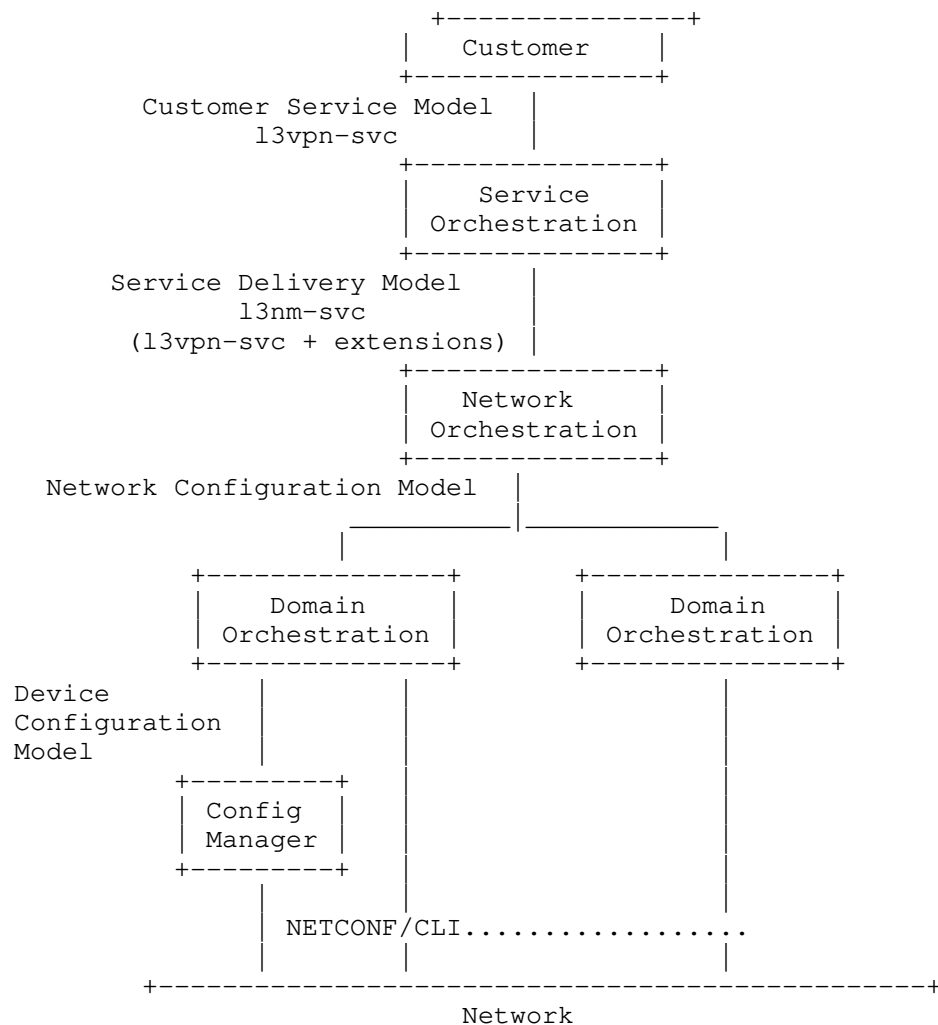


Figure 1: L3SM and L3NM

The L3SM and L3NM may also be set in the context of the ACTN architecture [RFC8453]. Figure 2 shows the Customer Network Controller (CNC), the Multi-Domain Service Coordinator (MDSC), and the Provisioning Network Controller (PNC). It also shows the interfaces between these functional units: the CNC-MDSC Interface (CMI), the MDSC-PNC Interface (MPI), and the Southbound Interface (SBI).

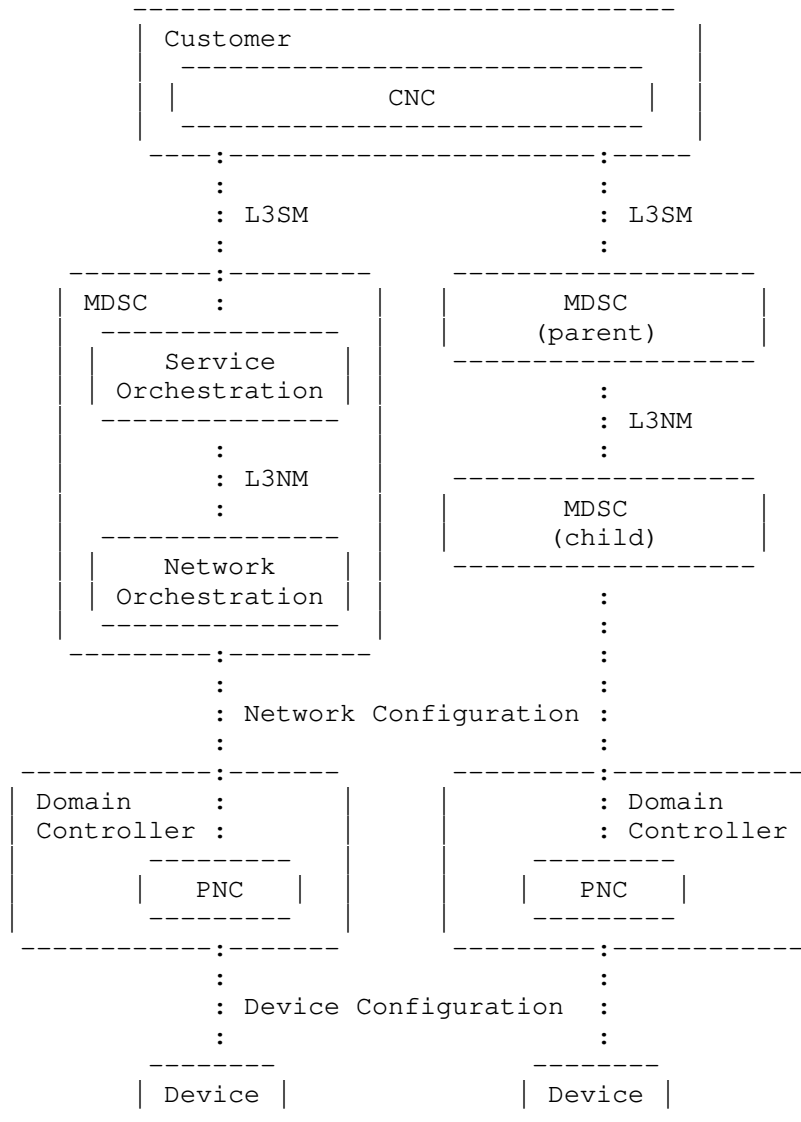


Figure 2: L3SM and L3NM in the Context of ACTN

3. YANG MODEL EXPLANATION

The scenarios covered in the L3NM model includes: the integration of Ethernet and encapsulation parameters, the extension for transport resources (e.g., Route targets and Route distinguishers) to be orchestrated from the management system, far-end configuration of PEs

not managed by the management system and the definition for PE identification.

3.1. STRUCTURE OF THE MODEL

The YANG module is divided into three main containers: "vpn-services", "sites" and "vpn-profiles".

3.2. SITE AND BEARERS

A site, as per [RFC8299], represents a connection of a customer office to one or more VPN services. As this YANG module, is the network view, each site is associated with a list of bearers. A bearer is the layer two connections with the site. In the module it is assumed that the bearer has been allocated by the Service Provider (e.g., by the service orchestrator). The bearer is associated to a network element and a port. Hence, a bearer is not just a bearer-reference, but also a true reference to a given port in the service provider network.

3.3. BEARER AND ETHERNET ENCAPSULATION

The definition of a L3VPN is commonly specified not only at the IP layer, but also requires to identify parameters at the Ethernet layer, such as encapsulation type (e.g., VLAN, QinQ, QinAny, VxLAN, etc.). This specification is not supported in [RFC8299], whilst it suggests that any extension on this direction shall be implemented via augmentation of the bearer container. The extension defined to cope with these parameters uses the connection container inside the site-network-access defined by the [RFC8466]. This container defines protocol parameters to enable connectivity at Layer 2. In the context of L3SM, the augmentation includes only mandatory parameters for the service configuration, which are mainly related to the interface encapsulation. Other definitions from L2SM connection container are left aside. For example, Link Aggregation (LAG) information is not required and it shall be configured prior to the service configuration, being the aggregated interface identified in the model as the bearer-reference, as discussed later in Section 3.4.

3.4. MULTI-DOMAIN RESOURCE MANAGEMENT

The implementation of L3VPN services which span across administratively separated domains (i.e., that are under the administration of different management systems or controllers) requires some network resources to be synchronized between systems. Particularly, there are two resources that must be orchestrated and manage to avoid asymmetric (non-functional) configuration, or the usage of unavailable resources. For example, RTs shall be

synchronized between PEs. When every PE is controlled by the same management system, RT allocation can be performed by the system. In cases where the service spans across multiple management systems, this task of allocating RTs has to be aligned across the domains, therefore, the service model must provide a way to specify RTs. In addition, RDs must also be synchronized to avoid collisions in RD allocation between separate systems. An incorrect allocation might lead to the same RD and IP prefixes being exported by different PE routers.

3.5. REMOTE FAR-END CONFIGURATION

Depending on the control plane implementation, different network scenarios might require additional information for the L3VPN service to be configured and active. For example, an L3VPN Option C service, if no reflection of IPv4 VPN routes is configured via ASBR or route reflector, may require additional configuration (e.g. a new BGP neighbor) to be coordinated between both management systems. This definition requires for every management system participant in the VPN to receive not just their own sites and site-network-accesses, but also to receive information about external ones, identified as an external site-network-access-type. In addition, this particular site-network-access is augmented to include the loopback address of the far-end (remote/external) PE router.

3.6. PROVIDE EDGE IDENTIFICATION POINT

[RFC8299] states that the "bearer-reference" parameter is used in cases where the customer has already ordered a network connection to the service provider (SP) apart from the IP VPN site and wants to reuse this connection. The string used is an internal reference from the SP and describes the already-available connection. Usually, a client interface (either a customer one or an interface used by the SP) is already in place and connected, although it has not being use previously. In some other cases (e.g., for stitching purposes), the termination of a VPN service is done over logical terminations within a PE router.

The bearer-reference must serve as a strict unequivocal parameters to identify the connection between a PE and a client (CE). This means that, despite the type is maintained as a string and there is no restriction in the way this data is formed, the bearer-reference must serve as the unique way to identify the PE router and the client interface. This, together with the encapsulation augments proposed in Section 3.2, serves as the way to identify the client interface and configure L2 specific parameters.

4. DESIGN OF THE DATA MODEL

The augmentations defined in this document are organised per scenario, as defined in Section 3. The case described Section 3.4 does not need any further extension of the data model and only requires a more restricted definition on how the data model is used for PE router and client port identification, so no augmentation is implemented for this scenario.

The augmentations implemented are distributed as follows:

- o An extension including RT and RD definition for the L3VPN, following the YANG definitions from BESS-L3VPN. This extension was developed creating a container "ie-profiles" under the VPN Service. All the import-export information can be created and reused for several VPN-Nodes.
 - * If the "ie-profile" is empty the domain controller should automatically assign RD and RTs. This is not valid for a multi-domain scenario
- o The second augmentation copes with the information from a remote PE not directly under management system supervision. This augmentation does not follow any previously defined model and includes the loopback IP address of the external router.
- o The third augmentation copes with a pseudowire termination under a VPN service. This termination requires the management of the Virtual Circuit Identifier under the VPN service.
- o Access-group-id has been added within the site network access in order to allow associations between interfaces that have similar behaviors. For example, identify two interfaces in dual homing distribution.
- o The last augmentation includes information below layer 3 that is required for the service. In particular, we include information related to clients interface encapsulation and aggregation.

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

```
|----- EXAMPLE -----|  
  
module: ietf-l3vpn-ntw  
  +--rw l3vpn-ntw  
    +--rw vpn-profiles  
      | +--rw valid-provider-identifiers
```

```

    +--rw cloud-identifier* [id] {cloud-access}?
    |   +--rw id      string
    +--rw encryption-profile-identifier* [id]
    |   +--rw id      string
    +--rw qos-profile-identifier* [id]
    |   +--rw id      string
    +--rw bfd-profile-identifier* [id]
    |   +--rw id      string
    +--rw routing-profile-identifier* [id]
    |   +--rw id      string
+--rw vpn-services
+--rw vpn-service* [vpn-id]
+--rw vpn-id          svc-id
+--rw customer-name?  string
+--rw vpn-service-topology? identityref
+--rw description?    string
+--rw ie-profiles
+--rw ie-profile* [ie-profile-id]
+--rw ie-profile-id  string
+--rw rd?            rt-types:route-distinguisher
+--rw vpn-targets
+--rw vpn-target* [route-target]
+--rw route-target    rt-types:route-target
t
+--rw route-target-type rt-types:route-target
t-type
+--rw vpn-nodes
+--rw vpn-node* [vpn-node-id ne-id]
+--rw vpn-node-id  string
+--rw description? string
+--rw ne-id        string
+--rw router-id?   inet:ip-address
+--rw address-family? address-family
+--rw node-role?   identityref
+--rw rd?          rt-types:route-distinguisher
+--rw vpn-targets
+--rw vpn-target* [route-target]
+--rw route-target    rt-types:route-target
t
+--rw route-target-type rt-types:route-target
t-type
+--rw status
+--rw admin-enabled?  boolean
+--ro oper-status?    operational-type
+--rw maximum-routes
+--rw address-family* [af]
+--rw af              address-family
+--rw maximum-routes? uint32
+--rw node-ie-profile? -> /l3vpn-ntw/vpn-services/v
pn-service/ie-profiles/ie-profile/ie-profile-id
+--rw cloud-accesses {cloud-access}?
+--rw cloud-access* [cloud-identifier]
+--rw cloud-identifier -> /l3vpn-ntw/vpn-profil
es/valid-provider-identifiers/cloud-identifier/id

```

```

|         |         |         | +--rw (list-flavor)?
|         |         |         |   +---:(permit-any)
|         |         |         |     | ++-rw permit-any?           empty
|         |         |         |   +---:(deny-any-except)
|         |         |         |     | ++-rw permit-site*       -> /l3vpn-ntw/sites/site
/site-id
|         |         |         |   +---:(permit-any-except)
|         |         |         |     ++-rw deny-site*          -> /l3vpn-ntw/sites/site
/site-id
|         |         |         | +--rw address-translation
|         |         |         |   +--rw nat44
|         |         |         |     ++-rw enabled?                boolean
|         |         |         |     ++-rw nat44-customer-address?    inet:ipv4-addre
ss
|         |         |         | +--rw multicast {multicast}?
|         |         |         |   +--rw enabled?                    boolean
|         |         |         |   +--rw customer-tree-flavors
|         |         |         |     | ++-rw tree-flavor*            identityref
|         |         |         |   +--rw rp
|         |         |         |     +--rw rp-group-mappings
|         |         |         |       +--rw rp-group-mapping* [id]
|         |         |         |         +--rw id                      uint16
|         |         |         |         +--rw provider-managed
|         |         |         |           +--rw enabled?              boolean
|         |         |         |           +--rw rp-redundancy?        boolean
|         |         |         |           +--rw optimal-traffic-delivery? boolean
|         |         |         |       +--rw rp-address                inet:ip-address
|         |         |         |       +--rw groups
|         |         |         |         +--rw group* [id]
|         |         |         |           +--rw id                      uint16
|         |         |         |           +--rw (group-format)
|         |         |         |             +---:(singleaddress)
|         |         |         |               | ++-rw group-address?    inet:ip-addr
ess
|         |         |         |               +---:(startend)
|         |         |         |                 +--rw group-start?      inet:ip-addr
ess
|         |         |         |                 +--rw group-end?        inet:ip-addr
ess
|         |         |         | +--rw rp-discovery
|         |         |         |   +--rw rp-discovery-type?          identityref
|         |         |         |   +--rw bsr-candidates
|         |         |         |     +--rw bsr-candidate-address*      inet:ip-address
+--rw carrierscarrier?          boolean {carrierscarrier}?
+--rw extranet-vpns {extranet-vpn}?
+--rw extranet-vpn* [vpn-id]
+--rw vpn-id                     svc-id
+--rw local-sites-role?          identityref
+--rw sites
+--rw site* [site-id]
+--rw site-id                     svc-id
+--rw description?                string
+--rw requested-site-start?        yang:date-and-time
+--rw requested-site-stop?         yang:date-and-time

```

```

+--rw locations
|   +--rw location* [location-id]
|       +--rw location-id      svc-id
|       +--rw address?         string
|       +--rw postal-code?     string
|       +--rw state?           string
|       +--rw city?            string
|       +--rw country-code?    string
+--rw devices
|   +--rw device* [device-id]
|       +--rw device-id        svc-id
|       +--rw location         -> ../../../../locations/location/lo
cation-id
|           +--rw management
|               +--rw address-family? address-family
|               +--rw address         inet:ip-address
+--rw site-diversity {site-diversity}?
|   +--rw groups
|       +--rw group* [group-id]
|           +--rw group-id      string
+--rw management
|   +--rw type      identityref
+--rw site-vpn-flavor?      identityref
+--rw maximum-routes
|   +--rw address-family* [af]
|       +--rw af          address-family
|       +--rw maximum-routes? uint32
+--rw security
|   +--rw authentication
|   +--rw encryption {encryption}?
|       +--rw enabled?    boolean
|       +--rw layer?      enumeration
|   +--rw encryption-profile
|       +--rw (profile)?
|           +--:(provider-profile)
|           |   +--rw profile-name?    -> /l3vpn-ntw/vpn-profil
es/valid-provider-identifiers/encryption-profile-identifier/id
|           +--:(customer-profile)
|           |   +--rw algorithm?        string
|           +--rw (key-type)?
|               +--:(psk)
|               +--rw preshared-key?    string
+--rw service
|   +--rw qos {qos}?
|       +--rw qos-classification-policy
|           +--rw rule* [id]
|               +--rw id                  string
|               +--rw (match-type)?
|                   +--:(match-flow)
|                   +--rw match-flow

```

						+-rw dscp?	inet:dscp
						+-rw dot1p?	uint8
						+-rw ipv4-src-prefix?	inet:ipv4-p
refix							
						+-rw ipv6-src-prefix?	inet:ipv6-p
refix							
						+-rw ipv4-dst-prefix?	inet:ipv4-p
refix							
						+-rw ipv6-dst-prefix?	inet:ipv6-p
refix							
						+-rw l4-src-port?	inet:port-n
umber							
						+-rw target-sites*	svc-id {tar
get-sites}?							
						+-rw l4-src-port-range	
						+-rw lower-port?	inet:port-numbe
r							
						+-rw upper-port?	inet:port-numbe
r							
						+-rw l4-dst-port?	inet:port-n
umber							
						+-rw l4-dst-port-range	
						+-rw lower-port?	inet:port-numbe
r							
						+-rw upper-port?	inet:port-numbe
r							
						+-rw protocol-field?	union
						+-: (match-application)	
						+-rw match-application?	identityref
						+-rw target-class-id?	string
						+-rw qos-profile	
						+-rw (qos-profile)?	
						+-: (standard)	
						+-rw profile?	-> /l3vpn-ntw/vpn-profile
s/valid-provider-identifiers/qos-profile-identifier/id						+-rw direction?	identityref
						+-: (custom)	
						+-rw classes {qos-custom}?	
						+-rw class* [class-id]	
						+-rw class-id	string
						+-rw direction?	identityref
						+-rw rate-limit?	decimal64
						+-rw latency	
						+-rw (flavor)?	
						+-: (lowest)	
						+-rw use-lowest-latency?	e
mpty							
						+-: (boundary)	
						+-rw latency-boundary?	u
int16							
						+-rw jitter	
						+-rw (flavor)?	
						+-: (lowest)	
						+-rw use-lowest-jitter?	em
pty							
						+-: (boundary)	
						+-rw latency-boundary?	ui
nt32							
						+-rw bandwidth	
						+-rw guaranteed-bw-percent	deci
mal64							
						+-rw end-to-end?	empty
y							

```
|  +--rw carrierscarrier {carrierscarrier}?  
|  |  +--rw signalling-type?  enumeration  
|  +--rw multicast {multicast}?
```

```

    +--rw multicast-site-type?          enumeration
    +--rw multicast-address-family
    |   +--rw ipv4?    boolean {ipv4}?
    |   +--rw ipv6?    boolean {ipv6}?
    +--rw protocol-type?                enumeration
+--rw traffic-protection {fast-reroute}?
|   +--rw enabled?    boolean
+--rw routing-protocols
|   +--rw routing-protocol* [type]
|   |   +--rw type          identityref
|   |   +--rw routing-profiles* [id]
|   |   |   +--rw id        -> /l3vpn-ntw/vpn-profiles/valid-pro
vider-identifiers/routing-profile-identifier/id
|   |   |   +--rw type?    ie-type
|   |   +--rw ospf {rtg-ospf}?
|   |   |   +--rw address-family*    address-family
|   |   |   +--rw area-address        yang:dotted-quad
|   |   |   +--rw metric?             uint16
|   |   |   +--rw mtu?                uint16
|   |   |   +--rw security
|   |   |   |   +--rw auth-key?    string
|   |   |   +--rw sham-links {rtg-ospf-sham-link}?
|   |   |   |   +--rw sham-link* [target-site]
|   |   |   |   |   +--rw target-site    svc-id
|   |   |   |   |   +--rw metric?      uint16
|   |   +--rw bgp {rtg-bgp}?
|   |   |   +--rw autonomous-system    uint32
|   |   |   +--rw address-family*      address-family
|   |   |   +--rw neighbor?            inet:ip-address
|   |   |   +--rw multihop?            uint8
|   |   |   +--rw security
|   |   |   |   +--rw auth-key?    string
|   |   +--rw static
|   |   |   +--rw cascaded-lan-prefixes
|   |   |   |   +--rw ipv4-lan-prefixes* [lan next-hop] {ipv4}?
|   |   |   |   |   +--rw lan          inet:ipv4-prefix
|   |   |   |   |   +--rw lan-tag?    string
|   |   |   |   |   +--rw next-hop    inet:ipv4-address
|   |   |   |   +--rw ipv6-lan-prefixes* [lan next-hop] {ipv6}?
|   |   |   |   |   +--rw lan          inet:ipv6-prefix
|   |   |   |   |   +--rw lan-tag?    string
|   |   |   |   |   +--rw next-hop    inet:ipv6-address
|   |   +--rw rip {rtg-rip}?
|   |   |   +--rw address-family*      address-family
|   |   +--rw vrrp {rtg-vrrp}?
|   |   |   +--rw address-family*      address-family
+--ro actual-site-start?                yang:date-and-time
+--ro actual-site-stop?                 yang:date-and-time
+--rw site-bearers

```

```

    +--rw bearer* [bearer-id]
    |   +--rw bearer-id      string
    |   +--rw BearerType?   identityref
    |   +--rw ne-id?        string
    |   +--rw port-id?      string
    |   +--rw lag-id?       string
+--rw site-network-accesses
  +--rw site-network-access* [site-network-access-id]
  |   +--rw site-network-access-id      svc-id
  |   +--rw description?                 string
  |   +--rw status
  |   |   +--rw admin-enabled?   boolean
  |   |   +--ro oper-status?     operational-type
  |   +--rw site-network-access-type?   identityref
  +--rw (location-flavor)
  |   +--:(location)
  |   |   +--rw location-reference?  -> ../../../../locatio
ns/location/location-id
  |   +--:(device)
  |   |   +--rw device-reference?    -> ../../../../devices
/device/device-id
  +--rw access-diversity {site-diversity}?
  |   +--rw groups
  |   |   +--rw group* [group-id]
  |   |   |   +--rw group-id      string
  |   +--rw constraints
  |   |   +--rw constraint* [constraint-type]
  |   |   |   +--rw constraint-type  identityref
  |   |   |   +--rw target
  |   |   |   |   +--rw (target-flavor)?
  |   |   |   |   |   +--:(id)
  |   |   |   |   |   |   +--rw group* [group-id]
  |   |   |   |   |   |   |   +--rw group-id      string
  |   |   |   |   |   +--:(all-accesses)
  |   |   |   |   |   |   +--rw all-other-accesses?  empty
  |   |   |   |   +--:(all-groups)
  |   |   |   |   |   +--rw all-other-groups?        empty
  +--rw bearer
  |   +--rw requested-type {requested-type}?
  |   |   +--rw requested-type?  string
  |   |   +--rw strict?          boolean
  |   +--rw always-on?           boolean {always-on}?
  |   +--rw bearer-reference?    string {bearer-reference
}?
  |   +--rw connection
  |   |   +--rw encapsulation-type?  identityref
  |   |   +--rw tagged-interface
  |   |   |   +--rw type?              identityref
  |   |   |   +--rw dot1q-vlan-tagged {dot1q}?
  |   |   |   |   +--rw tag-type?      identityref
  |   |   |   |   +--rw cvlan-id?      uint16

```



```

| | | +--rw priority-tagged
| | | | +---rw tag-type? identityref
+--rw qinq {qinq}?
| | | | +---rw tag-type? identityref
| | | | +---rw svlan-id uint16
| | | | +---rw cvlan-id uint16
+--rw qinany {qinany}?
| | | | +---rw tag-type? identityref
| | | | +---rw svlan-id uint16
+--rw vxlan {vxlan}?
| | | | +---rw vni-id uint32
| | | | +---rw peer-mode? identityref
| | | | +---rw peer-list* [peer-ip]
| | | | | +---rw peer-ip inet:ip-address
+--rw pseudowire
| | | | +---rw vcid? uint32
+--rw ip-connection
| | | +--rw ipv4 {ipv4}?
| | | | +---rw address-allocation-type? identityref
| | | | +--rw provider-dhcp
| | | | | +---rw provider-address? ine
t:ipv4-address
| | | | +---rw prefix-length? uin
t8
| | | | +--rw (address-assign)?
| | | | | +---:(number)
| | | | | | +---rw number-of-dynamic-address? uin
t16
| | | | | +---:(explicit)
| | | | | | +--rw customer-addresses
| | | | | | | +--rw address-group* [group-id]
| | | | | | | +---rw group-id string
| | | | | | | +---rw start-address? inet:ipv4
-address
| | | | | | | +---rw end-address? inet:ipv4
-address
| | | | +--rw dhcp-relay
| | | | | +---rw provider-address? inet:ipv4-add
ress
| | | | +---rw prefix-length? uint8
| | | | +--rw customer-dhcp-servers
| | | | | +---rw server-ip-address* inet:ipv4-addr
ess
| | | | +--rw addresses
| | | | | +---rw provider-address? inet:ipv4-address
| | | | | +---rw customer-address? inet:ipv4-address
| | | | | +---rw prefix-length? uint8
+--rw ipv6 {ipv6}?
| | | | +---rw address-allocation-type? identityref
| | | | +--rw provider-dhcp
| | | | | +---rw provider-address? ine
t:ipv6-address
| | | | +---rw prefix-length? uin
t8
| | | | +--rw (address-assign)?
| | | | | +---:(number)
| | | | | | +---rw number-of-dynamic-address? uin
t16

```

```

+---:(explicit)
+---rw customer-addresses
+---rw address-group* [group-id]
+---rw group-id string
+---rw start-address? inet:ipv6
- address
+---rw end-address? inet:ipv6
- address
+---rw dhcp-relay
+---rw provider-address? inet:ipv6-add
ress
+---rw prefix-length? uint8
+---rw customer-dhcp-servers
+---rw server-ip-address* inet:ipv6-addr
ess
+---rw addresses
+---rw provider-address? inet:ipv6-address
+---rw customer-address? inet:ipv6-address
+---rw prefix-length? uint8
+---rw oam
+---rw bfd {bfd}?
+---rw enabled? boolean
+---rw (holdtime)?
+---:(fixed)
| +---rw fixed-value? uint32
+---:(profile)
+---rw profile-name? -> /l3vpn-ntw/vpn-
n-profiles/valid-provider-identifiers/bfd-profile-identifier/id
+---rw security
+---rw authentication
+---rw encryption {encryption}?
| +---rw enabled? boolean
| +---rw layer? enumeration
+---rw encryption-profile
+---rw (profile)?
| +---:(provider-profile)
| | +---rw profile-name? -> /l3vpn-ntw/vpn-
n-profiles/valid-provider-identifiers/encryption-profile-identifier/id
+---:(customer-profile)
+---rw algorithm? string
+---rw (key-type)?
+---:(psk)
+---rw preshared-key? string
+---rw service
+---rw svc-input-bandwidth uint64
+---rw svc-output-bandwidth uint64
+---rw svc-mtu uint16
+---rw qos {qos}?
| +---rw qos-classification-policy
| | +---rw rule* [id]
| | +---rw id string
| | +---rw (match-type)?
| | | +---:(match-flow)
| | | +---rw match-flow

```

dscp						+--rw dscp?	inet:
						+--rw dot1p?	uint8
ipv4-prefix						+--rw ipv4-src-prefix?	inet:
ipv6-prefix						+--rw ipv6-src-prefix?	inet:
ipv4-prefix						+--rw ipv4-dst-prefix?	inet:
ipv6-prefix						+--rw ipv6-dst-prefix?	inet:
port-number						+--rw l4-src-port?	inet:
d {target-sites}?						+--rw target-sites*	svc-i
						+--rw l4-src-port-range	
-number						+--rw lower-port?	inet:port
-number						+--rw upper-port?	inet:port
port-number						+--rw l4-dst-port?	inet:
						+--rw l4-dst-port-range	
-number						+--rw lower-port?	inet:port
-number						+--rw upper-port?	inet:port
						+--rw protocol-field?	union
						+--:(match-application)	
ref						+--rw match-application?	identity
						+--rw target-class-id?	string
						+--rw qos-profile	
						+--rw (qos-profile)?	
						+--:(standard)	
						+--rw profile?	-> /l3vpn-ntw/vpn-p
rofiles/valid-provider-identifiers/qos-profile-identifier/id						+--rw direction?	identityref
						+--:(custom)	
						+--rw classes {qos-custom}?	
						+--rw class* [class-id]	
						+--rw class-id	string
						+--rw direction?	identityref
						+--rw rate-limit?	decimal64
						+--rw latency	
						+--rw (flavor)?	
						+--:(lowest)	
y? empty						+--rw use-lowest-latenc	
						+--:(boundary)	
uint16						+--rw latency-boundary?	
						+--rw jitter	
						+--rw (flavor)?	
						+--:(lowest)	
? empty						+--rw use-lowest-jitter	
						+--:(boundary)	
uint32						+--rw latency-boundary?	
						+--rw bandwidth	
decimal64						+--rw guaranteed-bw-percent	

```
empty          | |          +--rw end-to-end?
               | +--rw carrierscarrier {carrierscarrier}?
               | | +--rw signalling-type?  enumeration
               | +--rw multicast {multicast}?
```

```

|         +--rw multicast-site-type?          enumeration
|         +--rw multicast-address-family
|         |   +--rw ipv4?    boolean {ipv4}?
|         |   +--rw ipv6?    boolean {ipv6}?
|         +--rw protocol-type?                enumeration
+--rw routing-protocols
|   +--rw routing-protocol* [type]
|   +--rw type                identityref
|   +--rw routing-profiles* [id]
|   |   +--rw id              -> /l3vpn-ntw/vpn-profiles/val
id-provider-identifiers/routing-profile-identifier/id
|   |   +--rw type?          ie-type
+--rw ospf {rtg-ospf}?
|   +--rw address-family*    address-family
|   +--rw area-address        yang:dotted-quad
|   +--rw metric?            uint16
|   +--rw mtu?               uint16
|   +--rw security
|   |   +--rw auth-key?      string
+--rw sham-links {rtg-ospf-sham-link}?
|   +--rw sham-link* [target-site]
|   |   +--rw target-site    svc-id
|   |   +--rw metric?        uint16
+--rw bgp {rtg-bgp}?
|   +--rw autonomous-system  uint32
|   +--rw address-family*    address-family
|   +--rw neighbor?          inet:ip-address
|   +--rw multihop?          uint8
|   +--rw security
|   |   +--rw auth-key?      string
+--rw static
|   +--rw cascaded-lan-prefixes
|   |   +--rw ipv4-lan-prefixes* [lan next-hop] {
ipv4}?
|   |   |   +--rw lan          inet:ipv4-prefix
|   |   |   +--rw lan-tag?     string
|   |   |   +--rw next-hop     inet:ipv4-address
|   |   +--rw ipv6-lan-prefixes* [lan next-hop] {
ipv6}?
|   |   |   +--rw lan          inet:ipv6-prefix
|   |   |   +--rw lan-tag?     string
|   |   |   +--rw next-hop     inet:ipv6-address
|   +--rw rip {rtg-rip}?
|   |   +--rw address-family*  address-family
+--rw vrrp {rtg-vrrp}?
|   +--rw address-family*      address-family
+--rw availability
|   +--rw access-priority?     uint32
+--rw node-id?                -> /l3vpn-ntw/vpn-s
services/vpn-service/vpn-nodes/vpn-node/vpn-node-id
+--rw service-id?            -> /l3vpn-ntw/vpn-s
services/vpn-service/vpn-id
+--rw access-group-id?        yang:uuid

```

Figure 3

5. YANG MODULE

```
|----- EXAMPLE -----|  
  
<CODE BEGINS>file "ietf-l3vpn-ntw@2019-09-13.YANG"  
module ietf-l3vpn-ntw {  
  yang-version 1.1;  
  namespace "urn:ietf:params:xml:ns:yang:ietf-l3vpn-ntw";  
  prefix l3vpn-ntw;  
  import ietf-inet-types {  
    prefix inet;  
  }  
  import ietf-yang-types {  
    prefix yang;  
  }  
  import ietf-netconf-acm {  
    prefix nacm;  
  }  
  import ietf-routing-types {  
    prefix rt-types;  
  }  
  organization  
    "Individual draft";  
  contact  
    "Currently discussed in WG List: <mailto:opsawg@ietf.org>  
    Editor: Oscar Gonzalez de Dios  
      <mailto:oscar.gonzalezdedios@telefonica.com>";  
  
  description  
    "This YANG module defines a generic network-oriented model  
    for the configuration of Layer 3 VPNs.  
    Copyright (c) 2019 IETF Trust and the persons identified as  
    authors of the code. All rights reserved.  
  
    Redistribution and use in source and binary forms, with or  
    without modification, is permitted pursuant to, and subject to  
    the license terms contained in, the Simplified BSD License set  
    forth in Section 4.c of the IETF Trust's Legal Provisions  
    Relating to IETF Documents  
    (https://trustee.ietf.org/license-info).  
  
    This version of this YANG module is part of RFC XXXX  
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself  
    for full legal notices.  
  
    The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
```

NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document are to be interpreted as described in BCP 14 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.";

```
revision 2019-09-24 {
  description
    "Initial document. The document as a whole is based on L3SM
    module, defined in RFC 8299, modified to fit the requirements
    of the platforms at the network layer.";
  reference
    "RFC 8049.";
}
/* Features */
feature cloud-access {
  description
    "Allows the VPN to connect to a CSP.";
}
feature multicast {
  description
    "Enables multicast capabilities in a VPN.";
}
feature ipv4 {
  description
    "Enables IPv4 support in a VPN.";
}
feature ipv6 {
  description
    "Enables IPv6 support in a VPN.";
}
feature lan-tag {
  description
    "Enables LAN Tag support in a VPN Policy filter.";
}
feature carrierscarrier {
  description
    "Enables support of CsC.";
}
feature extranet-vpn {
  description
    "Enables support of extranet VPNs.";
}
feature site-diversity {
  description
    "Enables support of site diversity constraints.";
}
feature encryption {
  description
```

```
"Enables support of encryption.";
}
feature qos {
  description
    "Enables support of classes of services.";
}
feature qos-custom {
  description
    "Enables support of the custom QoS profile.";
}
feature rtg-bgp {
  description
    "Enables support of the BGP routing protocol.";
}
feature rtg-rip {
  description
    "Enables support of the RIP routing protocol.";
}
feature rtg-ospf {
  description
    "Enables support of the OSPF routing protocol.";
}
feature rtg-ospf-sham-link {
  description
    "Enables support of OSPF sham links.";
}
feature rtg-vrrp {
  description
    "Enables support of the VRRP routing protocol.";
}
feature fast-reroute {
  description
    "Enables support of Fast Reroute.";
}
feature bfd {
  description
    "Enables support of BFD.";
}
feature always-on {
  description
    "Enables support of the 'always-on' access constraint.";
}
feature requested-type {
  description
    "Enables support of the 'requested-type' access constraint.";
}
feature bearer-reference {
  description
```



```
    "Enables support of the 'bearer-reference' access constraint.";
}
feature target-sites {
    description
    "Enables support of the 'target-sites' match flow parameter.";
}
feature input-bw {
    description
    "Enables support of the 'input-bw' limit.";
}
feature dot1q {
    description
    "Enables support of the 'dot1q' encapsulation.";
}
feature qinq {
    description
    "Enables support of the 'qinq' encapsulation.";
}
feature qinany {
    description
    "Enables support of the 'qinany' encapsulation.";
}
feature vxlan {
    description
    "Enables support of the 'vxlan' encapsulation.";
}
/* Typedefs */
typedef svc-id {
    type string;
    description
    "Defines a type of service component identifier.";
}
typedef template-id {
    type string;
    description
    "Defines a type of service template identifier.";
}
typedef address-family {
    type enumeration {
        enum ipv4 {
            description
            "IPv4 address family.";
        }
        enum ipv6 {
            description
            "IPv6 address family.";
        }
    }
}
```

```
description
  "Defines a type for the address family.";
}

typedef ie-type {
  type enumeration {
    enum "import" {
      value 0;
      description "Import routing profile.";
    }
    enum "export" {
      value 1;
      description "Export routing profile";
    }
    enum "both" {
      value 2;
      description "Import/Export routing profile";
    }
  }
  description
    "Defines Import-Export routing profiles.
    Those are able to be reused between vpn-nodes";
}

typedef operational-type {
  type enumeration {
    enum "up" {
      value 0;
      description "Operational status UP.";
    }
    enum "down" {
      value 1;
      description "Operational status DOWN";
    }
    enum "unknown" {
      value 2;
      description "Operational status UNKNOWN";
    }
  }
  description
    "This is a read-only attribute used to determine the
    status of a particular element";
}

/* Identities */
identity site-network-access-type {
  description
    "Base identity for site-network-access type.";
}
```

```
}
identity point-to-point {
  base site-network-access-type;
  description
    "Identity for point-to-point connection.";
}
/* Extension */
identity pseudowire {
  base site-network-access-type;
  description
    "Identity for pseudowire connection.";
}
/* End of Extension */
identity multipoint {
  base site-network-access-type;
  description
    "Identity for multipoint connection.
    Example: Ethernet broadcast segment.";
}
identity placement-diversity {
  description
    "Base identity for site placement constraints.";
}
identity bearer-diverse {
  base placement-diversity;
  description
    "Identity for bearer diversity.
    The bearers should not use common elements.";
}
identity pe-diverse {
  base placement-diversity;
  description
    "Identity for PE diversity.";
}
identity pop-diverse {
  base placement-diversity;
  description
    "Identity for POP diversity.";
}
identity linecard-diverse {
  base placement-diversity;
  description
    "Identity for linecard diversity.";
}
identity same-pe {
  base placement-diversity;
  description
    "Identity for having sites connected on the same PE.";
```

```
}
identity same-bearer {
  base placement-diversity;
  description
    "Identity for having sites connected using the same bearer.";
}
identity customer-application {
  description
    "Base identity for customer application.";
}
identity web {
  base customer-application;
  description
    "Identity for Web application (e.g., HTTP, HTTPS).";
}
identity mail {
  base customer-application;
  description
    "Identity for mail application.";
}
identity file-transfer {
  base customer-application;
  description
    "Identity for file transfer application (e.g., FTP, SFTP).";
}
identity database {
  base customer-application;
  description
    "Identity for database application.";
}
identity social {
  base customer-application;
  description
    "Identity for social-network application.";
}
identity games {
  base customer-application;
  description
    "Identity for gaming application.";
}
identity p2p {
  base customer-application;
  description
    "Identity for peer-to-peer application.";
}
identity network-management {
  base customer-application;
  description
```

```
"Identity for management application
(e.g., Telnet, syslog, SNMP).";
}
identity voice {
  base customer-application;
  description
  "Identity for voice application.";
}
identity video {
  base customer-application;
  description
  "Identity for video conference application.";
}
identity embb {
  base customer-application;
  description
  "Identity for an enhanced Mobile Broadband (eMBB)
  application. Note that an eMBB application demands
  network performance with a wide variety of
  characteristics, such as data rate, latency,
  loss rate, reliability, and many other parameters.";
}
identity urllc {
  base customer-application;
  description
  "Identity for an Ultra-Reliable and Low Latency
  Communications (URLLC) application. Note that a
  URLLC application demands network performance
  with a wide variety of characteristics, such as latency,
  reliability, and many other parameters.";
}
identity mmhc {
  base customer-application;
  description
  "Identity for a massive Machine Type
  Communications (mMTC) application. Note that an
  mMTC application demands network performance
  with a wide variety of characteristics, such as data
  rate, latency, loss rate, reliability, and many
  other parameters.";
}
identity site-vpn-flavor {
  description
  "Base identity for the site VPN service flavor.";
}
identity site-vpn-flavor-single {
  base site-vpn-flavor;
  description
```

```
"Base identity for the site VPN service flavor.
Used when the site belongs to only one VPN.";
}
identity site-vpn-flavor-multi {
  base site-vpn-flavor;
  description
    "Base identity for the site VPN service flavor.
    Used when a logical connection of a site
    belongs to multiple VPNs.";
}
identity site-vpn-flavor-sub {
  base site-vpn-flavor;
  description
    "Base identity for the site VPN service flavor.
    Used when a site has multiple logical connections.
    Each connection may belong to different multiple VPNs.";
}
identity site-vpn-flavor-nni {
  base site-vpn-flavor;
  description
    "Base identity for the site VPN service flavor.
    Used to describe an NNI option A connection.";
}
identity management {
  description
    "Base identity for site management scheme.";
}
identity co-managed {
  base management;
  description
    "Base identity for co-managed site.";
}
identity customer-managed {
  base management;
  description
    "Base identity for customer-managed site.";
}
identity provider-managed {
  base management;
  description
    "Base identity for provider-managed site.";
}
identity address-allocation-type {
  description
    "Base identity for address-allocation-type for PE-CE link.";
}
identity provider-dhcp {
  base address-allocation-type;
```

```
    description
    "Provider network provides DHCP service to customer.";
}
identity provider-dhcp-relay {
    base address-allocation-type;
    description
    "Provider network provides DHCP relay service to customer.";
}
identity provider-dhcp-slaac {
    base address-allocation-type;
    description
    "Provider network provides DHCP service to customer,
    as well as SLAAC.";
}
identity static-address {
    base address-allocation-type;
    description
    "Provider-to-customer addressing is static.";
}
identity slaac {
    base address-allocation-type;
    description
    "Use IPv6 SLAAC.";
}
identity site-role {
    description
    "Base identity for site type.";
}
identity any-to-any-role {
    base site-role;
    description
    "Site in an any-to-any IP VPN.";
}
identity spoke-role {
    base site-role;
    description
    "Spoke site in a Hub-and-Spoke IP VPN.";
}
identity hub-role {
    base site-role;
    description
    "Hub site in a Hub-and-Spoke IP VPN.";
}
identity vpn-topology {
    description
    "Base identity for VPN topology.";
}
identity any-to-any {
```

```
base vpn-topology;
description
  "Identity for any-to-any VPN topology.";
}
identity hub-spoke {
  base vpn-topology;
  description
    "Identity for Hub-and-Spoke VPN topology.";
}
identity hub-spoke-disjoint {
  base vpn-topology;
  description
    "Identity for Hub-and-Spoke VPN topology
    where Hubs cannot communicate with each other.";
}
identity multicast-tree-type {
  description
    "Base identity for multicast tree type.";
}
identity ssm-tree-type {
  base multicast-tree-type;
  description
    "Identity for SSM tree type.";
}
identity asm-tree-type {
  base multicast-tree-type;
  description
    "Identity for ASM tree type.";
}
identity bidir-tree-type {
  base multicast-tree-type;
  description
    "Identity for bidirectional tree type.";
}
identity multicast-rp-discovery-type {
  description
    "Base identity for RP discovery type.";
}
identity auto-rp {
  base multicast-rp-discovery-type;
  description
    "Base identity for Auto-RP discovery type.";
}
identity static-rp {
  base multicast-rp-discovery-type;
  description
    "Base identity for static type.";
}
```



```
identity bsr-rp {
  base multicast-rp-discovery-type;
  description
    "Base identity for BSR discovery type.";
}
identity routing-protocol-type {
  description
    "Base identity for routing protocol type.";
}
identity ospf {
  base routing-protocol-type;
  description
    "Identity for OSPF protocol type.";
}
identity bgp {
  base routing-protocol-type;
  description
    "Identity for BGP protocol type.";
}
identity static {
  base routing-protocol-type;
  description
    "Identity for static routing protocol type.";
}
identity rip {
  base routing-protocol-type;
  description
    "Identity for RIP protocol type.";
}
identity vrrp {
  base routing-protocol-type;
  description
    "Identity for VRRP protocol type.
    This is to be used when LANs are directly connected
    to PE routers.";
}
identity direct {
  base routing-protocol-type;
  description
    "Identity for direct protocol type.";
}
identity protocol-type {
  description
    "Base identity for protocol field type.";
}
identity tcp {
  base protocol-type;
  description
```

```
"TCP protocol type.";
}
identity udp {
  base protocol-type;
  description
    "UDP protocol type.";
}

identity icmp {
  base protocol-type;
  description
    "ICMP protocol type.";
}
identity icmp6 {
  base protocol-type;
  description
    "ICMPv6 protocol type.";
}
identity gre {
  base protocol-type;
  description
    "GRE protocol type.";
}
identity ipip {
  base protocol-type;
  description
    "IP-in-IP protocol type.";
}
identity hop-by-hop {
  base protocol-type;
  description
    "Hop-by-Hop IPv6 header type.";
}
identity routing {
  base protocol-type;
  description
    "Routing IPv6 header type.";
}
identity esp {
  base protocol-type;
  description
    "ESP header type.";
}
identity ah {
  base protocol-type;
  description
    "AH header type.";
}
```

```
identity vpn-policy-filter-type {
  description
    "Base identity for VPN Policy filter type.";
}
identity ipv4 {
  base vpn-policy-filter-type;
  description
    "Identity for IPv4 Prefix filter type.";
}
identity ipv6 {
  base vpn-policy-filter-type;
  description
    "Identity for IPv6 Prefix filter type.";
}
identity lan {
  base vpn-policy-filter-type;
  description
    "Identity for LAN Tag filter type.";
}

identity qos-profile-direction {
  description
    "Base identity for QoS profile direction.";
}

identity site-to-wan {
  base qos-profile-direction;
  description
    "Identity for Site-to-WAN direction.";
}
identity wan-to-site {
  base qos-profile-direction;
  description
    "Identity for WAN-to-Site direction.";
}
identity both {
  base qos-profile-direction;
  description
    "Identity for both WAN-to-Site direction
    and Site-to-WAN direction.";
}

/* Extended Identities */

identity encapsulation-type {
  description
    "Identity for the encapsulation type.";
}
```

```
identity untagged-int {
    base encapsulation-type;
    description
        "Identity for Ethernet type.";
}

identity tagged-int {
    base encapsulation-type;
    description
        "Identity for the VLAN type.";
}

identity eth-inf-type {
    description
        "Identity of the Ethernet interface type.";
}

identity tagged {
    base eth-inf-type;
    description
        "Identity of the tagged interface type.";
}

identity untagged {
    base eth-inf-type;
    description
        "Identity of the untagged interface type.";
}

identity lag {
    base eth-inf-type;
    description
        "Identity of the LAG interface type.";
}

identity bearer-inf-type {
    description
        "Identity for the bearer interface type.";
}

identity port-id {
    base bearer-inf-type;
    description
        "Identity for the priority-tagged interface.";
}

identity lag-id {
    base bearer-inf-type;
    description
```

```
    "Identity for the priority-tagged interface.";
}

identity tagged-inf-type {
    description
        "Identity for the tagged interface type.";
}

identity priority-tagged {
    base tagged-inf-type;
    description
        "Identity for the priority-tagged interface.";
}

identity qinq {
    base tagged-inf-type;
    description
        "Identity for the QinQ tagged interface.";
}

identity dot1q {
    base tagged-inf-type;
    description
        "Identity for the dot1Q VLAN tagged interface.";
}

identity qinany {
    base tagged-inf-type;
    description
        "Identity for the QinAny tagged interface.";
}

identity vxlan {
    base tagged-inf-type;
    description
        "Identity for the VXLAN tagged interface.";
}

identity tag-type {
    description
        "Base identity from which all tag types are derived.";
}

identity c-vlan {
    base tag-type;
    description
        "A CVLAN tag, normally using the 0x8100 Ethertype.";
}
```

```
identity s-vlan {
    base tag-type;
    description
        "An SVLAN tag.";
}

identity c-s-vlan {
    base tag-type;
    description
        "Using both a CVLAN tag and an SVLAN tag.";
}

identity vxlan-peer-mode {
    description
        "Base identity for the VXLAN peer mode.";
}

identity static-mode {
    base vxlan-peer-mode;
    description
        "Identity for VXLAN access in the static mode.";
}

identity bgp-mode {
    base vxlan-peer-mode;
    description
        "Identity for VXLAN access by BGP EVPN learning.";
}

identity bw-direction {
    description
        "Identity for the bandwidth direction.";
}

identity input-bw {
    base bw-direction;
    description
        "Identity for the input bandwidth.";
}

identity output-bw {
    base bw-direction;
    description
        "Identity for the output bandwidth.";
}

identity bw-type {
    description
```

```
    "Identity of the bandwidth type.";
}

identity bw-per-cos {
    base bw-type;
    description
        "Bandwidth is per CoS.";
}

identity bw-per-port {
    base bw-type;
    description
        "Bandwidth is per site network access.";
}

identity bw-per-site {
    base bw-type;
    description
        "Bandwidth is per site.  It is applicable to
        all the site network accesses within the site.";
}

identity bw-per-svc {
    base bw-type;
    description
        "Bandwidth is per VPN service.";
}

/* Groupings */
grouping vpn-service-cloud-access {
    container cloud-accesses {
        if-feature cloud-access;
        list cloud-access {
            key cloud-identifier;
            leaf cloud-identifier {
                type leafref {
                    path "/l3vpn-ntw/vpn-profiles/" +
                        "valid-provider-identifiers/cloud-identifier/id";
                }
            }
            description
                "Identification of cloud service.
                Local administration meaning.";
        }
        choice list-flavor {
            case permit-any {
                leaf permit-any {
                    type empty;
                    description
```

```
    "Allows all sites.";
  }
}
case deny-any-except {
  leaf-list permit-site {
    type leafref {
      path "/l3vpn-ntw/sites/site/site-id";
    }
    description
      "Site ID to be authorized.";
  }
}
case permit-any-except {
  leaf-list deny-site {
    type leafref {
      path "/l3vpn-ntw/sites/site/site-id";
    }
    description
      "Site ID to be denied.";
  }
}
description
  "Choice for cloud access policy.  By
  default, all sites in the IP VPN MUST
  be authorized to access the cloud.";
}
container address-translation {
  container nat44 {
    leaf enabled {
      type boolean;
      default false;
      description
        "Controls whether or not Network address
        translation from IPv4 to IPv4 (NAT44)
        [RFC3022] is required.";
    }
  }
  leaf nat44-customer-address {
    type inet:ipv4-address;
    description
      "Address to be used for network address
      translation from IPv4 to IPv4.  This is
      to be used if the customer is providing
      the IPv4 address.  If the customer address
      is not set, the model assumes that the
      provider will allocate the address.";
  }
  description
    "IPv4-to-IPv4 translation.";
```



```
    }
    description
    "Container for NAT.";
  }
  description
  "Cloud access configuration.";
}
description
"Container for cloud access configurations.";
}
description
"Grouping for VPN cloud definition.";
}
grouping multicast-rp-group-cfg {
  choice group-format {
    mandatory true;
    case singleaddress {
      leaf group-address {
        type inet:ip-address;
        description
        "A single multicast group address.";
      }
    }
    case startend {
      leaf group-start {
        type inet:ip-address;
        description
        "The first multicast group address in
        the multicast group address range.";
      }
      leaf group-end {
        type inet:ip-address;
        description
        "The last multicast group address in
        the multicast group address range.";
      }
    }
  }
  description
  "Choice for multicast group format.";
}
description
"This grouping defines multicast group or
multicast groups for RP-to-group mapping.";
}
grouping vpn-service-multicast {
  container multicast {
    if-feature multicast;
    leaf enabled {
```

```
    type boolean;
    default false;
    description
    "Enables multicast.";
}
container customer-tree-flavors {
  leaf-list tree-flavor {
    type identityref {
      base multicast-tree-type;
    }
    description
    "Type of tree to be used.";
  }
  description
  "Type of trees used by customer.";
}
container rp {
  container rp-group-mappings {
    list rp-group-mapping {
      key id;
      leaf id {
        type uint16;
        description
        "Unique identifier for the mapping.";
      }
    }
    container provider-managed {
      leaf enabled {
        type boolean;
        default false;
        description
        "Set to true if the Rendezvous Point (RP)
        must be a provider-managed node. Set to false
        if it is a customer-managed node.";
      }
      leaf rp-redundancy {
        type boolean;
        default false;
        description
        "If true, a redundancy mechanism for the RP
        is required.";
      }
    }
    leaf optimal-traffic-delivery {
      type boolean;
      default false;
      description
      "If true, the SP must ensure that
      traffic uses an optimal path. An SP may use
      Anycast RP or RP-tree-to-SPT switchover
```

```
    architectures.";
  }
  description
    "Parameters for a provider-managed RP.";
}
leaf rp-address {
  when "../provider-managed/enabled = 'false'" {
    description
      "Relevant when the RP is not provider-managed.";
  }
  type inet:ip-address;
  mandatory true;
  description
    "Defines the address of the RP.
    Used if the RP is customer-managed.";
}
container groups {
  list group {
    key id;
    leaf id {
      type uint16;
      description
        "Identifier for the group.";
    }
    uses multicast-rp-group-cfg;
    description
      "List of multicast groups.";
  }
  description
    "Multicast groups associated with the RP.";
}
description
  "List of RP-to-group mappings.";
}
description
  "RP-to-group mappings parameters.";
}
container rp-discovery {
  leaf rp-discovery-type {
    type identityref {
      base multicast-rp-discovery-type;
    }
    default static-rp;
    description
      "Type of RP discovery used.";
  }
}
container bsr-candidates {
  when "derived-from-or-self(../rp-discovery-type, "+
```

```
    "'l3vpn-ntw:bsr-rp')" {
      description
        "Only applicable if discovery type
        is BSR-RP.";
    }
    leaf-list bsr-candidate-address {
      type inet:ip-address;
      description
        "Address of BSR candidate.";
    }
    description
      "Container for List of Customer
      BSR candidate's addresses.";
  }
  description
    "RP discovery parameters.";
}
description
  "RP parameters.";
}
description
  "Multicast global parameters for the VPN service.";
}
description
  "Grouping for multicast VPN definition.";
}
grouping vpn-service-mpls {
  leaf carrierscarrier {
    if-feature carrierscarrier;
    type boolean;
    default false;
    description
      "The VPN is using CsC, and so MPLS is required.";
  }
  description
    "Grouping for MPLS CsC definition.";
}
grouping customer-location-info {
  container locations {
    list location {
      key location-id;
      leaf location-id {
        type svc-id;
        description
          "Identifier for a particular location.";
      }
      leaf address {
        type string;
      }
    }
  }
}
```

```
    description
    "Address (number and street) of the site.";
  }
  leaf postal-code {
    type string;
    description
    "Postal code of the site.";
  }
  leaf state {
    type string;
    description
    "State of the site.  This leaf can also be
    used to describe a region for a country that
    does not have states.";
  }
  leaf city {
    type string;
    description
    "City of the site.";
  }
  leaf country-code {
    type string {
      pattern '[A-Z]{2}';
    }
    description
    "Country of the site.
    Expressed as ISO ALPHA-2 code.";
  }
  description
  "Location of the site.";
}
description
"List of locations for the site.";
}
description
"This grouping defines customer location parameters.";
}
grouping site-group {
  container groups {
    list group {
      key group-id;
      leaf group-id {
        type string;
        description
        "Group-id the site belongs to.";
      }
    }
    description
    "List of group-ids.";
  }
}
```

```
    }
    description
    "Groups the site or site-network-access belongs to.";
  }
  description
  "Grouping definition to assign
  group-ids to site or site-network-access.";
}
grouping site-diversity {
  container site-diversity {
    if-feature site-diversity;
    uses site-group;
    description
    "Diversity constraint type. All
    site-network-accesses will inherit
    the group values defined here.";
  }
  description
  "This grouping defines site
  diversity parameters.";
}
grouping access-diversity {
  container access-diversity {
    if-feature site-diversity;
    uses site-group;
    container constraints {
      list constraint {
        key constraint-type;
        leaf constraint-type {
          type identityref {
            base placement-diversity;
          }
        }
        description
        "Diversity constraint type.";
      }
    }
    container target {
      choice target-flavor {
        default id;
        case id {
          list group {
            key group-id;
            leaf group-id {
              type string;
              description
              "The constraint will be applied against
              this particular group-id for this site
              network access level.";
            }
          }
        }
      }
    }
  }
}
```

```
        description
        "List of group-ids associated with one specific
        constraint for this site network access level.";
    }
}
case all-accesses {
    leaf all-other-accesses {
        type empty;
        description
        "The constraint will be applied against
        all other site network accesses of this site.";
    }
}
case all-groups {
    leaf all-other-groups {
        type empty;
        description
        "The constraint will be applied against
        all other groups managed by the customer.";
    }
}
description
"Choice for the target flavor definition.";
}
description
"The constraint will be applied against a
Specific target, and the target can be a list
of group-ids, all other site network accesses of
this site, or all other groups managed by the
customer.";
}
description
"List of constraints.";
}
description
"Placement constraints for this site network access.";
}
description
"Diversity parameters.";
}
description
"This grouping defines access diversity parameters.";
}
grouping operational-requirements {
    leaf requested-site-start {
        type yang:date-and-time;
        description
        "Optional leaf indicating requested date and
```

```
        time when the service at a particular site is
        expected to start.";
    }

    leaf requested-site-stop {
        type yang:date-and-time;
        description
            "Optional leaf indicating requested date and
            time when the service at a particular site is
            expected to stop.";
    }
    description
        "This grouping defines some operational
        parameters.";
}
grouping operational-requirements-ops {
    leaf actual-site-start {
        type yang:date-and-time;
        config false;
        description
            "Optional leaf indicating actual date and
            time when the service at a particular site
            actually started.";
    }
    leaf actual-site-stop {
        type yang:date-and-time;
        config false;
        description
            "Optional leaf indicating actual date and
            time when the service at a particular site
            actually stopped.";
    }
    description
        "This grouping defines some operational
        parameters.";
}
grouping flow-definition {
    container match-flow {
        leaf dscp {
            type inet:dscp;
            description
                "DSCP value.";
        }
        leaf dot1p {
            type uint8 {
                range "0..7";
            }
            description
```



```
    "802.1p matching.";
}
leaf ipv4-src-prefix {
    type inet:ipv4-prefix;
    description
        "Match on IPv4 src address.";
}
leaf ipv6-src-prefix {
    type inet:ipv6-prefix;
    description
        "Match on IPv6 src address.";
}
leaf ipv4-dst-prefix {
    type inet:ipv4-prefix;
    description
        "Match on IPv4 dst address.";
}
leaf ipv6-dst-prefix {
    type inet:ipv6-prefix;
    description
        "Match on IPv6 dst address.";
}
leaf l4-src-port {
    type inet:port-number;
    must "current() < ../l4-src-port-range/lower-port or "+
        "current() > ../l4-src-port-range/upper-port" {
        description
            "If l4-src-port and l4-src-port-range/lower-port and
            upper-port are set at the same time, l4-src-port
            should not overlap with l4-src-port-range.";
    }
    description
        "Match on Layer 4 src port.";
}
leaf-list target-sites {
    if-feature target-sites;
    type svc-id;
    description
        "Identify a site as traffic destination.";
}
container l4-src-port-range {
    leaf lower-port {
        type inet:port-number;
        description
            "Lower boundary for port.";
    }
    leaf upper-port {
        type inet:port-number;
```

```
    must ". >= ../lower-port" {
      description
        "Upper boundary for port.  If it
         exists, the upper boundary must be
         higher than the lower boundary.";
    }
    description
      "Upper boundary for port.";
  }
  description
    "Match on Layer 4 src port range.  When
     only the lower-port is present, it represents
     a single port.  When both the lower-port and
     upper-port are specified, it implies
     a range inclusive of both values.";
}
leaf l4-dst-port {
  type inet:port-number;
  must "current() < ../l4-dst-port-range/lower-port or "+
       "current() > ../l4-dst-port-range/upper-port" {
    description
      "If l4-dst-port and l4-dst-port-range/lower-port
       and upper-port are set at the same time,
       l4-dst-port should not overlap with
       l4-src-port-range.";
  }
  description
    "Match on Layer 4 dst port.";
}
container l4-dst-port-range {
  leaf lower-port {
    type inet:port-number;
    description
      "Lower boundary for port.";
  }
  leaf upper-port {
    type inet:port-number;
    must ". >= ../lower-port" {
      description
        "Upper boundary must be
         higher than lower boundary.";
    }
    description
      "Upper boundary for port.  If it exists,
       upper boundary must be higher than lower
       boundary.";
  }
  description
```

```
    "Match on Layer 4 dst port range.  When only
    lower-port is present, it represents a single
    port.  When both lower-port and upper-port are
    specified, it implies a range inclusive of both
    values.";
  }
  leaf protocol-field {
    type union {
      type uint8;
      type identityref {
        base protocol-type;
      }
    }
    description
      "Match on IPv4 protocol or IPv6 Next Header field.";
  }
  description
    "Describes flow-matching criteria.";
}
description
  "Flow definition based on criteria.";
}
grouping site-service-basic {
  leaf svc-input-bandwidth {
    type uint64;
    units bps;
    mandatory true;
    description
      "From the customer site's perspective, the service
      input bandwidth of the connection or download
      bandwidth from the SP to the site.";
  }
  leaf svc-output-bandwidth {
    type uint64;
    units bps;
    mandatory true;
    description
      "From the customer site's perspective, the service
      output bandwidth of the connection or upload
      bandwidth from the site to the SP.";
  }
}
leaf svc-mtu {
  type uint16;
  units bytes;
  mandatory true;
  description
    "MTU at service level.  If the service is IP,
    it refers to the IP MTU.  If CsC is enabled,
```

```
    the requested 'svc-mtu' leaf will refer to the
    MPLS MTU and not to the IP MTU.";
}
description
"Defines basic service parameters for a site.";
}
grouping site-protection {
  container traffic-protection {
    if-feature fast-reroute;
    leaf enabled {
      type boolean;
      default false;
      description
        "Enables traffic protection of access link.";
    }
    description
      "Fast Reroute service parameters for the site.";
  }
  description
    "Defines protection service parameters for a site.";
}
grouping site-service-mpls {
  container carrierscarrier {
    if-feature carrierscarrier;
    leaf signalling-type {
      type enumeration {
        enum ldp {
          description
            "Use LDP as the signalling protocol
            between the PE and the CE. In this case,
            an IGP routing protocol must also be activated.";
        }
        enum bgp {
          description
            "Use BGP (as per RFC 8277) as the signalling protocol
            between the PE and the CE.
            In this case, BGP must also be configured as
            the routing protocol.";
        }
      }
    }
    default bgp;
    description
      "MPLS signalling type.";
  }
  description
    "This container is used when the customer provides
    MPLS-based services. This is only used in the case
    of CsC (i.e., a customer builds an MPLS service using
```

```
    an IP VPN to carry its traffic).";
  }
  description
    "Defines MPLS service parameters for a site.";
}
grouping site-service-qos-profile {
  container qos {
    if-feature qos;
    container qos-classification-policy {
      list rule {
        key id;
        ordered-by user;
        leaf id {
          type string;
          description
            "A description identifying the
             qos-classification-policy rule.";
        }
        choice match-type {
          default match-flow;
          case match-flow {
            uses flow-definition;
          }
          case match-application {
            leaf match-application {
              type identityref {
                base customer-application;
              }
              description
                "Defines the application to match.";
            }
          }
        }
        description
          "Choice for classification.";
      }
      leaf target-class-id {
        type string;
        description
          "Identification of the class of service.
           This identifier is internal to the administration.";
      }
      description
        "List of marking rules.";
    }
    description
      "Configuration of the traffic classification policy.";
  }
  container qos-profile {
```

```
choice qos-profile {
  description
    "Choice for QoS profile.
    Can be standard profile or customized profile.";
  case standard {
    description
      "Standard QoS profile.";
    leaf profile {
      type leafref {
        path "/l3vpn-ntw/vpn-profiles/valid-provider-identifiers"+
          "/qos-profile-identifier/id";
      }
      description
        "QoS profile to be used.";
    }
    leaf direction {
      type identityref {
        base qos-profile-direction;
        default both;
        description
          "The direction to which the QoS profile
          is applied.";
      }
    }
  }
  case custom {
    description
      "Customized QoS profile.";
    container classes {
      if-feature qos-custom;
      list class {
        key class-id;
        leaf class-id {
          type string;
          description
            "Identification of the class of service.
            This identifier is internal to the
            administration.";
        }
        leaf direction {
          type identityref {
            base qos-profile-direction;
          }
          default both;
          description
            "The direction to which the QoS profile
            is applied.";
        }
        leaf rate-limit {
```

```
        type decimal64 {
            fraction-digits 5;
            range "0..100";
        }

        units percent;
        description
            "To be used if the class must be rate-limited.
            Expressed as percentage of the service
            bandwidth.";
    }

    container latency {
        choice flavor {
            case lowest {
                leaf use-lowest-latency {
                    type empty;
                    description
                        "The traffic class should use the path with the
                        lowest latency.";
                }
            }
            case boundary {
                leaf latency-boundary {
                    type uint16;
                    units msec;
                    default 400;
                    description
                        "The traffic class should use a path with a
                        defined maximum latency.";
                }
            }
        }
        description
            "Latency constraint on the traffic class.";
    }
    description
        "Latency constraint on the traffic class.";
}

container jitter {
    choice flavor {
        case lowest {
            leaf use-lowest-jitter {
                type empty;
                description
                    "The traffic class should use the path with the
                    lowest jitter.";
            }
        }
        case boundary {
```

```
    leaf latency-boundary {
      type uint32;
      units usec;
      default 40000;
      description
        "The traffic class should use a path with a
        defined maximum jitter.";
    }
  }
  description
    "Jitter constraint on the traffic class.";
}
description
  "Jitter constraint on the traffic class.";
}
container bandwidth {
  leaf guaranteed-bw-percent {
    type decimal64 {
      fraction-digits 5;
      range "0..100";
    }
    units percent;
    mandatory true;
    description
      "To be used to define the guaranteed bandwidth
      as a percentage of the available service bandwidth.";
  }
  leaf end-to-end {
    type empty;
    description
      "Used if the bandwidth reservation
      must be done on the MPLS network too.";
  }
  description
    "Bandwidth constraint on the traffic class.";
}
description
  "List of classes of services.";
}
description
  "Container for list of classes of services.";
}
}
description
  "QoS profile configuration.";
}
description
```



```
    "QoS configuration.";
  }
  description
    "This grouping defines QoS parameters for a site.";
}
grouping site-security-authentication {
  container authentication {
    description
      "Authentication parameters.";
  }
  description
    "This grouping defines authentication parameters for a site.";
}
grouping site-security-encryption {
  container encryption {
    if-feature encryption;
    leaf enabled {
      type boolean;
      default false;
      description
        "If true, traffic encryption on the connection is required.";
    }
    leaf layer {
      when "../enabled = 'true'" {
        description
          "Require a value for layer when enabled is true.";
      }
      type enumeration {
        enum layer2 {
          description
            "Encryption will occur at Layer 2.";
        }
        enum layer3 {
          description
            "Encryption will occur at Layer 3.
            For example, IPsec may be used when
            a customer requests Layer 3 encryption.";
        }
      }
    }
    description
      "Layer on which encryption is applied.";
  }
  description
    "";
}
container encryption-profile {
  choice profile {
    case provider-profile {
```

```
    leaf profile-name {
      type leafref {
        path "/l3vpn-ntw/vpn-profiles/valid-provider-identifiers"+
          "/encryption-profile-identifier/id";
      }
      description
        "Name of the SP profile to be applied.";
    }
  }
  case customer-profile {
    leaf algorithm {
      type string;
      description
        "Encryption algorithm to be used.";
    }
  }
  description
    "";
}
choice key-type {
  default psk;
  case psk {
    leaf preshared-key {
      type string;
      description
        "Pre-Shared Key (PSK) coming from the customer.";
    }
  }
  description
    "Choice of encryption profile.
    The encryption profile can be the provider profile
    or customer profile.";
}
description
  "This grouping defines encryption parameters for a site.";
}
description
  "";
}
grouping site-attachment-bearer {
  container bearer {
    container requested-type {
      if-feature requested-type;
      leaf requested-type {
        type string;
        description
          "Type of requested bearer: Ethernet, DSL,
          Wireless, etc. Operator specific.";
      }
    }
  }
}
```

```
    }
    leaf strict {
      type boolean;
      default false;
      description
        "Defines whether requested-type is a preference
        or a strict requirement.";
    }
    description
      "Container for requested-type.";
  }
  leaf always-on {
    if-feature always-on;
    type boolean;
    default true;
    description
      "Request for an always-on access type.
      For example, this could mean no dial access type.";
  }

  leaf bearer-reference {
    if-feature bearer-reference;
    type string;
    description
      "This is an internal reference for the SP.";
  }
  description
    "Bearer-specific parameters.
    To be augmented.";

  uses ethernet-params;

  uses pseudowire-params {
    when "/l3vpn-ntw/sites/site/site-network-accesses" +
        "/site-network-access/site-network-access-type ='pseudowire'"
    {
      description "pseudowire specific parameters";
    }
  }
}
description
  "Defines physical properties of a site attachment.";
}
grouping site-routing {
  container routing-protocols {
    list routing-protocol {
      key type;
    }
  }
}
```

```
leaf type {
  type identityref {
    base routing-protocol-type;
  }
  description
  "Type of routing protocol.";
}

list routing-profiles {
  key "id";

  leaf id {
    type leafref {
      path "/l3vpn-ntw/vpn-profiles/valid-provider-identifiers"+
        "/routing-profile-identifier/id";
    }
    description
    "Routing profile to be used.";
  }

  leaf type {
    type ie-type;
    description
    "Import, export or both.";
  }
}

description
"Import or Export profile reference";
}

container ospf {
  when "derived-from-or-self(.. /type, 'l3vpn-ntw:ospf')" {
    description
    "Only applies when protocol is OSPF.";
  }
  if-feature rtg-ospf;
  leaf-list address-family {
    type address-family;
    min-elements "1";
    description
    "If OSPF is used on this site, this node
    contains a configured value. This node
    contains at least one address family
    to be activated.";
  }
  leaf area-address {
    type yang:dotted-quad;
    mandatory true;
  }
}
```

```
        description
        "Area address.";
    }
    leaf metric {
        type uint16;
        default 1;
        description
        "Metric of the PE-CE link. It is used
        in the routing state calculation and
        path selection.";
    }
}

/* Extension */

leaf mtu {
    type uint16;
    description "Maximum transmission unit for a given
    OSPF link.";
}

uses security-params;

/* End of Extension */

container sham-links {
    if-feature rtg-ospf-sham-link;
    list sham-link {
        key target-site;
        leaf target-site {
            type svc-id;
            description
            "Target site for the sham link connection.
            The site is referred to by its ID.";
        }
        leaf metric {
            type uint16;
            default 1;
            description
            "Metric of the sham link. It is used in
            the routing state calculation and path
            selection. The default value is set
            to 1.";
        }
        description
        "Creates a sham link with another site.";
    }
    description
```

```
    "List of sham links.";
  }
  description
  "OSPF-specific configuration.";
}
container bgp {
  when "derived-from-or-self(..../type, 'l3vpn-ntw:bgp')" {
    description
    "Only applies when protocol is BGP.";
  }
  if-feature rtg-bgp;
  leaf autonomous-system {
    type uint32;
    mandatory true;
    description
    "Customer AS number in case the customer
    requests BGP routing.";
  }
  leaf-list address-family {
    type address-family;
    min-elements "1";
    description
    "If BGP is used on this site, this node
    contains a configured value. This node
    contains at least one address family
    to be activated.";
  }
  /* Extension */
  leaf neighbor {
    type inet:ip-address;
    description
    "IP address of the BGP neighbor.";
  }

  leaf multihop {
    type uint8;
    description
    "Describes the number of hops allowed between the
    given BGP neighbor and the PE router.";
  }

  uses security-params;

  description
  "BGP-specific configuration.";
}
container static {
  when "derived-from-or-self(..../type, 'l3vpn-ntw:static')" {
```

```
description
  "Only applies when protocol is static.
  BGP activation requires the SP to know
  the address of the customer peer.  When
  BGP is enabled, the 'static-address'
  allocation type for the IP connection
  MUST be used.";
}
container cascaded-lan-prefixes {
  list ipv4-lan-prefixes {
    if-feature ipv4;
    key "lan next-hop";
    leaf lan {
      type inet:ipv4-prefix;
      description
        "LAN prefixes.";
    }
    leaf lan-tag {
      type string;
      description
        "Internal tag to be used in VPN policies.";
    }
    leaf next-hop {
      type inet:ipv4-address;
      description
        "Next-hop address to use on the customer side.";
    }
  }
  description
    "List of LAN prefixes for the site.";
}
list ipv6-lan-prefixes {
  if-feature ipv6;
  key "lan next-hop";
  leaf lan {
    type inet:ipv6-prefix;
    description
      "LAN prefixes.";
  }
  leaf lan-tag {
    type string;
    description
      "Internal tag to be used in VPN policies.";
  }
  leaf next-hop {
    type inet:ipv6-address;
    description
      "Next-hop address to use on the customer side.";
  }
}
```

```
    description
    "List of LAN prefixes for the site.";
  }
  description
  "LAN prefixes from the customer.";
}
description
"Configuration specific to static routing.";
}
container rip {
  when "derived-from-or-self(..../type, 'l3vpn-ntw:rip')" {
    description
    "Only applies when the protocol is RIP. For IPv4,
    the model assumes that RIP version 2 is used.";
  }
  if-feature rtg-rip;
  leaf-list address-family {
    type address-family;
    min-elements "1";
    description
    "If RIP is used on this site, this node
    contains a configured value. This node
    contains at least one address family
    to be activated.";
  }
  description
  "Configuration specific to RIP routing.";
}
container vrrp {
  when "derived-from-or-self(..../type, 'l3vpn-ntw:vrrp')" {
    description
    "Only applies when protocol is VRRP.";
  }
  if-feature rtg-vrrp;
  leaf-list address-family {
    type address-family;
    min-elements "1";
    description
    "If VRRP is used on this site, this node
    contains a configured value. This node contains
    at least one address family to be activated.";
  }
  description
  "Configuration specific to VRRP routing.";
}
description
"List of routing protocols used on
the site. This list can be augmented.";
```



```
    }
    description
    "Defines routing protocols.";
  }
  description
  "Grouping for routing protocols.";
}
grouping site-attachment-ip-connection {

  container ip-connection {
    container ipv4 {
      if-feature ipv4;
      leaf address-allocation-type {
        type identityref {
          base address-allocation-type;
        }
        must "not (derived-from-or-self(current(), 'l3vpn-ntw:slaac') or "+
            "derived-from-or-self(current(), '"+
            "'l3vpn-ntw:provider-dhcp-slaac'))" {
          error-message "SLAAC is only applicable to IPv6";
        }
        description
        "Defines how addresses are allocated.
        If there is no value for the address
        allocation type, then IPv4 is not enabled.";
      }
    }
    container provider-dhcp {
      when "derived-from-or-self(..../address-allocation-type, '"+
          "'l3vpn-ntw:provider-dhcp')" {
        description
        "Only applies when addresses are allocated by DHCP.";
      }
    }
    leaf provider-address {
      type inet:ipv4-address;
      description
      "Address of provider side.  If provider-address is not
      specified, then prefix length should not be specified
      either.  It also implies provider-dhcp allocation is
      not enabled.  If provider-address is specified, then
      the prefix length may or may not be specified.";
    }
    leaf prefix-length {
      type uint8 {
        range "0..32";
      }
      must "(../provider-address)" {
        error-message
        "If the prefix length is specified, provider-address
```

```
        must also be specified.";
        description
            "If the prefix length is specified, provider-address
            must also be specified.";
    }
    description
        "Subnet prefix length expressed in bits.
        If not specified, or specified as zero,
        this means the customer leaves the actual
        prefix length value to the provider.";
    }
    choice address-assign {
        default number;
        case number {
            leaf number-of-dynamic-address {
                type uint16;
                default 1;
                description
                    "Describes the number of IP addresses
                    the customer requires.";
            }
        }
        case explicit {
            container customer-addresses {
                list address-group {
                    key "group-id";
                    leaf group-id {
                        type string;
                        description
                            "Group-id for the address range from
                            start-address to end-address.";
                    }
                }
                leaf start-address {
                    type inet:ipv4-address;
                    description
                        "First address.";
                }
                leaf end-address {
                    type inet:ipv4-address;
                    description
                        "Last address.";
                }
            }
            description
                "Describes IP addresses allocated by DHCP.
                When only start-address or only end-address
                is present, it represents a single address.
                When both start-address and end-address are
                specified, it implies a range inclusive of both
```

```

        addresses.  If no address is specified, it implies
        customer addresses group is not supported.";
    }
    description
    "Container for customer addresses is allocated by DHCP.";
}
}
    description
    "Choice for the way to assign addresses.";
}
    description
    "DHCP allocated addresses related parameters.";
}
container dhcp-relay {
    when "derived-from-or-self(..address-allocation-type, "+
    "'l3vpn-ntw:provider-dhcp-relay')" {
        description
        "Only applies when provider is required to implement
        DHCP relay function.";
    }
}
leaf provider-address {
    type inet:ipv4-address;
    description
    "Address of provider side.  If provider-address is not
    specified, then prefix length should not be specified
    either.  It also implies provider-dhcp allocation is
    not enabled.  If provider-address is specified, then
    prefix length may or may not be specified.";
}
leaf prefix-length {
    type uint8 {
        range "0..32";
    }
}
must "(../provider-address)" {
    error-message
    "If prefix length is specified, provider-address
    must also be specified.";
    description
    "If prefix length is specified, provider-address
    must also be specified.";
}
}
    description
    "Subnet prefix length expressed in bits.  If not
    specified, or specified as zero, this means the
    customer leaves the actual prefix length value
    to the provider.";
}
container customer-dhcp-servers {

```

```
leaf-list server-ip-address {
  type inet:ipv4-address;
  description
    "IP address of customer DHCP server.";
}
description
  "Container for list of customer DHCP servers.";
}
description
  "DHCP relay provided by operator.";
}
container addresses {
  when "derived-from-or-self(..../address-allocation-type, '"+
    "'l3vpn-ntw:static-address')" {
    description
      "Only applies when protocol allocation type is static.";
  }
  leaf provider-address {
    type inet:ipv4-address;
    description
      "IPv4 Address List of the provider side.
      When the protocol allocation type is static,
      the provider address must be configured.";
  }
  leaf customer-address {
    type inet:ipv4-address;
    description
      "IPv4 Address of customer side.";
  }
  leaf prefix-length {
    type uint8 {
      range "0..32";
    }
    description
      "Subnet prefix length expressed in bits.
      It is applied to both provider-address
      and customer-address.";
  }
  description
    "Describes IPv4 addresses used.";
}
description
  "IPv4-specific parameters.";
}
container ipv6 {
  if-feature ipv6;
  leaf address-allocation-type {
    type identityref {
```

```
    base address-allocation-type;
  }
  description
    "Defines how addresses are allocated.
    If there is no value for the address
    allocation type, then IPv6 is
    not enabled.";
}

container provider-dhcp {
  when "derived-from-or-self(..address-allocation-type, "+"
    "'l3vpn-ntw:provider-dhcp'") "+"
    "or derived-from-or-self(..address-allocation-type, "+"
    "'l3vpn-ntw:provider-dhcp-slaac'")" {
    description
      "Only applies when addresses are allocated by DHCP.";
  }
  leaf provider-address {
    type inet:ipv6-address;
    description
      "Address of the provider side. If provider-address
      is not specified, then prefix length should not be
      specified either. It also implies provider-dhcp
      allocation is not enabled. If provider-address is
      specified, then prefix length may or may
      not be specified.";
  }
  leaf prefix-length {
    type uint8 {
      range "0..128";
    }
    must "(../provider-address)" {
      error-message
        "If prefix length is specified, provider-address
        must also be specified.";
      description
        "If prefix length is specified, provider-address
        must also be specified.";
    }
    description
      "Subnet prefix length expressed in bits. If not
      specified, or specified as zero, this means the
      customer leaves the actual prefix length value
      to the provider.";
  }
  choice address-assign {
    default number;
    case number {
```

```
    leaf number-of-dynamic-address {
      type uint16;
      default 1;
      description
        "Describes the number of IP addresses the customer
        requires.";
    }
  }
  case explicit {
    container customer-addresses {
      list address-group {
        key "group-id";
        leaf group-id {
          type string;
          description
            "Group-id for the address range from
            start-address to end-address.";
        }
        leaf start-address {
          type inet:ipv6-address;
          description
            "First address.";
        }
        leaf end-address {
          type inet:ipv6-address;
          description
            "Last address.";
        }
        description
          "Describes IP addresses allocated by DHCP. When only
          start-address or only end-address is present, it
          represents a single address. When both start-address
          and end-address are specified, it implies a range
          inclusive of both addresses. If no address is
          specified, it implies customer addresses group is
          not supported.";
      }
      description
        "Container for customer addresses allocated by DHCP.";
    }
  }
  description
    "Choice for the way to assign addresses.";
}
description
  "DHCP allocated addresses related parameters.";
}
container dhcp-relay {
```

```
when "derived-from-or-self(..address-allocation-type, "+
    "'l3vpn-ntw:provider-dhcp-relay')" {
  description
  "Only applies when the provider is required
  to implement DHCP relay function.";
}

leaf provider-address {
  type inet:ipv6-address;
  description
  "Address of the provider side. If provider-address is
  not specified, then prefix length should not be
  specified either. It also implies provider-dhcp
  allocation is not enabled. If provider address
  is specified, then prefix length may or may
  not be specified.";
}

leaf prefix-length {
  type uint8 {
    range "0..128";
  }
  must "(../provider-address)" {
    error-message
    "If prefix length is specified, provider-address
    must also be specified.";
    description
    "If prefix length is specified, provider-address
    must also be specified.";
  }
  description
  "Subnet prefix length expressed in bits. If not
  specified, or specified as zero, this means the
  customer leaves the actual prefix length value
  to the provider.";
}

container customer-dhcp-servers {
  leaf-list server-ip-address {
    type inet:ipv6-address;
    description
    "This node contains the IP address of
    the customer DHCP server. If the DHCP relay
    function is implemented by the
    provider, this node contains the
    configured value.";
  }
  description
  "Container for list of customer DHCP servers.";
}

description
```

```
"DHCP relay provided by operator.";
}
container addresses {
  when "derived-from-or-self(..address-allocation-type, "+"
    "'l3vpn-ntw:static-address')" {
    description
      "Only applies when protocol allocation type is static.";
  }
  leaf provider-address {
    type inet:ipv6-address;
    description
      "IPv6 Address of the provider side. When the protocol
      allocation type is static, the provider address
      must be configured.";
  }
  leaf customer-address {
    type inet:ipv6-address;
    description
      "The IPv6 Address of the customer side.";
  }
  leaf prefix-length {
    type uint8 {
      range "0..128";
    }
    description
      "Subnet prefix length expressed in bits.
      It is applied to both provider-address and
      customer-address.";
  }
  description
    "Describes IPv6 addresses used.";
  description
    "IPv6-specific parameters.";
}
container oam {
  container bfd {
    if-feature bfd;
    leaf enabled {
      type boolean;
      default false;
      description
        "If true, BFD activation is required.";
    }
  }
  choice holdtime {
    default fixed;
    case fixed {
      leaf fixed-value {
```



```
    type uint32;
    units msec;
    description
        "Expected BFD holdtime expressed in msec. The customer
        may impose some fixed values for the holdtime period
        if the provider allows the customer use this function.
        If the provider doesn't allow the customer to use this
        function, the fixed-value will not be set.";
    }
}
case profile {
    leaf profile-name {
        type leafref {
            path "/l3vpn-ntw/vpn-profiles/valid-provider-identifiers/"+
                "bfd-profile-identifier/id";
        }
        description
            "Well-known SP profile name. The provider can propose
            some profiles to the customer, depending on the service
            level the customer wants to achieve. Profile names
            must be communicated to the customer.";
    }
    description
        "Well-known SP profile.";
}
description
    "Choice for holdtime flavor.";
}
description
    "Container for BFD.";
}
description
    "Defines the Operations, Administration, and Maintenance (OAM)
    mechanisms used on the connection. BFD is set as a fault
    detection mechanism, but the 'oam' container can easily
    be augmented by other mechanisms";
}
description
    "Defines connection parameters.";
}
description
    "This grouping defines IP connection parameters.";
}
grouping site-service-multicast {
    container multicast {
        if-feature multicast;
        leaf multicast-site-type {
            type enumeration {
```

```
enum receiver-only {
  description
    "The site only has receivers.";
}
enum source-only {
  description
    "The site only has sources.";
}
enum source-receiver {
  description
    "The site has both sources and receivers.";
}
}
default source-receiver;
description
  "Type of multicast site.";
}
container multicast-address-family {
  leaf ipv4 {
    if-feature ipv4;
    type boolean;
    default false;
    description
      "Enables IPv4 multicast.";
  }
  leaf ipv6 {
    if-feature ipv6;
    type boolean;
    default false;
    description
      "Enables IPv6 multicast.";
  }
  description
    "Defines protocol to carry multicast.";
}
leaf protocol-type {
  type enumeration {
    enum host {
      description
        "Hosts are directly connected to the provider network.
        Host protocols such as IGMP or MLD are required.";
    }
    enum router {
      description
        "Hosts are behind a customer router.
        PIM will be implemented.";
    }
    enum both {
```

```
    description
    "Some hosts are behind a customer router, and
    some others are directly connected to the
    provider network. Both host and routing protocols
    must be used. Typically, IGMP and PIM will be
    implemented.";
  }
}
default "both";
description
"Multicast protocol type to be used with the customer site.";
}
description
"Multicast parameters for the site.";
}
description
"Multicast parameters for the site.";
}
grouping site-management {
  container management {
    leaf type {
      type identityref {
        base management;
      }
      mandatory true;
      description
      "Management type of the connection.";
    }
    description
    "Management configuration.";
  }
  description
  "Management parameters for the site.";
}
grouping site-devices {
  container devices {
    when "derived-from-or-self(..management/type, "+
    "'l3vpn-ntw:provider-managed') or "+
    "derived-from-or-self(..management/type, 'l3vpn-ntw:co-managed') " {
      description
      "Applicable only for provider-managed or
      co-managed device.";
    }
  }
  list device {
    key device-id;
    leaf device-id {
      type svc-id;
      description
```

```
    "Identifier for the device.";
  }
  leaf location {
    type leafref {
      path "../../../locations/"+
        "location/location-id";
    }
    mandatory true;
    description
      "Location of the device.";
  }
  container management {
    when "derived-from-or-self ../../../management/type, "+
      "'l3vpn-ntw:co-managed'" {
      description
        "Applicable only for co-managed device.";
    }
    leaf address-family {
      type address-family;
      description
        "Address family used for management.";
    }
    leaf address {
      when "(../address-family)" {
        description
          "If address-family is specified, then address should
            also be specified. If address-family is not specified,
            then address should also not be specified.";
      }
      type inet:ip-address;
      mandatory true;
      description
        "Management address.";
    }
    description
      "Management configuration. Applicable only for
        co-managed device.";
  }
  description
    "List of devices requested by customer.";
}
description
  "Device configuration.";
}
description
  "Grouping for device allocation.";
}
grouping site-vpn-flavor {
```

```
leaf site-vpn-flavor {
  type identityref {
    base site-vpn-flavor;
  }
  default site-vpn-flavor-single;
  description
    "Defines the way the VPN multiplexing is done, e.g., whether
    the site belongs to a single VPN site or a multiVPN; or, in the case
    of a multiVPN, whether the logical accesses of the sites belong
    to the same set of VPNs or each logical access maps to
    different VPNs.";
}
description
  "Grouping for site VPN flavor.";
}
grouping site-maximum-routes {
  container maximum-routes {
    list address-family {
      key af;
      leaf af {
        type address-family;
        description
          "Address family.";
      }
      leaf maximum-routes {
        type uint32;
        description
          "Maximum prefixes the VRF can accept
          for this address family.";
      }
      description
        "List of address families.";
    }
    description
      "Defines 'maximum-routes' for the VRF.";
  }
  description
    "Defines 'maximum-routes' for the site.";
}
grouping site-security {
  container security {
    uses site-security-authentication;
    uses site-security-encryption;
    description
      "Site-specific security parameters.";
  }
  description
    "Grouping for security parameters.";
```

```
}
grouping site-service {
  container service {
    uses site-service-qos-profile;
    uses site-service-mpls;
    uses site-service-multicast;
    description
      "Service parameters on the attachment.";
  }
  description
    "Grouping for service parameters.";
}
grouping site-network-access-service {
  container service {
    uses site-service-basic;
    /* Extension */
    /* uses svc-bandwidth-params; */
    /* EoExt */
    uses site-service-qos-profile;
    uses site-service-mpls;
    uses site-service-multicast;
    description
      "Service parameters on the attachment.";
  }
  description
    "Grouping for service parameters.";
}
grouping vpn-extranet {
  container extranet-vpns {
    if-feature extranet-vpn;
    list extranet-vpn {
      key vpn-id;
      leaf vpn-id {
        type svc-id;
        description
          "Identifies the target VPN the local VPN want to access.";
      }
      leaf local-sites-role {
        type identityref {
          base site-role;
        }
      }
      default any-to-any-role;
      description
        "This describes the role of the
        local sites in the target VPN topology.  In the any-to-any VPN
        service topology, the local sites must have the same role, which
        will be 'any-to-any-role'.  In the Hub-and-Spoke VPN service
        topology or the Hub-and-Spoke disjoint VPN service topology,
```

```
    the local sites must have a Hub role or a Spoke role.";
  }
  description
    "List of extranet VPNs or target VPNs the local VPN is
    attached to.";
  }
  description
    "Container for extranet VPN configuration.";
  }
  description
    "Grouping for extranet VPN configuration.
    This provides an easy way to interconnect
    all sites from two VPNs.";
  }
  grouping site-attachment-availability {
    container availability {
      leaf access-priority {
        type uint32;
        default 100;
        description
          "Defines the priority for the access.
          The higher the access-priority value,
          the higher the preference of the
          access will be.";
      }
      description
        "Availability parameters (used for multihoming).";
    }
    description
      "Defines availability parameters for a site.";
  }
  grouping vpn-profile-cfg {
    container valid-provider-identifiers {
      list cloud-identifier {
        if-feature cloud-access;
        key id;
        leaf id {
          type string;
          description
            "Identification of cloud service.
            Local administration meaning.";
        }
        description
          "List for Cloud Identifiers.";
      }
      list encryption-profile-identifier {
        key id;
        leaf id {
```

```
    type string;
    description
      "Identification of the SP encryption profile
      to be used.  Local administration meaning.";
  }
  description
    "List for encryption profile identifiers.";
}
list qos-profile-identifier {
  key id;
  leaf id {
    type string;
    description
      "Identification of the QoS Profile to be used.
      Local administration meaning.";
  }
  description
    "List for QoS Profile Identifiers.";
}
list bfd-profile-identifier {
  key id;
  leaf id {
    type string;
    description
      "Identification of the SP BFD Profile to be used.
      Local administration meaning.";
  }
  description
    "List for BFD Profile identifiers.";
}

list routing-profile-identifier {
  key id;
  leaf id {
    type string;
    description
      "Identification of the routing Profile to be used
      by the routing-protocols within sites and site-
      network-accesses. Local administration meaning.";
  }
  description
    "List for Routing Profile Identifiers.";
}

nacm:default-deny-write;
description
  "Container for Valid Provider Identifies.";
}
```



```
    description
      "Grouping for VPN Profile configuration.";
  }
  grouping vpn-svc-cfg {
    leaf vpn-id {
      type svc-id;
      description
        "VPN identifier. Local administration meaning.";
    }
    leaf customer-name {
      type string;
      description
        "Name of the customer that actually uses the VPN service.
        In the case that any intermediary (e.g., Tier-2 provider
        or partner) sells the VPN service to their end user
        on behalf of the original service provider (e.g., Tier-1
        provider), the original service provider may require the
        customer name to provide smooth activation/commissioning
        and operation for the service.";
    }
    leaf vpn-service-topology {
      type identityref {
        base vpn-topology;
      }
      default any-to-any;
      description
        "VPN service topology.";
    }

    leaf description {
      type string;
      description
        "Textual description of a VPN service.";
    }
  }

  uses ie-profiles-params;
  uses vpn-nodes-params;
  uses vpn-service-cloud-access;
  uses vpn-service-multicast;
  uses vpn-service-mpls;
  uses vpn-extranet;
  description
    "Grouping for VPN service configuration.";
}
grouping site-top-level-cfg {
  uses operational-requirements;
  uses customer-location-info;
  uses site-devices;
```

```
uses site-diversity;
uses site-management;
uses site-vpn-flavor;
uses site-maximum-routes;
uses site-security;
uses site-service;
uses site-protection;
uses site-routing;
description
"Grouping for site top-level configuration.";
}
grouping site-network-access-top-level-cfg {

  /* Extension */

  uses status-params;

  /* End of Extension */

  leaf site-network-access-type {
    type identityref {
      base site-network-access-type;
    }
    default point-to-point;
    description
    "Describes the type of connection, e.g.,
    point-to-point or multipoint.";
  }
  choice location-flavor {
    case location {
      when "derived-from-or-self ../../management/type, "+
        "'l3vpn-ntw:customer-managed'" {
        description
        "Applicable only for customer-managed device.";
      }
      leaf location-reference {
        type leafref {
          path "../../locations/location/location-id";
        }
        description
        "Location of the site-network-access.";
      }
    }
  }
  case device {
    when "derived-from-or-self ../../management/type, "+
      "'l3vpn-ntw:provider-managed' or "+
      "derived-from-or-self ../../management/type, "+
      "'l3vpn-ntw:co-managed'" {
```

```
    description
      "Applicable only for provider-managed or co-managed device.";
  }
  leaf device-reference {
    type leafref {
      path "../..../devices/device/device-id";
    }
    description
      "Identifier of CE to use.";
  }
}
mandatory true;
description
  "Choice of how to describe the site's location.";
}
uses access-diversity;
uses site-attachment-bearer;
uses site-attachment-ip-connection;
uses site-security;
uses site-network-access-service;
uses site-routing;
uses site-attachment-availability;
description
  "Grouping for site network access top-level configuration.";
}

/* Extensions */

/* Bearers in a site */
grouping site-bearer-params {

  container site-bearers {
    list bearer {
      key "bearer-id";

      leaf bearer-id {
        type string;
        description "";
      }

      leaf BearerType {
        type identityref {
          base bearer-inf-type;
        }
        description
          "Request for an Bearer access type."
      }
    }
  }
}
```

```
        Choose between port or lag connection type.";
    }

    leaf ne-id {
        type string;
        description
            "NE-id reference.";
    }

    leaf port-id {
        type string;
        description
            "Port-id in format slot/ card /port.";
    }

    leaf lag-id {
        type string;
        description
            "lag-id in format id.";
    }
    description
        "Parameters used to identify each bearer";
    }
    description
        "Grouping to reuse the site bearer assignment";
    }
    description
        "Grouping to reuse the site bearer assignment";
    }

/* UNUSED */
grouping svc-bandwidth-params {
    container svc-bandwidth {
        if-feature "input-bw";
        list bandwidth {
            key "direction type";
            leaf direction {
                type identityref {
                    base bw-direction;
                }
                description
                    "Indicates the bandwidth direction. It can be
                     the bandwidth download direction from the SP to
                     the site or the bandwidth upload direction from
                     the site to the SP.";
            }
            leaf type {
                type identityref {
```

```
        base bw-type;
    }
    description
        "Bandwidth type.  By default, the bandwidth type
        is set to 'bw-per-cos'.";
}
leaf cos-id {
    when "derived-from-or-self(..../type, "
        + "'l3vpn-ntw:bw-per-cos')" {
        description
            "Relevant when the bandwidth type is set to
            'bw-per-cos'.";
    }
    type uint8;
    description
        "Identifier of the CoS, indicated by DSCP or a
        CE-VLAN CoS (802.1p) value in the service frame.
        If the bandwidth type is set to 'bw-per-cos',
        the CoS ID MUST also be specified.";
}
leaf vpn-id {
    when "derived-from-or-self(..../type, "
        + "'l3vpn-ntw:bw-per-svc')" {
        description
            "Relevant when the bandwidth type is
            set as bandwidth per VPN service.";
    }
    type svc-id;
    description
        "Identifies the target VPN.  If the bandwidth
        type is set as bandwidth per VPN service, the
        vpn-id MUST be specified.";
}
leaf cir {
    type uint64;
    units "bps";
    mandatory true;
    description
        "Committed Information Rate.  The maximum number
        of bits that a port can receive or send over
        an interface in one second.";
}
leaf cbs {
    type uint64;
    units "bps";
    mandatory true;
    description
        "Committed Burst Size (CBS).  Controls the bursty
```

```
        nature of the traffic. Traffic that does not
        use the configured Committed Information Rate
        (CIR) accumulates credits until the credits
        reach the configured CBS.";
    }
    leaf eir {
        type uint64;
        units "bps";
        description
            "Excess Information Rate (EIR), i.e., excess frame
            delivery allowed that is not subject to an SLA.
            The traffic rate can be limited by the EIR.";
    }
    leaf ebs {
        type uint64;
        units "bps";
        description
            "Excess Burst Size (EBS). The bandwidth available
            for burst traffic from the EBS is subject to the
            amount of bandwidth that is accumulated during
            periods when traffic allocated by the EIR
            policy is not used.";
    }
    leaf pir {
        type uint64;
        units "bps";
        description
            "Peak Information Rate, i.e., maximum frame
            delivery allowed. It is equal to or less
            than the sum of the CIR and the EIR.";
    }
    leaf pbs {
        type uint64;
        units "bps";
        description
            "Peak Burst Size. It is measured in bytes per
            second.";
    }
    description
        "List of bandwidth values (e.g., per CoS,
        per vpn-id).";
}
description
    "From the customer site's perspective, the service
    input/output bandwidth of the connection or
    download/upload bandwidth from the SP/site
    to the site/SP.";
```

```
        description
            " ";
    }

    grouping status-params {
        container status {
            leaf admin-enabled {
                type boolean;
                description
                    "Administrative Status UP/DOWN";
            }
            leaf oper-status {
                type operational-type;
                config false;
                description
                    "Operations status";
            }
            description "";
        }
        description
            "Grouping used to join operational and administrative status
            is re used in the Site Network Access and in the VPN-Node";
    }

    /* Parameters related to vpn-nodes (VRF config.) */
    grouping vpn-nodes-params {
        container vpn-nodes {
            description "";

            list vpn-node {
                key "vpn-node-id ne-id";

                leaf vpn-node-id {
                    type string;
                    description "";
                }

                leaf description {
                    type string;
                    description
                        "Textual description of a VPN node.";
                }

                leaf ne-id {
                    type string;
                    description "";
                }
            }
        }
    }
}
```

```
    }

    leaf router-id {
      type inet:ip-address;
      description
        "router-id information can be ipv4/6 addresses";
    }

    leaf address-family {
      type address-family;
      description
        "Address family used for router-id information.";
    }

    leaf node-role {
      type identityref {
        base site-role;
      }
      default any-to-any-role;
      description
        "Role of the vpn-node in the IP VPN.";
    }
    uses rt-rd;
    uses status-params;

    /* Here we use the name given to the existing structure in sites */
    uses site-maximum-routes;

    leaf node-ie-profile {
      type leafref {
        path "/l3vpn-ntw/vpn-services/"+
          "vpn-service/ie-profiles/ie-profile/ie-profile-id";
      }
      description "";
    }
    description "";
  }
  description "Grouping to define VRF-specific configuration.";
}

/* Parameters related to import and export profiles (RTs RDs.) */
grouping ie-profiles-params {
  container ie-profiles {
    list ie-profile {
      key "ie-profile-id";
      leaf ie-profile-id {
        type string;
      }
    }
  }
}
```



```
        description
            "";
    }
    uses rt-rd;
    description
    "";
    }
    description
    "";
    }
    description
    "Grouping to specify rules for route import and export";
}

grouping pseudowire-params {
    container pseudowire {
        /*leaf far-end {*/
        /*  description "IP of the remote peer of the pseudowire.";*/
        /*  type inet:ip-address;*/
        /*}*/
        leaf vcid {
            type uint32;
            description
            "PW or VC identifier.";
        }
        description
        "Pseudowire termination parameters";
    }
    description
    "Grouping pseudowire termination parameters";
}

grouping security-params {
    container security {
        leaf auth-key {
            type string;
            description
            "MD5 authentication password for the connection towards the
            customer edge.";
        }
        description
        "Container for aggregating any security parameter for routing
        sessions between a PE and a CE.";
    }
    description
    "Grouping to define security parameters";
}
```

```
grouping ethernet-params {
  container connection {
    leaf encapsulation-type {
      type identityref {
        base encapsulation-type;
      }
      default "untagged-int";
      description
        "Encapsulation type. By default, the
         encapsulation type is set to 'untagged'.";
    }
    container tagged-interface {
      leaf type {
        type identityref {
          base tagged-inf-type;
        }
        default "priority-tagged";
        description
          "Tagged interface type. By default,
           the type of the tagged interface is
           'priority-tagged'.";
      }
      container dot1q-vlan-tagged {
        when "derived-from-or-self(..../type, "
          + "'l3vpn-ntw:dot1q') " {
          description
            "Only applies when the type of the tagged
             interface is 'dot1q'.";
        }
        if-feature "dot1q";
        leaf tag-type {
          type identityref {
            base tag-type;
          }
          default "c-vlan";
          description
            "Tag type. By default, the tag type is
             'c-vlan'.";
        }
        leaf cvlan-id {
          type uint16;
          description
            "VLAN identifier.";
        }
        description
          "Tagged interface.";
      }
    }
    container priority-tagged {
```

```
when "derived-from-or-self(..type, "
  + "'l3vpn-ntw:priority-tagged')" {
  description
    "Only applies when the type of the tagged
    interface is 'priority-tagged'.";
}
leaf tag-type {
  type identityref {
    base tag-type;
  }
  default "c-vlan";
  description
    "Tag type. By default, the tag type is
    'c-vlan'.";
}
description
  "Priority tagged.";
}
container qinq {
  when "derived-from-or-self(..type, "
    + "'l3vpn-ntw:qinq')" {
    description
      "Only applies when the type of the tagged
      interface is 'qinq'.";
  }
  if-feature "qinq";
  leaf tag-type {
    type identityref {
      base tag-type;
    }
    default "c-s-vlan";
    description
      "Tag type. By default, the tag type is
      'c-s-vlan'.";
  }
  leaf svlan-id {
    type uint16;
    mandatory true;
    description
      "SVLAN identifier.";
  }
  leaf cvlan-id {
    type uint16;
    mandatory true;
    description
      "CVLAN identifier.";
  }
  description
```

```
    "QinQ.";
}
container qinany {
    when "derived-from-or-self(..type, "
        + "'l3vpn-ntw:qinany')" {
        description
            "Only applies when the type of the tagged
            interface is 'qinany'.";
    }
    if-feature "qinany";
    leaf tag-type {
        type identityref {
            base tag-type;
        }
        default "s-vlan";
        description
            "Tag type. By default, the tag type is
            's-vlan'.";
    }
    leaf svlan-id {
        type uint16;
        mandatory true;
        description
            "Service VLAN ID.";
    }
    description
        "Container for QinAny.";
}
container vxlan {
    when "derived-from-or-self(..type, "
        + "'l3vpn-ntw:vxlan')" {
        description
            "Only applies when the type of the tagged
            interface is 'vxlan'.";
    }
    if-feature "vxlan";
    leaf vni-id {
        type uint32;
        mandatory true;
        description
            "VXLAN Network Identifier (VNI).";
    }
    leaf peer-mode {
        type identityref {
            base vxlan-peer-mode;
        }
        default "static-mode";
        description
```

```
        "Specifies the VXLAN access mode. By default,
        the peer mode is set to 'static-mode'.";
    }
    list peer-list {
        key "peer-ip";
        leaf peer-ip {
            type inet:ip-address;
            description
                "Peer IP.";
        }
        description
            "List of peer IP addresses.";
    }
    description
        "QinQ.";
}
description
    "Container for tagged interfaces.";
}
description
    "Encapsulation types";
}
description
    "Grouping to define encapsulation types";
}

grouping rt-rd {
    leaf rd {
        type rt-types:route-distinguisher;
        description
            "";
    }
    container vpn-targets {
        description
            "Set of route-targets to match for import and export routes
            to/from VRF";
        uses rt-types:vpn-route-targets;
    }
    description
        "";
}

/* Main blocks */
container l3vpn-ntw {
    container vpn-profiles {
        uses vpn-profile-cfg;
        description
            "Container for VPN Profiles.";
    }
}
```

```

    }
    container vpn-services {
        list vpn-service {
            key vpn-id;
            uses vpn-svc-cfg;
            description
                "List of VPN services.";
        }
        description
            "Top-level container for the VPN services.";
    }
    container sites {
        list site {
            key site-id;
            leaf site-id {
                type svc-id;
                description
                    "Identifier of the site.";
            }
            leaf description {
                type string;
                description
                    "Textual description of a site.";
            }
        }
        uses site-top-level-cfg;
        uses operational-requirements-ops;
        uses site-bearer-params;
        container site-network-accesses {
            list site-network-access {
                key site-network-access-id;
                leaf site-network-access-id {
                    type svc-id;
                    description
                        "Identifier for the access.";
                }
                leaf description {
                    type string;
                    description
                        "Textual description of a VPN service.";
                }
            }
            uses site-network-access-top-level-cfg;
            leaf node-id {
                type leafref{
                    path "/l3vpn-ntw/vpn-services/vpn-service/vpn-nodes/vpn-node/vpn-node-
id";
                }
                description
                    "Reference the VPN node id";
            }
        }
    }

```

```
leaf service-id {
  type leafref{
    path "/l3vpn-ntw/vpn-services/vpn-service/vpn-id";
  }
  description
    "Reference the VPN node id";
}
leaf access-group-id {
  type yang:uuid;
  description
    "Reference the Access Goup ID.
    It is used to group and identify SNA with common behavior
    such as dual-homming";
}
description
  "List of accesses for a site.";
}
description
  "List of accesses for a site.";
}
description
  "List of sites.";
}
description
  "Container for sites.";
}
description
  "Main container for L3VPN service configuration.";
}
}
```

Figure 4

6. IANA CONSIDERATIONS

This memo includes no request to IANA.

7. SECURITY CONSIDERATIONS

All the security considerations of [RFC8299] apply to this document. Subsequent versions will provide additional security considerations.

8. IMPLEMENTATION STATUS

This section will be used to track the status of the implementations of the model. It is aimed at being removed if the document becomes RFC.

9. ACKNOWLEDGEMENTS

Thanks to Adrian Farrel and Miguel Cros for the suggestions on the document. Thanks to Stephane Litowski and Philip Eardlay for the review. Lots of thanks for the discussions on opsawg mailing list and at IETF meeting. Some of the comments have already been incorporated and the other part of the comments will be addressed in the next versions.

This work was supported in part by the European Commission funded H2020-ICT-2016-2 METRO-HAUL project (G.A. 761727).

10. CONTRIBUTORS

Daniel King
Old Dog Consulting
Email: daniel@olddog.co.uk

Samier Barguil
Telefonica
Email: samier.barguilgiraldo.ext@telefonica.com

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

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

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

[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>>.

- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.
- [RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.

Authors' Addresses

Alejandro Aguado
Nokia
Madrid
ES

Email: alejandro.aguado_martin@nokia.com

Oscar Gonzalez de Dios (editor)
Telefonica
Madrid
ES

Email: oscar.gonzalezdedios@telefonica.com

Victor Lopez
Telefonica
Madrid
ES

Email: victor.lopezalvarez@telefonica.com

Daniel Voyer
Bell Canada
CA

Email: daniel.voyer@bell.ca

Luis Angel Munoz
Vodafone
ES

Email: luis-angel.munoz@vodafone.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2020

E. Lear
Cisco Systems
July 01, 2019

Controller Identification Extension for MUD
draft-lear-opsawg-mud-controller-candidates-00

Abstract

Manufacturer Usage Descriptions (MUD) are a means by which devices can establish expectations about how they are intended to behave, and how the network should treat them. This extension provides a means for a MUD controller to identify itself through its own MUD file.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 2, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. The ietf-mud-controller model extension	3
2.1. The controller-candidate augmentation to the MUD YANG model	3
3. Examples	5
4. Security Considerations	5
5. Normative References	6
Appendix A. Changes from Earlier Versions	6
Author's Address	6

1. Introduction

Manufacturer Usage Descriptions (MUD) [RFC8520] provides a means for devices to identify what they are and what sort of network access they need. Two abstractions made available in that specification are "controller" and "my-controller". As initially specified, these devices must be identified by the network administrator. To simplify the administrator's life, a means of identifying devices that are candidates to be a controller for another device is desirable.

This memo specifies an extension that makes use of the MUD file of the controller itself. It also sets the groundwork to create a RESTful interface for applications that are serving as controllers to make use of the same grouping. However, that work is left either for a future version of this draft, or a future specification.

For example, a light switch might identify as a candidate controller for luminaires. A thermostat might identify as a controller for an air conditioner or heater.

To address the case where "my-controller" is used, the manufacturer of a candidate controller must list the MUD URLs of devices that it knows it can serve. For example, if thermostat manufacturer "Example Thermostat, Inc" knows that it can properly control a heater made by "Example Heating, Inc", and the heater has a mud url of "https://heating.example.com/mudurls/heater1000.json", the thermostat would list that or an expression matching that URL in its MUD URL. The heater's MUD file would be expected to contain a "my-controller" statement in order for this controller to be considered a candidate.

To address the case where "controller" is used, then the class indicated by the controller must be named instead of a MUD URL. In our previous example, let us assume that the heater1000.json file contains a "controller" statement with a class URI of "https://heating.example.com/theromstats". In this case, the

controller would contain a statement that indicates it can be a controller for class "https://heating.example.com".

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. The ietf-mud-controller model extension

We now formally define this extension. This is done in two parts. First, the extension name "controller-candidate" is listed in the "extensions" array of the MUD file.

Second, the "mud" container is augmented to contain two optional lists, one that contains mud-urls that can be matched for "my-controller", and a second list that contains classes that can be matched against "controller".

This is done as follows.

```
module: ietf-mud-controller-candidate
  augment /mud:mud:
    +--rw controller-candidates
      +--rw urls*      inet:uri
      +--rw classes*   inet:uri
```

2.1. The controller-candidate augmentation to the MUD YANG model

```
<CODE BEGINS>file "ietf-mud-controller-candidate@2019-06-20.yang"
module ietf-mud-controller-candidate {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mud-controller-candidate";
  prefix mud-cc;

  import ietf-mud {
    prefix "mud";
  }

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

  organization
    "IETF OPSAWG (Ops Area) Working Group";
  contact
```

```
"WG Web: http://tools.ietf.org/wg/opsawg/
WG List: opsawg@ietf.org
Author: Eliot Lear
       lear@cisco.com
";
description

"This YANG module augments the ietf-mud model to provide for two
optional lists to indicate that this device type may be used as
a controller for other MUD-enabled devices.

Copyright (c) 2019 IETF Trust and the persons identified as
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and subject to
the license terms contained in, the Simplified BSD License set
forth in Section 4.c of the IETF Trust's Legal Provisions
Relating to IETF Documents
(https://trustee.ietf.org/license-info).

This version of this YANG module is part of RFC XXXX
(https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
for full legal notices.

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
'MAY', and 'OPTIONAL' in this document are to be interpreted as
described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
they appear in all capitals, as shown here.
";

revision 2019-06-20 {
  description
    "Initial proposed standard.";
  reference "RFC XXXX: QoS for MUD Specification";
}

grouping mud-controller-candidates {
  description
    "Controller candidate grouping";
  container controller-candidates {
    description
      "Lists of controller candidates.";
    leaf-list urls {
      type inet:uri;
      description
        "a list of mud urls this device is designed to control.
```

```
    Each entry may end with a * to indicate a wildcard that
    matches zero or more of characters from that point.  A
    wildcard MUST NOT appear in the authority section of the
    URL.";
  }
  leaf-list classes {
    type inet:uri;
    description
      "A list of URIs that are used as classes in MUD files,
      indicating that this device can serve as a controller
      in those classes.";
  }
}

augment "/mud:mud" {
  uses mud-controller-candidates;
  description
    "add controller candidate list";
}
}
<CODE ENDS>
```

3. Examples

TBD

4. Security Considerations

All security considerations of [RFC8520] apply equally to this extension. In addition, some care should be given to claims that a device is permitted to be a controller in any given circumstances. Complete automation requires far more context than is currently specified here. Some form of confirmation or selection is required by an administrator. This memo simply makes it easier for administrator to identify candidates for controller selection.

IANA Considerations =====

The IANA is requested to add "controller-candidate" to the MUD extensions registry as follows:

Extension Name: controller-candidate
Standard reference: This document

5. 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>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [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>>.
- [RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <<https://www.rfc-editor.org/info/rfc8520>>.

Appendix A. Changes from Earlier Versions

Draft -00:

- o Initial revision

Author's Address

Eliot Lear
Cisco Systems
Richtistrasse 7
Wallisellen CH-8304
Switzerland

Phone: +41 44 878 9200
Email: lear@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 6, 2020

E. Lear
Cisco Systems
M. Ranganathan
NIST
July 05, 2019

Reporting MUD behavior to vendors
draft-lear-opsawg-mud-reporter-00

Abstract

As with other technology, manufacturers would like to understand how networks implementing MUD are treating devices that are providing MUD URLs and MUD files. This memo specifies an extension to MUD that permits certain behaviors to be reported.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 6, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. The mud-reporter-extension model extension	3
2.1. The mud-reporter-extension augmentation to the MUD YANG model	4
2.2. The Reporter record format	6
3. RESTful interface at the collector	11
4. Examples	12
5. Privacy Considerations	12
6. Security Considerations	13
7. References	13
7.1. Normative References	13
7.2. Informative References	13
Appendix A. Changes from Earlier Versions	14
Authors' Addresses	14

1. Introduction

Manufacturer Usage Descriptions (MUD) [RFC8520] provides a means for devices to identify what they are and what sort of network access they need. When a device with a MUD URL and a MUD file is fielded in volume, manufacturers may be curious as to whether it is getting the access it needs. There are a few several reasons why a device would not be getting the access it needs. Some examples include:

- o The MUD file permits access only to a controller but there is none.
- o The MUD file permits access only to same-manufacturer or model but there is none.
- o The MUD file permits access to a particular Internet service, but the name of that service has not been resolved (or name resolution failed).
- o The administrator overrode the recommendations in the MUD file.

This memo sets out to provide manufacturers indications regarding what has happened, in a similar vein to how DMARC is used to report message drops to message senders [RFC7489].

In order to provide meaningful reporting, it is necessary to indicate whether or not the above abstractions are in use at a given time, and any public IP addresses that have been mapped to domain names by the local deployment. A communication method that may establish the source of the reporter is also necessary, as well as the MUD URL in use at the time of the report.

This memo specifies a YANG model for reporting and a means for transmitting the report, and appropriate extensions to the MUD file to indicate how to report and how often.

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. The mud-reporter-extension model extension

We now formally define this extension. This is done in two parts. First, the extension name "reporter" is listed in the "extensions" array of the MUD file.

Second, the "mud" container is augmented with a container that points to where to report and how often.

This is done as follows:

```
module: ietf-mud-controller-candidate
  augment /mud:mud:
    +--rw reporter
      +--rw report-uri    inet:uri
      +--rw frequency?    uint32
```

Finally the logging format is defined as follows:

```

module: ietf-mud-reporter
+--rw mud-reporter
  +--rw mudurl?          inet:uri
  +--rw mud-report* [time]
    +--rw time              yang:timestamp
    +--rw opaqueidentifier? string
    +--rw direction?       enumeration
    +--rw mycontrollers?    uint32
    +--rw controllers* [uri]
      +--rw uri              inet:uri
      +--rw count?          uint32
      +--rw ipaddress?      inet:ip-address
    +--rw samemanufacturers? uint32
    +--rw manufacturers* [authority]
      +--rw authority        inet:host
      +--rw count?          uint32
      +--rw ipaddress?      inet:ip-address
    +--rw models* [uri]
      +--rw uri              inet:uri
      +--rw count?          uint32
      +--rw ipaddress?      inet:ip-address
    +--rw domains* [hostname]
      +--rw hostname         inet:host
      +--rw ip-addresses*    inet:ip-address
    +--rw manufacturer?     string
    +--rw model?            string
    +--rw local-networks?   boolean
    +--rw controller?       string
    +--rw drop-count?       uint32

```

2.1. The mud-reporter-extension augmentation to the MUD YANG model

```

<CODE BEGINS>file "ietf-mud-reporter-extension@2019-06-21.yang"
module ietf-mud-reporter-extension {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mud-reporter-extension";
  prefix mud-reporter-extension;

  import ietf-mud {
    prefix "mud";
  }

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

  organization

```

```
"IETF OPSAWG (Ops Area) Working Group";
contact
  "WG Web: http://tools.ietf.org/wg/opsawg/
  WG List: opsawg@ietf.org
  Author: Eliot Lear
  lear@cisco.com
  ";
description

  "This YANG module augments the ietf-mud model to provide for two
  optional lists to indicate that this device type may be used as
  a controller for other MUD-enabled devices.

  Copyright (c) 2019 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject to
  the license terms contained in, the Simplified BSD License set
  forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX
  (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
  for full legal notices.

  The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
  NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
  'MAY', and 'OPTIONAL' in this document are to be interpreted as
  described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.
  ";

revision 2019-06-21 {
  description
    "Initial proposed standard.";
  reference "RFC XXXX: Extension for MUD Reporting";
}

grouping mud-reporter-extension {
  description
    "Reporter information grouping";
  container reporter {
    description "Reporter information";
    leaf report-uri {
      type inet:uri;
      description
```

```
        "Restful endpoint for reporter information.";
    }
    leaf frequency {
        type uint32
        {
            range "60..max";
        }
        default 1440;
        description
            "The minimum period of time in minutes that a deployment
            should report.";
    }
}

augment "/mud:mud" {
    uses mud-reporter-extension;
    description
        "add reporter extension";
}
}
<CODE ENDS>
```

2.2. The Reporter record format

```
<CODE BEGINS>file "ietf-mud-reporter@2019-06-21.yang"
module ietf-mud-reporter {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-mud-reporter";
    prefix mud-reporter;

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

    organization
        "IETF OPSAWG (Ops Area) Working Group";
    contact
        "WG Web: http://tools.ietf.org/wg/opsawg/
        WG List: opsawg@ietf.org
        Author: Eliot Lear
        lear@cisco.com
        ";
    description
        "This YANG module specifies the reporting format for MUD managers
```

to use when they are reporting to manufacturers.

Copyright (c) 2019 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX (<https://www.rfc-editor.org/info/rfcXXXX>); see the RFC itself for full legal notices.

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document are to be interpreted as described in BCP 14 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.
";

```
revision 2019-06-21 {
  description
    "Initial proposed standard.";
  reference
    "RFC XXXX: Extension for MUD Reporting";
}

container mud-reporter {
  uses mud-reporter-grouping;
  description "Reporter Information.";
}

grouping mud-reporter-grouping {

  description
    "MUD reporter container.";
  leaf mudurl {
    type inet:uri;
    description
      "The MUD-URL for which the report is being sent.";
  }
  list mud-report {
    key "time";
    description
      "individual records.";
  }
}
```

```
leaf time {
  type yang:timestamp;
  description
    "when this happened.";
}
leaf opaqueidentifier {
  type string;
  description
    "This is an identifier that maps to a particular
    device. Its value MUST NOT be mappable back to
    any identifying information about the device. It
    may be a suitable hash, such as SHA256.";
}
leaf direction {
  type enumeration {
    enum to-device {
      description
        "packet was traveling toward the device";
    }
    enum from-device {
      description
        "packet was traveling away from the device";
    }
  }
  description
    "which way packet is going";
}
leaf mycontrollers {
  type uint32;
  description
    "how many entries for my-controller.";
}
list controllers {
  key "uri";
  description
    "list of controllers and how many there were.";
  leaf uri {
    type inet:uri;
    description
      "the class URI of this controller";
  }
  leaf count {
    type uint32;
    description
      "number of devices serving this class.";
  }
  leaf ipaddress {
    type inet:ip-address;
```



```
        description
            "IP address of the controller.  Note that the MUD
            reporter MUST NOT transmit this contents of this
            node to the manufacturer.";
    }
}
leaf samemanufacturers {
    type uint32;
    description
        "number of devices matching same
        manufacturer.";
}
list manufacturers {
    key "authority";
    description
        "list of models and how many there were.";
    leaf authority {
        type inet:host;
        description
            "the manufacturer domain";
    }
    leaf count {
        type uint32;
        description
            "number of devices serving this class.";
    }
    leaf ipaddress {
        type inet:ip-address;
        description
            "IP address of the controller.  Note that the MUD
            reporter MUST NOT transmit this contents of this
            node to the manufacturer.";
    }
}
list models {
    key "uri";
    description
        "list of models and how many there were.";
    leaf uri {
        type inet:uri;
        description
            "the URI of this model";
    }
    leaf count {
        type uint32;
        description
            "number of devices serving this class.";
    }
}
```

```
    leaf ipaddress {
      type inet:ip-address;
      description
        "IP address of the controller. Note that the MUD
        reporter MUST NOT transmit this contents of this
        node to the manufacturer.";
    }
  }
  list domains {
    key "hostname";
    description
      "list of hosts, and ip addresses if known.";
    leaf hostname {
      type inet:host;
      description
        "the host listed";
    }
    leaf-list ip-addresses {
      type inet:ip-address;
      description
        "ipv4 or v6 address mapping for this host if
        known.";
    }
  }
  uses class-drop-count;
}

grouping class-drop-count {
  description
    "Destination fields of acl violating packet are classfied.";
  leaf manufacturer {
    type string;
    description
      "manufacturer name";
  }
  leaf model {
    type string;
    description
      "model name";
  }
  leaf local-networks {
    type boolean;
    description
      "this packet matches the local networks
      classification";
  }
  leaf controller {
```

```
        type string;
        description
            "controller name";
    }
    leaf drop-count {
        type uint32;
        description
            "number of packets dropped for this classification";
    }
}
}
```

<CODE ENDS>

3. RESTful interface at the collector

```
<CODE BEGINS>file "ietf-mud-reporter-collector@2019-06-21.yang"
module iETF-mud-reporter-collector {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-mud-reporter-collector";
    prefix "mud-collector";

    import iETF-mud-reporter {
        prefix "reporter";
    }
    organization
        "IETF OPSAWG (Ops Area) Working Group";
    contact
        "WG Web: http://tools.ietf.org/wg/opsawg/
        WG List: opsawg@ietf.org
        Author: Eliot Lear
        lear@cisco.com
        Author: Mudumbai Ranganathan
        mranga@nist.gov
        ";
    description
        "This YANG module specifies the reporting format for MUD managers
        to use when they are reporting to manufacturers."
```

Copyright (c) 2019 IETF Trust and the persons identified as
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and subject to
the license terms contained in, the Simplified BSD License set
forth in Section 4.c of the IETF Trust's Legal Provisions
Relating to IETF Documents
(<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX (<https://www.rfc-editor.org/info/rfcXXXX>); see the RFC itself for full legal notices.

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document are to be interpreted as described in BCP 14 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.

```
";
revision 2019-06-21 {
  description
    "Initial proposed standard.";
  reference
    "RFC XXXX: Extension for MUD Reporting";
}
rpc post-mud-report {
  description
    "Rpc interface that must be supported by collection point.";
  input {
    container mud-report {
      uses reporter:mud-reporter-grouping;
      description "MUD report";
    }
  }
}
}

<CODE ENDS>
```

4. Examples

TBD

5. Privacy Considerations

Using this reporting mechanisms does not reveal internal IP addresses. Instead, it simply indicates whether a given abstraction is in use, and how many instances there are. What is revealed to the manufacturer is that one or more devices reporting a particular MUD-URL is located at a particular deployment. In addition, as of this draft, reportable events include only administratively dropped packets, and the times they were dropped.

In order to report the sorts of errors discussed in this memo, a deployment must determine which packets from a given device have either been or would be dropped due to an administrative filter rule.

6. Security Considerations

All security considerations of [RFC8520] apply equally to this extension. In addition, some care should be given to claims that a device is permitted to be a controller in any given circumstances. Complete automation requires far more context than is currently specified here. Some form of confirmation or selection is required by an administrator. This memo simply makes it easier for administrator to identify candidates for controller selection.

IANA Considerations =====

The IANA is requested to add "controller-candidate" to the MUD extensions registry as follows:

Extension Name: reporter
Standard reference: This document

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [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>>.
- [RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <<https://www.rfc-editor.org/info/rfc8520>>.

7.2. Informative References

- [RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <<https://www.rfc-editor.org/info/rfc7489>>.

Appendix A. Changes from Earlier Versions

Draft -00:

- o Initial revision

Authors' Addresses

Eliot Lear
Cisco Systems
Richtistrasse 7
Wallisellen CH-8304
Switzerland

Phone: +41 44 878 9200
Email: lear@cisco.com

Mudumbai Ranganathan
NIST
100 Bureau Dr.
Gaithersburg
U.S.A

Phone: +1 301 975 2857
Email: mranga@nist.gov

OPSWG WG
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

T. Reddy
McAfee
D. Wing
Citrix
July 8, 2019

MUD (D)TLS profiles for IoT devices
draft-reddy-opswg-mud-tls-00

Abstract

This memo extends Manufacturer Usage Description (MUD) to model DTLS and TLS usage. This allows a network element to notice abnormal DTLS or TLS usage which has been strong indicator of other software running on the endpoint, typically malware.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	4
3. Overview of MUD (D)TLS profiles for IoT devices	4
4. (D)TLS profile YANG module	5
4.1. Tree Structure	5
4.2. YANG Module	6
5. (D)TLS 1.3 handshake	10
5.1. Encrypted SNI	10
5.2. Full (D)TLS 1.3 handshake inspection	11
6. MUD File Example	12
7. Security Considerations	13
8. IANA Considerations	13
9. Acknowledgments	13
10. References	14
10.1. Normative References	14
10.2. Informative References	15
Authors' Addresses	16

1. Introduction

Encryption is necessary to protect the privacy of end users using IoT devices. In a network setting, TLS [RFC8446] and DTLS [I-D.ietf-tls-dtls13] are the dominant protocols to provide encryption for IoT device traffic. Unfortunately in conjunction with IoT applications rise of encryption, malware is also using encryption which thwarts network-based analysis such as deep packet inspection (DPI). Other mechanisms are needed to notice malware is running on the IoT device.

Malware frequently uses its own libraries for its activities, and those libraries are re-used much like any other software engineering project. Research [malware] indicates there are observable differences in how malware uses encryption compared with non-malware uses encryption. There are several interesting findings specific to DTLS and TLS which were found common to malware:

- o Older and weaker cryptographic parameters (e.g., TLS_RSA_WITH_RC4_128_SHA).
- o TLS SNI and server certificates are composed of subjects with characteristics of a domain generation algorithm (DGA) (e.g., www.33mhwt2j.net).
- o Higher use of self-signed certificates compared with typical legitimate software.

- o Discrepancies in the server name indication (SNI) TLS extension in the ClientHello message and the DNS names in the SubjectAltName(SAN) X.509 extension in the server certificate message.
- o Discrepancies in the key exchange algorithm and the client public key length in comparison with legitimate flows. As a reminder, Client Key Exchange message has been removed from TLS 1.3.
- o Lower diversity in TLS client advertised TLS extensions compared to legitimate clients.

If observable (D)TLS profile parameters are used, the following discusses the favorable impact on network security:

- o Although IoT devices that have a single or small number of uses might have very broad communication patterns. In such a case, MUD rules using ACLs on its own is not suitable for these IoT devices but observable (D)TLS profile parameters can be used for such IoT devices to permit intended use and to block malicious behaviour of IoT devices.
- o Several TLS deployments have been vulnerable to active Man-In-The-Middle (MITM) attacks because of lack of certificate validation. By observing (D)TLS profile parameters, a network element can detect when the TLS SNI mismatches the SubjectAltName and detect when the server's certificate is invalid, and alert those situations.
- o IoT device can learn a new skill, and the new skill changes the way the IoT device communicates with other devices located in the local network and Internet. In other words, if IP addresses and domain names the IoT device connects to rapidly changes and MUD rules using ACLs cannot be rapidly updated, observable (D)TLS profile parameters can be used to permit intended use and to block malicious behaviour of IoT device.

This document extends MUD [RFC8520] to model observable (D)TLS profile parameters. Using these (D)TLS profile parameters, an active MUD-enforcing firewall can identify MUD non-compliant DTLS and TLS behavior that can indicate malware is running on the IoT device. This detection can prevent malware download, block access to malicious domains, enforce use of strong ciphers, stop data exfiltration, etc. In addition, organizations may have policies around acceptable ciphers and certificates on the websites the IoT devices connect to. Examples include no use of old and less secure versions of TLS, no use of self-signed certificates, deny-list or accept-list of Certificate Authorities, valid certificate expiration

time, etc. These policies can be enforced by observing the (D)TLS profile parameters. Enterprise firewall can use the IoT device's (D)TLS profile parameters to identify legitimate flows by observation of (D)TLS sessions, and can make inferences to permit legitimate flows and to block malicious flows. The proposed technique is also suitable in deployments where decryption techniques are not ideal due to privacy concerns, non-cooperating end-points and expense.

2. 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.

"(D)TLS" is used for statements that apply to both Transport Layer Security [RFC8446] and Datagram Transport Layer Security [RFC6347]. Specific terms are used for any statement that applies to either protocol alone.

3. Overview of MUD (D)TLS profiles for IoT devices

In Enterprise networks, protection and detection are typically done both on end hosts and in the network. Host agents have deep visibility on the devices where they are installed, whereas the network has broader visibility. Installing host agents may not be a viable option on IoT devices, and network-based security can only be used to protect such IoT devices. (D)TLS profile parameters of IoT device can be used by middle-boxes to detect and block malware communication, while at the same time preserving the privacy of legitimate uses of encryption. Middle-boxes need not proxy (D)TLS but can passively observe the parameters of (D)TLS handshakes from IoT devices and gain good visibility into TLS 1.0 to 1.2 parameters and partial visibility into TLS 1.3 parameters. Malicious agents can try to use the (D)TLS profile parameters as legitimate agents to evade detection but it becomes a challenge to mimic the behavior of various IoT device types and IoT device models from several manufacturers. In other words, malware developers will have to develop malicious agents per IoT device type, manufacturer and model (which will be several thousands), infect the device with specific malware agent and will have keep up with the updates to (D)TLS profile parameters of IoT devices. Further, the malware command and control server certificates needs to be signed by the same certifying authorities trusted by the IoT devices.

4. (D)TLS profile YANG module

This document specifies a YANG module for representing (D)TLS profile. The (D)TLS profile YANG module provides a method for firewall to observe the (D)TLS profile parameters in the (D)TLS handshake to permit intended use and to block malicious behavior. This module uses the common YANG types defined in [RFC6991], rules defined in [RFC8519] and cryptographic types defined in [I-D.ietf-netconf-crypto-types].

The (D)TLS profile parameters include the following:

- o (D)TLS versions supported by the IoT device
- o List of supported symmetric encryption algorithms
- o List of supported compression methods
- o List of extension types
- o List of client key exchange algorithms and the client public key lengths in versions prior to (D)TLS 1.3
- o List of trust anchor certificates used by the IoT device. Note that server certificate is encrypted in (D)TLS 1.3 and the middle-box without acting as (D)TLS proxy cannot validate the server certificate.
- o List of DHE or ECDHE groups supported by the client
- o List signature algorithms the client can validate in X.509 server certificates
- o List of SPKI pin sets pre-configured on the client to validate self-signed server certificates or raw public keys
- o If SNI mismatch is allowed or not, and if SNI mismatch is allowed, the server names for which SNI mismatch is allowed.

If the (D)TLS profile parameters are not observed in a (D)TLS session from the IoT device, the default behaviour is to block the (D)TLS session.

4.1. Tree Structure

This document augments the "ietf-mud" MUD YANG module defined in [RFC8520] for signaling the IoT device (D)TLS profile. This document

defines the YANG module "reddy-opsawg-mud-tls-profile", which has the following tree structure:

```
module: reddy-opsawg-mud-tls-profile
  augment /mud:mud/mud:from-device-policy:
    +--rw client-profile
      +--rw tls-profiles* [protocol-version supported_versions]
        +--rw protocol-version          uint16
        +--rw supported_versions        boolean
        +--rw encryption-algorithms*    encryption-algorithm
        +--rw compression-methods*     compression-method
        +--rw extension-types*          extension-type
        +--rw acceptlist-ta-certs*      ct:trust-anchor-cert-cms
        +--rw SPKI-pin-sets*            SPKI-pin-set
        +--rw SPKI-hash-algorithm       ct:hash-algorithm-t
        +--rw supported-groups*         supported-group
        +--rw signature-algorithms*     signature-algorithm
        +--rw client-public-keys
          | +--rw key-exchange-algorithms* key-exchange-algorithm
          | +--rw client-public-key-lengths* client-public-key-length
        +--rw SNI-mismatch-allowed?    boolean
        +--rw server-name*              inet:domain-name
        +--rw actions
          +--rw forwarding              identityref
```

4.2. YANG Module

```
module reddy-opsawg-mud-tls-profile {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:reddey-opsawg-mud-tls-profile";
  prefix mud-tls-profile;

  import ietf-crypto-types {
    prefix ct;
    reference "draft-ietf-netconf-crypto-types-01:
              Common YANG Data Types for Cryptography";
  }

  import ietf-inet-types {
    prefix inet;
    reference "Section 4 of RFC 6991";
  }

  import ietf-mud {
    prefix mud;
    reference "RFC 8520";
  }
}
```

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

organization
  "IETF Operations and Management Area Working Group Working Group";
contact
  "Editor: Konda, Tirumaleswar Reddy
    <mailto:TirumaleswarReddy_Konda@McAfee.com>";

description
  "This module contains YANG definition for configuring
    aliases for resources and filtering rules using DOTS
    data channel.

    Copyright (c) 2019 IETF Trust and the persons identified as
    authors of the code. All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (http://trustee.ietf.org/license-info).

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

revision 2019-06-12 {
  description
    "Initial revision.";
}

typedef compression-method {
  type uint8;
  description "Compression method.";
}

typedef extension-type {
  type uint16;
  description "Extension type.";
}

typedef encryption-algorithm {
  type uint16;
```

```
    description "Encryption algorithms.";
}

typedef supported-group {
    type uint16;
    description "supported DHE or ECDHE group.";
}

typedef SPKI-pin-set {
    type binary;
    description "Subject Public Key Info pin set.";
}

typedef signature-algorithm {
    type uint16;
    description "Signature algorithm";
}

typedef key-exchange-algorithm {
    type uint8;
    description "key exchange algorithm";
}

typedef client-public-key-length {
    type uint8;
    description "client public key length";
}

augment "/mud:mud/mud:from-device-policy" {
    container client-profile {
        list tls-profiles {
            key "protocol-version supported_versions";
            description
                "(D)TLS version profiles supported by the client";
            leaf protocol-version {
                type uint16;
                description "Legacy protocol version";
            }
            leaf supported_versions {
                type boolean;
                description "supported versions extension for TLS 1.3";
            }
            leaf-list encryption-algorithms {
                type encryption-algorithm;
                description "Encryption algorithms";
            }
            leaf-list compression-methods {
                type compression-method;
                description "Compression methods";
            }
        }
    }
}
```

```
}
leaf-list extension-types {
  type extension-type;
  description "Extension Types";
}
leaf-list acceptlist-ta-certs {
  type ct:trust-anchor-cert-cms;
  description
    "A list of trust anchor certificates used by the client";
}
leaf-list SPKI-pin-sets {
  type SPKI-pin-set;
  description
    "A list of SPKI pin sets pre-configured on the client
    to validate self-signed server certificate or
    raw public key";
}
leaf SPKI-hash-algorithm {
  type ct:hash-algorithm-t;
  description
    "cryptographic hash algorithm used to generate the SPKI pinset";
}
leaf-list supported-groups {
  type supported-group;
  description
    "A list of DHE or ECDHE groups supported by the client";
}
leaf-list signature-algorithms {
  type signature-algorithm;
  description
    "A list signature algorithms the client can validate
    in X.509 certificates.";
}
container client-public-keys {
  when "../supported_versions = 'false'";
  leaf-list key-exchange-algorithms {
    type key-exchange-algorithm;
    description
      "Key exchange algorithms supported by the client";
  }
  leaf-list client-public-key-lengths {
    type client-public-key-length;
    description
      "client public key lengths";
  }
}
leaf SNI-mismatch-allowed {
  type boolean;
```

```

    default "false";
    description
      "If set to 'false', SNI mismatch is not allowed.";
  }
  leaf-list server-name {
    when "../SNI-mismatch-allowed = 'true'";
    type inet:domain-name;
    description
      "Server names (FQDN) for which SNI mismatch is allowed.";
  }
  container actions {
    description
      "Definitions of action for this profile.";
    leaf forwarding {
      type identityref {
        base ietf-acl:forwarding-action;
      }
      mandatory true;
      description
        "Specifies the forwarding action for the (D)TLS profile.";
      reference
        "RFC 8519: YANG Data Model for Network Access  
Control Lists (ACLs)";
    }
  }
}
}
}
}

```

5. (D) TLS 1.3 handshake

In (D)TLS 1.3, full (D)TLS handshake inspection is not possible since all (D)TLS handshake messages excluding the ClientHello message are encrypted. (D)TLS 1.3 has introduced new extensions in the handshake record layers called Encrypted Extensions. Using these extensions handshake messages will be encrypted and network devices (such as a firewall) are incapable deciphering the handshake, thus cannot view the server certificate. However, a few parameters in the ServerHello are still visible such as the chosen cipher. Note that Client Key Exchange message has been removed from (D)TLS 1.3.

5.1. Encrypted SNI

To increase privacy, encrypted SNI [I-D.ietf-tls-sni-encryption] prevents passive observation of the TLS Server Name Indication and to effectively provide privacy protection, SNI encryption needs to be used in conjunction with DNS encryption (e.g., DNS-over-(D)TLS or

DNS- over-HTTPS). Firewall inspecting the (D)TLS 1.3 handshake cannot decrypt encrypted SNI. If an IoT device is configured to use public DNS-over-(D)TLS or DNS- over-HTTPS servers, the policy enforcement point is moved to that public server, which cannot enforce the MUD policy based on domain names (Section 8 of [RFC8520]). Thus the use of a public DNS-over-(D)TLS or DNS- over-HTTPS server is incompatible with MUD. A local DNS server is necessary to allow MUD policy enforcement on the local network ([I-D.ietf-doh-resolver-associated-doh] and [I-D.reddy-dprive-bootstrap-dns-server]).

5.2. Full (D)TLS 1.3 handshake inspection

Middle-box needs to act as a (D)TLS 1.3 proxy to observe the parameters of (D)TLS handshakes from IoT devices and gain good visibility into TLS 1.3 parameters. The following steps explain the mechanism to automatically bootstrap IoT devices with local network's CA certificates and to enable the middle-box to act as a (D)TLS 1.3 proxy.

- o Bootstrapping Remote Secure Key Infrastructures (BRSKI) discussed in [I-D.ietf-anima-bootstrapping-keyinfra] provides a solution for secure automated bootstrap of devices. BRSKI specifies means to provision credentials on devices to be used to operationally access networks. In addition, BRSKI provides an automated mechanism for the bootstrap distribution of CA certificates from the Enrollment over Secure Transport (EST) [RFC7030] server. The IoT device can use BRSKI to automatically bootstrap the IoT device using the IoT manufacturer provisioned X.509 certificate, in combination with a registrar provided by the local network and IoT device manufacturer's authorizing service (MASA).
- 1. The IoT device authenticates to the local network using the IoT manufacturer provisioned X.509 certificate. The IoT device can request and get a voucher from the MASA service via the registrar. The voucher is signed by the MASA service and includes the local network's CA public key.
- 2. The IoT device validates the signed voucher using the manufacturer installed trust anchor associated with the MASA, stores the CA's public key and validates the provisional TLS connection to the registrar.
- 3. The IoT device requests the full EST distribution of current CA certificates (Section 5.9.1 in [I-D.ietf-anima-bootstrapping-keyinfra]) from the registrar operating as a BRSKI-EST server. The IoT device stores the CA certificates as Explicit Trust Anchor database entries. The

IoT device uses the Explicit Trust Anchor database to validate the server certificate.

4. The middle-box uses the "supported_versions" TLS extension (defined in TLS 1.3 to negotiate the supported TLS versions between client and server) to determine the TLS version. During the (D)TLS handshake, If (D)TLS version 1.3 is used, the middle-box ((D)TLS proxy) modifies the certificate provided by the server and signs it with the private key from the local CA certificate. The middle-box has visibility into further exchanges between the IoT device and server which enables it to inspect the (D)TLS 1.3 handshake, enforce the MUD (D)TLS profile and can inspect subsequent network traffic.
5. The IoT device uses the Explicit Trust Anchor database to validate the server certificate.

The proposed technique empowers the middle-box to reject (D)TLS 1.3 sessions that violate the MUD (D)TLS profile.

6. MUD File Example

This example below contains (D)TLS profile parameters for a IoT device. JSON encoding of YANG modelled data [RFC7951] is used to illustrate the example.

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://example.com/IoTDevice",
    "last-update": "2019-18-06T03:56:40.105+10:00",
    "cache-validity": 100,
    "is-supported": true,
    "systeminfo": "IoT device name",
    "redy-opsawg-mud-tls-profile:from-device-policy": {
      "client-profile": {
        "tls-version-profile" : [
          {
            "protocol-version" : 771,
            "supported_versions_ext" : "FALSE",
            "encryption-algorithms" : [31354, 4865, 4866, 4867],
            "extension-types" : [10],
            "supported-groups" : [29],
            "actions": {
              "forwarding": "accept"
            }
          }
        ]
      }
    }
  }
}
```

7. Security Considerations

Security considerations in [RFC8520] need to be taken into consideration.

8. IANA Considerations

This document requests IANA to register the following URIs in the "ns" subregistry within the "IETF XML Registry" [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:redy-opsawg-mud-tls-profile
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

9. Acknowledgments

TODO

10. References

10.1. Normative References

- [I-D.ietf-netconf-crypto-types]
Watsen, K. and H. Wang, "Common YANG Data Types for Cryptography", draft-ietf-netconf-crypto-types-10 (work in progress), July 2019.
- [I-D.ietf-tls-dtls13]
Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", draft-ietf-tls-dtls13-31 (work in progress), March 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [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>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <<https://www.rfc-editor.org/info/rfc8519>>.

10.2. Informative References

- [I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-22 (work in progress), June 2019.
- [I-D.ietf-doh-resolver-associated-doh]
Hoffman, P., "Associating a DoH Server with a Resolver", draft-ietf-doh-resolver-associated-doh-03 (work in progress), March 2019.
- [I-D.ietf-tls-sni-encryption]
Huitema, C. and E. Rescorla, "Issues and Requirements for SNI Encryption in TLS", draft-ietf-tls-sni-encryption-04 (work in progress), November 2018.
- [I-D.reddy-dprive-bootstrap-dns-server]
K, R., Wing, D., Richardson, M., and M. Boucadair, "A Bootstrapping Procedure to Discover and Authenticate DNS-over-(D)TLS and DNS-over-HTTPS Servers", draft-reddy-dprive-bootstrap-dns-server-04 (work in progress), June 2019.
- [malware] Anderson, B., Paul, S., and D. McGrew, "Deciphering Malware's use of TLS (without Decryption)", July 2016, <<https://arxiv.org/abs/1607.01639>>.
- [RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <<https://www.rfc-editor.org/info/rfc7030>>.
- [RFC7478] Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-Time Communication Use Cases and Requirements", RFC 7478, DOI 10.17487/RFC7478, March 2015, <<https://www.rfc-editor.org/info/rfc7478>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [RFC8445] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <<https://www.rfc-editor.org/info/rfc8445>>.

[RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <<https://www.rfc-editor.org/info/rfc8520>>.

Authors' Addresses

Tirumaleswar Reddy
McAfee, Inc.
Embassy Golf Link Business Park
Bangalore, Karnataka 560071
India

Email: kondtir@gmail.com

Dan Wing
Citrix Systems, Inc.
4988 Great America Pkwy
Santa Clara, CA 95054
USA

Email: danwing@gmail.com

OPSAWG
Internet-Draft
Intended status: Informational
Expires: 26 August 2022

H. Song
Futurewei
F. Qin
China Mobile
H. Chen
China Telecom
J. Jin
LG U+
J. Shin
SK Telecom
22 February 2022

A Framework for In-situ Flow Information Telemetry
draft-song-opsawg-ifit-framework-17

Abstract

As network scale increases and network operation becomes more sophisticated, existing Operation, Administration, and Maintenance (OAM) methods are no longer sufficient to meet the monitoring and measurement requirements. Emerging data-plane on-path telemetry techniques which provide high-precision flow insight and which issue notifications in real time can supplement existing proactive and reactive methods that run in active and passive modes. These new approaches are collectively known as in-situ flow information telemetry (IFIT). They enable quality of experience for users and applications, and identification of network faults and deficiencies.

This document outlines a high-level framework for IFIT to collect and correlate performance measurement information from the network. It identifies the components that coordinate existing protocol tools and telemetry mechanisms, and addresses deployment challenges for flow-oriented on-path telemetry techniques, especially in carrier networks.

The document is a guide for system designers applying the referenced techniques. It is also intended to motivate further work to enhance the OAM ecosystem.

Status of This Memo

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

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

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

This Internet-Draft will expire on 26 August 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
1.1. Classification and Modes of On-path Telemetry	4
1.2. Requirements and Challenges	6
1.3. Scope	8
1.4. Relationship with Network Telemetry Framework (NTF)	8
1.5. Glossary	9
2. Architectural Concepts and Key Components	9
2.1. Reference Deployment	9
2.2. Key Components	11
2.2.1. Flexible Flow, Packet, and Data Selection	11
2.2.2. Flexible Data Export	13
2.2.3. Dynamic Network Probe	15
2.2.4. On-demand Technique Selection and Integration	17
2.3. IFIT for Reflective Telemetry	18
2.3.1. Intelligent Multipoint Performance Monitoring	19
2.3.2. Intent-based Network Monitoring	19
3. Guidance for Solution Developers	20
3.1. Encapsulation in Transport Protocols	20
3.2. Tunneling Support	21
3.3. Deployment Automation	21

4. Security Considerations	22
5. IANA Considerations	22
6. Contributors	22
7. Acknowledgments	22
8. References	22
8.1. Normative References	22
8.2. Informative References	23
Authors' Addresses	27

1. Introduction

Efficient network operation increasingly relies on high-quality data-plane telemetry to provide the necessary visibility into the behavior of traffic flows and network resources. Existing Operation, Administration, and Maintenance (OAM) methods, which include proactive and reactive techniques, running both active and passive modes, are no longer sufficient to meet the monitoring and measurement requirements when networks becomes more autonomous [RFC8993] and application-aware [I-D.li-apn-framework]. The complexity of today's networks and service quality requirements demand new high-precision and real-time OAM techniques.

The ability to expedite network failure detection, fault localization, and recovery mechanisms, particularly in the case of soft failures or path degradation is expected, and it must not cause service disruption. Emerging on-path telemetry techniques can provide high-precision flow insight and real-time network issue notification (e.g., jitter, latency, packet loss, significant bit error variations, and unequal load-balancing). On-Path Telemetry (OPT) refers to data-plane telemetry techniques that directly tap and measure network traffic by embedding instructions or metadata into user packets. The data provided by on-path telemetry are especially useful for verifying Service Level Agreement (SLA) compliance, user experience enhancement, service path enforcement, fault diagnosis, and network resource optimization. It is essential to recognize that existing work on this topic includes a variety of on-path telemetry techniques, including In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data], IOAM Direct Export (DEX) [I-D.ietf-ippm-ioam-direct-export], Marking-based Postcard-based Telemetry (PBT-M) [I-D.song-ippm-postcard-based-telemetry], Enhanced Alternate Marking (EAM) [I-D.zhou-ippm-enhanced-alternate-marking], and Hybrid Two-Step (HTS) [I-D.mirsky-ippm-hybrid-two-step], have been developed or proposed. These techniques can provide flow information on the entire forwarding path on a per-packet basis in real-time. The aforementioned on-path telemetry techniques differ from the active and passive OAM schemes in that they directly modify and monitor the user packets in networks so as to achieve high measurement accuracy. Formally, these on-path telemetry techniques can be classified as the

OAM hybrid type I, since they involve "augmentation or modification of the stream of interest, or employment of methods that modify the treatment of the streams", according to [RFC7799]. We name these techniques as "In-situ Flow Information Telemetry" (IFIT).

On-path telemetry is useful for application-aware networking operations, not only in data center and enterprise networks, but also in carrier networks which may cross multiple domains. The techniques can provide benefits for carrier network operators in various scenarios. For example, it is critical for the operators who offer high-bandwidth, latency and loss-sensitive services such as video streaming and online gaming to closely monitor the relevant flows in real-time as the basis for any further optimizations.

This framework document is intended to guide system designers attempting to use the referenced techniques as well as to motivate further work to enhance the telemetry ecosystem. It highlights requirements and challenges, outlines important techniques that are applicable, and provides examples of how these might be applied for critical use cases.

The document scope is discussed in Section 1.3.

1.1. Classification and Modes of On-path Telemetry

The operation of IFIT differs from both active OAM and passive OAM as defined in [RFC7799]. It does not generate any active probe packets or passively observe unmodified user packets. Instead, it modifies selected user packets in order to collect useful information about them. Therefore, the operation is categorized as the hybrid OAM type I method per [RFC7799].

This hybrid OAM type I method can be further partitioned into two modes [passport-postcard]. In the passport mode, each node on the path can add telemetry data to the user packets (i.e., stamps the passport). The accumulated data trace is exported at a configured end node. In the postcard mode, each node directly exports the telemetry data using an independent packet (i.e., sends a postcard) while the user packets are unmodified. It is possible to combine the two modes together in one solution. We call this the hybrid mode.

Figure 1 shows the classification of the on-path telemetry techniques.

Mode	Passport	Postcard	Hybrid
Technique	IOAM Trace IOAM E2E	IOAM DEX PBT-M EAM	Multicast Telemetry HTS

Figure 1: On-path Telemetry Technique Classification

IOAM Trace and E2E options are described in [I-D.ietf-ippm-ioam-data].

EAM is described in [I-D.zhou-ippm-enhanced-alternate-marking].

IOAM DEX option is described in [I-D.ietf-ippm-ioam-direct-export].

PBT-M is described in [I-D.song-ippm-postcard-based-telemetry].

Multicast Telemetry is described in [I-D.ietf-mboned-multicast-telemetry].

HTS is described in [I-D.mirsky-ippm-hybrid-two-step].

The advantages of the passport mode include:

- * It automatically retains the telemetry data correlation along the entire path. The self-describing feature simplifies the data consumption.
- * The on-path data for a packet is only exported once so the data export overhead is low.
- * Only the head and tail nodes of the paths need to be configured for header insertion and removal, so the configuration overhead is low.

The disadvantages of the passport mode include:

- * The telemetry data carried by user packets inflate the packet size, which may be undesirable or prohibitive.
- * Approaches for encapsulating the instruction header and data in transport protocols need to be standardized.
- * Carrying sensitive data along the path is vulnerable to security and privacy breach.

- * If a packet is dropped on the path, the data collected are also lost.

The postcard mode complements the passport mode. The advantages of the postcard mode include:

- * Either there is no packet header overhead (e.g., PBT-M) or the overhead is small and fixed (e.g., IOAM DEX).
- * The encapsulation requirement may be avoided (e.g., PBT-M).
- * The telemetry data can be secured before export.
- * Even if a packet is dropped on the path, the partial data collected are still available.

The disadvantages of the postcard mode include:

- * Telemetry data are spread in multiple postcards so extra effort is needed to correlate the data.
- * Every node exports a postcard for a packet which increases the data export overhead.
- * In case of PBT-M, every node on the path needs to be configured, so the configuration overhead is high.
- * In case of IOAM DEX, the transport encapsulation requirement remains.

The hybrid mode either tailors for some specific application scenario (e.g., Multicast Telemetry) or provides some alternative approach (e.g., HTS). A postcard can be sent per segment of a path or the telemetry data can be carried in a companion packet following each monitored use packet. The hybrid mode combines the advantages of both the passport mode and the postcard mode, but it may incur extra processing complexity.

1.2. Requirements and Challenges

Although on-path telemetry is beneficial, successfully applying such techniques in carrier networks must consider performance, deployability, and flexibility. Specifically, we need to address the following practical deployment challenges:

- * C1: On-path telemetry incurs extra packet processing which may cause stress on the network data plane. The potential impact on the forwarding performance creates an unfavorable "observer

effect" (where the actions of performing on-path telemetry may change the behavior of the traffic being measured). This will not only damage the fidelity of the measurement, but also defy the purpose of the measurement.

- * C2: On-path telemetry can generate a considerable amount of data which may claim too much transport bandwidth and inundate the servers for data collection, storage, and analysis. For example, if the technique is applied to all the traffic, one node may collect a few tens of bytes as telemetry data for each packet. The whole forwarding path might accumulate telemetry data with a size similar to or even exceeding that of the original packet.
- * C3: The collectible data defined currently are essential but limited. This, in turn, limits the management and operational techniques that can be applied. Flexibility and extensibility of data definition, aggregation, acquisition, and filtering, must be considered.
- * C4: Applying only a single underlying on-path telemetry technique may miss some important events or lead to incorrect results. For example, packet drop can cause the loss of the flow telemetry data and the packet drop location and reason remains unknown if only the In-situ OAM trace option is used. A comprehensive solution needs the flexibility to switch between different underlying techniques and adjust the configurations and parameters at runtime. Thus, system-level orchestration is needed.
- * C5: We must provide solutions to support an incremental deployment strategy. That is, we need to support established encapsulation schemes for various predominant protocols such as Ethernet, IPv6, and MPLS with backward compatibility and properly handle various transport tunnels.
- * C6: The development of simplified on-path telemetry primitives and models for configuration and queries is essential. Telemetry models may be utilized via an API-based telemetry service for external applications, for end-to-end performance measurement and application performance monitoring. Standard-based protocols and methods are needed for network configuration and programming, and telemetry data pre-processing and export, to provide interoperability.

1.3. Scope

Following the network telemetry framework discussed in [I-D.ietf-opsawg-ntf], this document focuses on the on-path telemetry, a specific class of data-plane telemetry techniques, and provides a high-level framework which addresses the challenges for deployment listed in Section 1.2, especially in carrier networks.

This document aims to clarify the problem space, essential requirements, and summarizes best practices and general system design considerations. This document provides some examples to show the novel network telemetry applications under the framework.

As an informational document, it describes an open framework with a few key components. The framework does not enforce any specific implementation on each component, neither does it define interfaces (e.g., API, protocol) between components. The choice of underlying on-path telemetry techniques and other implementation details is determined by the application implementer. Therefore, the framework is not a solution specification. It only provides a high-level overview and is not necessarily a mandatory recommendation for on-path telemetry applications.

The standardization of the underlying techniques and interfaces mentioned in this document is undertaken by various working groups. Due to the limited scope and intended status of this document, it has no overlap or conflict with those works.

1.4. Relationship with Network Telemetry Framework (NTF)

[I-D.ietf-opsawg-ntf] describes a Network Telemetry Framework (NTF). One dimension used by NTF to partition network telemetry techniques and systems is based on the three planes in networks (i.e., control plane, management plane, and forwarding plane) and external data sources. IFIT fits in the category of forwarding-plane telemetry and deals with the specific on-path technical branch of the forwarding-plane telemetry.

According to NTF, an on-path telemetry application mainly subscribes to event-triggered or streaming data. The key functional components of IFIT described in Section 2.2 match the general components in NTF with more specific functions. "On-demand Technique Selection and Integration" is an application layer function, matching the "Data Query, Analysis, and Storage" component in NTF; "Flexible Flow, Packet, and Data Selection" matches the "Data Configuration and Subscription" component; "Flexible Data Export" matches the "Data Encoding and Export" component; "Dynamic Network Probe" matches the "Data Generation and Processing" component.

1.5. Glossary

This section defines and explains the acronyms and terms used in this document.

On-path Telemetry: Remotely acquiring performance and behavior data about network flows on a per-packet basis on the packet's forwarding path. The term refers to a class of data-plane telemetry techniques, including IOAM, PBT, EAM, and HTS. Such techniques may need to mark user packets, or insert instruction/metadata into the headers of user packets.

IFIT: In-situ Flow Information Telemetry is a high-level reference framework that shows how network data-plane monitoring and measurement applications can address the deployment challenges of the flow-oriented on-path telemetry techniques.

Reflective Telemetry: The reflective telemetry functions in a dynamic and closed-loop fashion. A new telemetry action is provisioned as a result of self-knowledge acquired through prior telemetry actions.

2. Architectural Concepts and Key Components

To address the challenges mentioned in Section 1.2, a high-level framework which can help to build a workable and efficient on-path telemetry application is presented. In-situ Flow Information Telemetry (IFIT) is dedicated to on-path telemetry data about user and application traffic flows. It covers a class of on-path telemetry techniques and works at a level higher than any specific underlying technique. The framework is comprised of some key functional components (Section 2.2). By assembling these components, IFIT supports reflective telemetry that enables autonomous network operations (Section 2.3).

2.1. Reference Deployment

Figure 2 shows a reference deployment scenario of on-path telemetry.

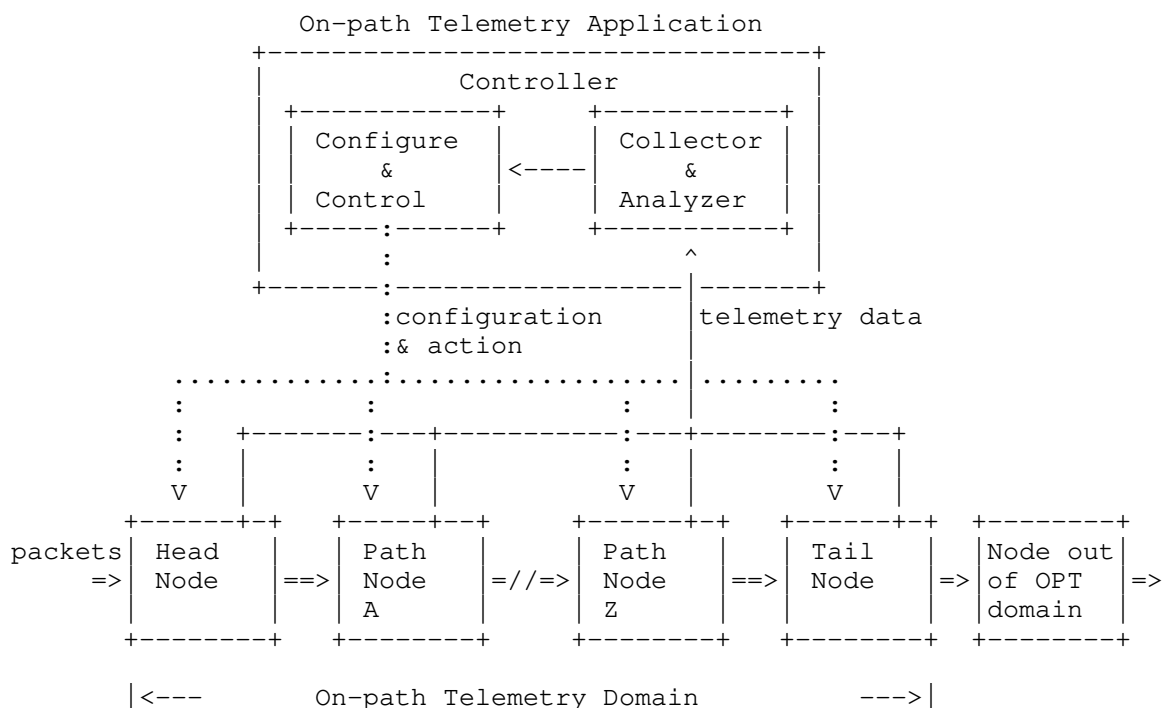


Figure 2: Deployment Scenario

An on-path telemetry application can conduct network data-plane monitoring and measurement tasks over a limited domain [RFC8799] by applying one or more underlying techniques. The application contains multiple elements, including configuring the network nodes and processing the telemetry data. The application usually uses a logically centralized controller for configuring the network nodes in the domain, and collecting and analyzing telemetry data. The configuration determines which underlying technique is used, what telemetry data are of interest, which flows and packets are concerned with, how the telemetry data are collected, etc. The process can be dynamic and interactive: after the telemetry data processing and analyzing, the application may instruct the controller to modify the configuration of the nodes, which affects the future telemetry data collection.

From the system-level view, it is recommended to use standardized configuration and data collection interfaces, regardless of the underlying technique. The specification of these interfaces and the implementation of the controller are out of scope for this document.

The on-path telemetry domain encompasses the head nodes and the end nodes, and may cross multiple network domains. The head nodes are responsible for enabling the on-path telemetry functions and the end nodes are responsible for terminating them. All capable nodes in this domain will be capable of executing the instructed on-path telemetry function. It is important to note that any application must, through configuration and policy, guarantee that any packet with on-path telemetry header and metadata will not leak out of the domain.

The underlying on-path telemetry techniques covered by the IFIT framework can be of any modes discussed in Section 1.1.

2.2. Key Components

The key components of IFIT to address the challenges listed in Section 1.2 are as follows. The components are described in more detail in the sections that follow.

- * Flexible flow, packet, and data selection policy, addressing the challenge C1 described in Section 1;
- * Flexible data export, addressing the challenge C2;
- * Dynamic network probe, addressing C3;
- * On-demand technique selection and integration, addressing C4.

Note that the challenges C5 and C6 are mostly standard-related, and are fundamental to IFIT. We discuss the protocol implications and guidance for solution developers in Section 3.

2.2.1. Flexible Flow, Packet, and Data Selection

In most cases, it is impractical to enable data collection for all the flows and for all the packets in a flow due to the potential performance and bandwidth impact. Therefore, a workable solution usually need to select only a subset of flows and flow packets on which to enable data collection, even though this means the loss of some information and accuracy.

In the data plane, a flow filter like those used for an Access Control List (ACL) provides an ideal means to determine the subset of flows. An application can set a sample rate or probability to a flow to allow only a subset of flow packets to be monitored, collect a different set of data for different packets, and disable or enable data collection on any specific network node. An application can further allow any node to accept or deny the data collection process in full or partially.

Based on these flexible mechanisms, IFIT allows applications to apply flexible flow and data selection policies to suit their requirements. The applications can dynamically change the policies at any time based on the network load, processing capability, focus of interest, and any other criteria.

2.2.1.1. Block Diagram

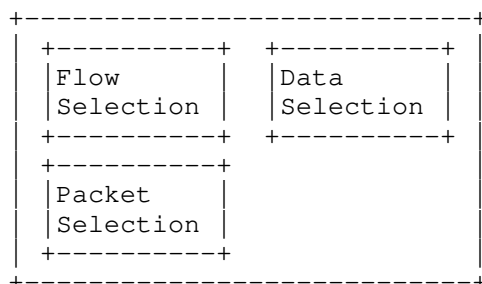


Figure 3: Flexible Flow, Packet, and Data Selection

Figure 3 shows the block diagram of this component. The flow selection block defines the policies to choose target flows for monitoring. Flow has different granularity. A basic flow is defined by 5-tuple IP header fields. Flow can also be aggregated at interface level, tunnel level, protocol level, and so on. The packet selection block defines the policies to choose packets from a target flow. The policy can be either a sampling interval, a sampling probability, or some specific packet signature. The data selection block defines the set of data to be collected. This can be changed on a per-packet or per-flow basis.

2.2.1.2. Example: Sketch-guided Elephant Flow Selection

Network operators are usually more interested in elephant flows which consume more resource and are sensitive to changes in network conditions. A CountMin Sketch [CMSketch] can be used on the data path of the head nodes, which identifies and reports the elephant flows periodically. The controller maintains a current set of elephant flows and dynamically enables the on-path telemetry for only these flows.

2.2.1.3. Example: Adaptive Packet Sampling

Applying on-path telemetry on all packets of the selected flows can still be out of reach. A sample rate should be set for these flows and telemetry should only be enabled on the sampled packets. However, the head nodes have no clue on the proper sampling rate. An overly high rate would exhaust the network resource and even cause packet drops; An overly low rate, on the contrary, would result in the loss of information and inaccuracy of measurements.

An adaptive approach can be used based on the network conditions to dynamically adjust the sampling rate. Every node gives user traffic forwarding higher priority than telemetry data export. In case of network congestion, the telemetry can sense some signals from the data collected (e.g., deep buffer size, long delay, packet drop, and data loss). The controller may use these signals to adjust the packet sampling rate. In each adjustment period (i.e., RTT of the feedback loop), the sampling rate is either decreased or increased in response of the signals. An Additive Increase/Multiplicative Decrease (AIMD) policy similar to the TCP flow control mechanism for rate adjustment can be used.

2.2.2. Flexible Data Export

The flow telemetry data can catch the dynamics of the network and the interactions between user traffic and network. Nevertheless, the data may contain redundancy. It is advisable to remove the redundancy from the data in order to reduce the data transport bandwidth and server processing load.

In addition to efficient export data encoding (e.g., IPFIX [RFC7011] or protobuf (<https://developers.google.com/protocol-buffers/>)), nodes have several other ways to reduce the export data by taking advantage of network device's capability and programmability. Nodes can cache the data and send the accumulated data in batches if the data is not time sensitive. Various deduplication and compression techniques can be applied on the batched data.

From the application perspective, an application may only be interested in some special events which can be derived from the telemetry data. For example, in the case that the forwarding delay of a packet exceeds a threshold, or a flow changes its forwarding path is of interest, it is unnecessary to send the original raw data to the data collecting and processing servers. Rather, IFIT takes advantage of the in-network computing capability of network devices to process the raw data and only push the event notifications to the subscribing applications.

Such events can be expressed as policies. A policy can request data export only on change, on exception, on timeout, or on threshold.

2.2.2.1. Block Diagram

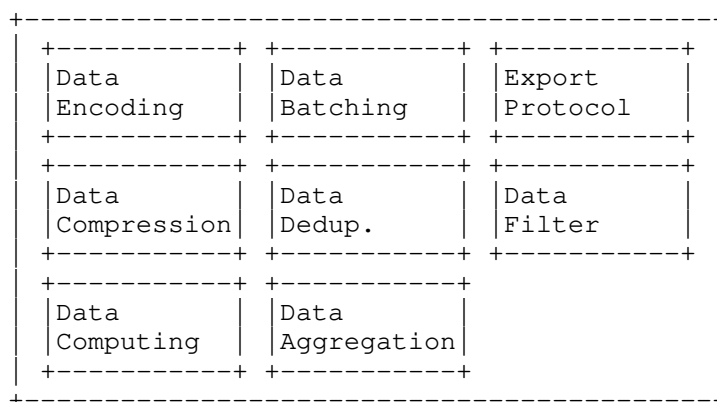


Figure 4: Flexible Data Export

Figure 4 shows the block diagram of this component. The data encoding block defines the method to encode the telemetry data. The data batching block defines the size of batch data buffered at the device side before export. The export protocol block defines the protocol used for telemetry data export. The data compression block defines the algorithm to compress the raw data. The data deduplication block defines the algorithm to remove the redundancy in the raw data. The data filter block defines the policies to filter the needed data. The data computing block defines the policies to preprocess the raw data and generate some new data. The data aggregation block defines the procedure to combine and synthesize the data.

2.2.2.2. Example: Event-based Anomaly Monitor

Network operators are interested in anomalies such as path change, network congestion, and packet drop. Such anomalies are hidden in raw telemetry data (e.g., path trace, timestamp). Such anomalies can be described as events and programmed into the device data plane. Only the triggered events are exported. For example, if a new flow appears at any node, a path change event is triggered; if the packet delay exceeds a predefined threshold in a node, the congestion event is triggered; if a packet is dropped due to buffer overflow, a packet drop event is triggered.

The export data reduction due to such optimization is substantial. For example, given a single 5-hop 10Gbps path, assume a moderate number of 1 million packets per second are monitored, and the telemetry data plus the export packet overhead consume less than 30 bytes per hop. Without such optimization, the bandwidth consumed by the telemetry data can easily exceed 1Gbps (more than 10% of the path bandwidth). When the optimization is used, the bandwidth consumed by the telemetry data is negligible. Moreover, the pre-processed telemetry data greatly simplify the work of data analyzers.

2.2.3. Dynamic Network Probe

Due to limited data plane resource and network bandwidth, it is unlikely one can monitor all the data all the time. On the other hand, the data needed by applications may be arbitrary but ephemeral. It is critical to meet the dynamic data requirements with limited resource.

Fortunately, data plane programmability allows new data probes to be dynamically loaded. These on-demand probes are called Dynamic Network Probes (DNP). DNP is the technique to enable probes for customized data collection in different network planes. When working with an on-path telemetry technique, DNP is loaded into the data plane through incremental programming or configuration. The DNP can effectively conduct data generation, processing, and aggregation.

DNP introduces flexibility and extensibility to IFIT. It can implement the optimizations for export data reduction motioned in the previous section. It can also generate custom data as required by today's and tomorrow's applications.

2.2.3.1. Block Diagram

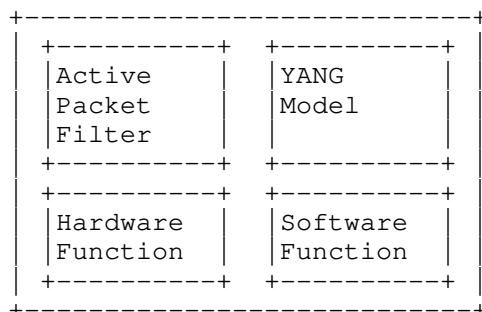


Figure 5: Dynamic Network Probes

Figure 5 shows the block diagram of this component. The active packet filter block is available in most hardware and it defines DNP through dynamically update the packet filtering policies (including flow selection and action). YANG models can be dynamically deployed to enable different data processing and filtering functions. Some hardware allows dynamically loading hardware-based functions into the forwarding path at runtime through mechanisms such as reserved pipelines and function stubs. Dynamically loadable software functions can be implemented in the control processors in capable nodes.

2.2.3.2. Examples

Following are some possible DNPs that can be dynamically deployed to support applications.

On-demand Flow Sketch: A flow sketch is a compact online data structure (usually a variation of multi-hashing table) for approximate estimation of multiple flow properties. It can be used to facilitate flow selection. The aforementioned CountMin Sketch [CMSketch] is such an example. Since a sketch consumes data plane resources, it should only be deployed when actually needed.

Smart Flow Filter: The policies that choose flows and packet sampling rate can change during the lifetime of an application.

Smart Statistics: An application may need to count flows based on different flow granularity or maintain hit counters for selected flow table entries.

Smart Data Reduction: DNP can be used to program the events that conditionally trigger data export.

2.2.4. On-demand Technique Selection and Integration

With multiple underlying data collection and export techniques at its disposal, IFIT can flexibly adapt to different network conditions and different application requirements.

For example, depending on the types of data that are of interest, IFIT may choose either passport or postcard mode to collect the data; if an application needs to track down where the packets are lost, switching from passport mode to postcard mode should be supported.

IFIT can further integrate multiple data plane monitoring and measurement techniques together and present a comprehensive data plane telemetry solution.

Based on the application requirements and the real-time telemetry data analysis results, new configurations and actions can be deployed.

2.2.4.1. Block Diagram

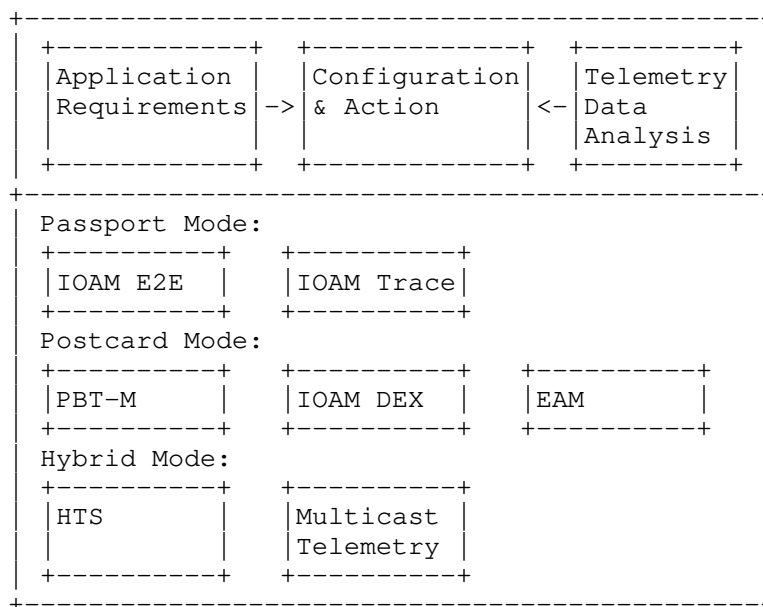


Figure 6: Technique Selection and Integration

Figure 6 shows the block diagram of this component, which lists the candidate on-path telemetry techniques.

Located in the logically centralized controller, this component makes all the control and configuration dynamically to the capable nodes in the domain which will affect the future telemetry data. The configuration and action decisions are based on the inputs from the application requirements and the realtime telemetry data analysis results. Note that here the telemetry data source is not limited to the data plane. The data can come form all the sources mentioned in [I-D.ietf-opsawg-ntf], including external data sources.

2.3. IFIT for Reflective Telemetry

The components described in Section 2.2 can work together to support reflective telemetry, as shown in Figure 7.

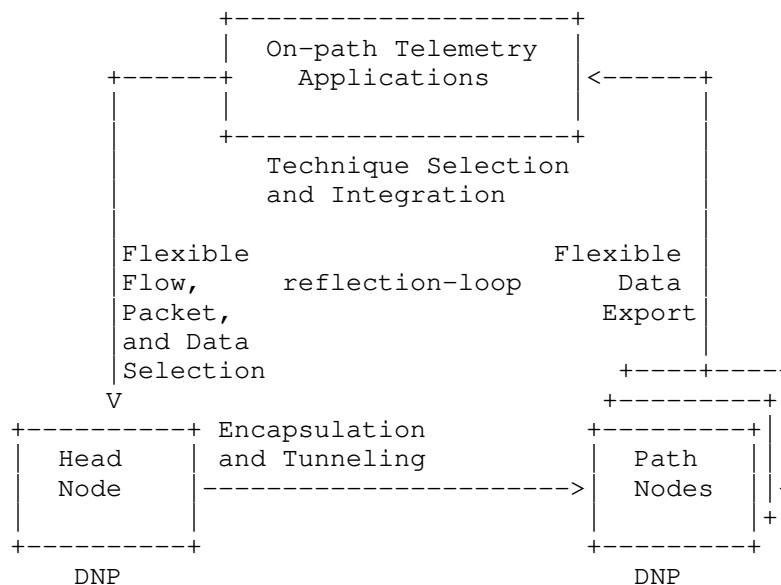


Figure 7: IFIT-based Reflective Telemetry

An application may pick a suite of telemetry techniques based on its requirements and apply an initial technique to the data plane. It then configures the head nodes to decide the initial target flows/packets and telemetry data set, the encapsulation and tunneling scheme based on the underlying network architecture, and the IFIT-capable nodes to decide the initial telemetry data export policy. Based on the network condition and the analysis results of the telemetry data, the application can change the telemetry technique, the flow/data selection policy, and the data export approach in real time without breaking the normal network operation. Many of such dynamic changes can be done through loading and unloading DNP.

The reflective telemetry enabled by the IFIT allows numerous new applications. Two examples are provided below.

2.3.1. Intelligent Multipoint Performance Monitoring

[RFC8889] describes an intelligent performance management based on the network condition. The idea is to split the monitoring network into clusters. The cluster partition that can be applied to every type of network graph and the possibility to combine clusters at different levels enable the so-called Network Zooming. It allows a controller to calibrate the network telemetry, so that it can start without examining in depth and monitor the network as a whole. In case of necessity (packet loss or too high delay), an immediate detailed analysis can be reconfigured. In particular, the controller, that is aware of the network topology, can set up the most suitable cluster partition by changing the traffic filter or activate new measurement points and the problem can be localized with a step-by-step process.

An application on top of the controllers can manage such mechanism, whose dynamic and reflective operations are supported by the IFIT framework.

2.3.2. Intent-based Network Monitoring

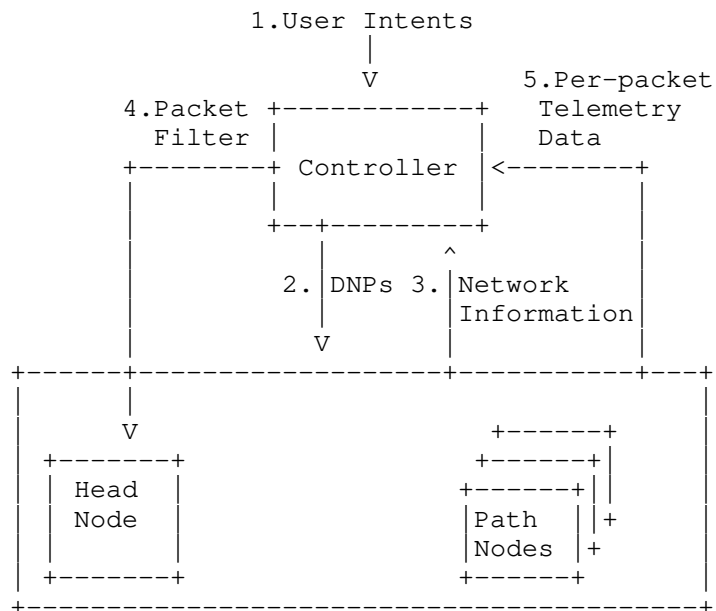


Figure 8: Intent-based Monitoring

In this example, a user can express high level intents for network monitoring. The controller translates an intent and configures the corresponding DNPs in capable nodes which collect necessary network information. Based on the real-time information feedback, the controller runs a local algorithm to determine the suspicious flows. It then deploys specific packet filters to the head node to initiate the high precision per-packet on-path telemetry for these flows.

3. Guidance for Solution Developers

Having a high-level framework covering a class of related techniques promotes a holistic approach for standard development and helps to avoid duplicated efforts and piecemeal solutions that only focus on a specific technique while omitting the compatibility and extensibility issues, which is important to a healthy ecosystem for network telemetry.

A complete IFIT-based solution needs standard interfaces for configuration and data extraction, and standard encapsulation on various transport protocols. It may also need standard API and primitives for application programming and deployment.

[I-D.ietf-ippm-ioam-deployment] summarizes some techniques for encapsulation and data export for IOAM. Solution developers need to consider the aspects set out in the following subsections.

3.1. Encapsulation in Transport Protocols

Since the introduction of IOAM, the IOAM option header encapsulation schemes in various network protocols have been defined (e.g., [I-D.ietf-ippm-ioam-ipv6-options]). Similar encapsulation schemes are needed to cover the other on-path telemetry techniques. Meanwhile, the on-path telemetry header/data encapsulation schemes in some popular protocols, such as MPLS and SRv6, are also needed. PBT-M [I-D.song-ippm-postcard-based-telemetry] does not introduce new headers to the packets so the trouble of encapsulation for a new header is avoided. While there are some proposals which allow new header encapsulation in MPLS packets (e.g., [I-D.song-mpls-extension-header]) or in SRv6 packets (e.g., [I-D.song-spring-siam]), they are still in their infancy stage and require further work. Before standards are available, in a confined domain, pre-standard encapsulation approaches may be applied.

3.2. Tunneling Support

In carrier networks, it is common for user traffic to traverse various tunnels for QoS, traffic engineering, or security. Both the uniform mode and the pipe mode for tunnel support are required and described in [I-D.song-ippm-ioam-tunnel-mode]. The uniform mode treats the nodes in a tunnel uniformly as the nodes outside of the tunnel on a path. In contrast, the pipe mode abstracts all the nodes between the tunnel ingress and egress as a circuit so no nodes in the tunnel is visible to the nodes outside of the tunnel. With such flexibility, the operator can either gain a true end-to-end visibility or apply a hierarchical approach which isolates the monitoring domain between customer and provider.

3.3. Deployment Automation

Standard approaches that automate the function configuration, and capability query and advertisement, could either be deployed in a centralized fashion or a distributed fashion. The draft [I-D.ietf-ippm-ioam-yang] provides a YANG model for IOAM configuration. Similar models needs to be defined for other techniques. It is also helpful to provide standards-based approaches for configuration in various network environments. For example, in Segment Routing (SR) networks, extensions to BGP or Path Computation Element Communication Protocol (PCEP) can be defined to distribute SR policies carrying on-path telemetry information, so that telemetry behavior can be enabled automatically when the SR policy is applied. [I-D.chen-pce-sr-policy-ifit] defines extensions to PCEP to configure SR policies for on-path telemetry. [I-D.ietf-idr-sr-policy-ifit] defines extensions to BGP for the same purpose. Additional capability discovery and dissemination will be needed for other types of networks.

To realize the potential of on-path telemetry, programming and deploying DNPs are important. ForCES [RFC5810] is a standard protocol for network device programming, which can be used for DNP deployment. Currently some related works such as [I-D.www-netmod-event-yang] and [I-D.bwd-netmod-eca-framework] have proposed to use YANG models to define the smart policies which can be used to implement DNPs. In the future, other approaches for hardware and software-based functions can be development to enhance the programmability and flexibility.

4. Security Considerations

In addition to the specific security issues discussed in each individual document on on-path telemetry, this document considers the overall security issues at the system level. This should serve as a guide to the on-path telemetry application developers and users. General security and privacy considerations for any network telemetry system are also discussed in [I-D.ietf-opsawg-ntf].

Since the on-path telemetry techniques work on the network forwarding plane, the IFIT framework poses some security risks. The important and sensitive information about a network could be exposed to an attacker. Further, the on-path telemetry data might swamp various parts of the network, leading to a possible DoS attack.

Fortunately, security measures can be enforced on various parts of the framework to mitigate such threats. For example, the configuration can filter and rate limit the monitored traffic; encryption and authentication can be applied on the exported telemetry data; different underlying techniques can be chosen to adapt to the different network conditions.

5. IANA Considerations

This document includes no request to IANA.

6. Contributors

Other major contributors of this document include Giuseppe Fioccola, Daniel King, Zhenqiang Li, Zhenbin Li, Tianran Zhou, and James Guichard.

7. Acknowledgments

We thank Diego Lopez, Shwetha Bhandari, Joe Clarke, Adrian Farrel, Frank Brockners, Al Morton, Alex Clemm, Alan DeKok, Benoit Claise, and Warren Kumari for their constructive suggestions for improving this document.

8. References

8.1. Normative References

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

- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [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>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.

8.2. Informative References

- [CMSketch] Cormode, G. and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications", 2005, <<http://dx.doi.org/10.1016/j.jalgor.2003.12.001>>.
- [I-D.bwd-netmod-eca-framework]
Boucadair, M., Wu, Q., Wang, M., King, D., and C. Xie, "Framework for Use of ECA (Event Condition Action) in Network Self Management", Work in Progress, Internet-Draft, draft-bwd-netmod-eca-framework-00, 3 November 2019, <<https://www.ietf.org/archive/id/draft-bwd-netmod-eca-framework-00.txt>>.
- [I-D.chen-pce-sr-policy-ifit]
Chen, H., Yuan, H., Zhou, T., Li, W., Fioccola, G., and Y. Wang, "PCEP SR Policy Extensions to Enable IFIT", Work in Progress, Internet-Draft, draft-chen-pce-sr-policy-ifit-02, 10 July 2020, <<https://www.ietf.org/archive/id/draft-chen-pce-sr-policy-ifit-02.txt>>.
- [I-D.herbert-ipv4-eh]
Herbert, T., "IPv4 Extension Headers and Flow Label", Work in Progress, Internet-Draft, draft-herbert-ipv4-eh-01, 2 May 2019, <<https://www.ietf.org/archive/id/draft-herbert-ipv4-eh-01.txt>>.
- [I-D.ietf-idr-sr-policy-ifit]
Qin, F., Yuan, H., Zhou, T., Fioccola, G., and Y. Wang, "BGP SR Policy Extensions to Enable IFIT", Work in Progress, Internet-Draft, draft-ietf-idr-sr-policy-ifit-03, 10 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-idr-sr-policy-ifit-03.txt>>.

[I-D.ietf-ippm-ioam-data]

Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-data-17, 13 December 2021, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-data-17.txt>>.

[I-D.ietf-ippm-ioam-deployment]

Brockners, F., Bhandari, S., Bernier, D., and T. Mizrahi, "In-situ OAM Deployment", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-deployment-00, 19 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-deployment-00.txt>>.

[I-D.ietf-ippm-ioam-direct-export]

Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-direct-export-07, 13 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-direct-export-07.txt>>.

[I-D.ietf-ippm-ioam-ipv6-options]

Bhandari, S. and F. Brockners, "In-situ OAM IPv6 Options", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-ipv6-options-07, 6 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-ipv6-options-07.txt>>.

[I-D.ietf-ippm-ioam-yang]

Zhou, T., Guichard, J., Brockners, F., and S. Raghavan, "A YANG Data Model for In-Situ OAM", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-yang-03, 25 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-yang-03.txt>>.

[I-D.ietf-mboned-multicast-telemetry]

Song, H., McBride, M., Mirsky, G., Mishra, G., Asaeda, H., and T. Zhou, "Multicast On-path Telemetry Solutions", Work in Progress, Internet-Draft, draft-ietf-mboned-multicast-telemetry-02, 4 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-mboned-multicast-telemetry-02.txt>>.

[I-D.ietf-opsawg-ntf]

Song, H., Qin, F., Martinez-Julia, P., Ciavaglia, L., and A. Wang, "Network Telemetry Framework", Work in Progress, Internet-Draft, draft-ietf-opsawg-ntf-13, 3 December 2021, <<https://www.ietf.org/archive/id/draft-ietf-opsawg-ntf-13.txt>>.

[I-D.li-apn-framework]

Li, Z., Peng, S., Voyer, D., Li, C., Liu, P., Cao, C., Mishra, G., Ebisawa, K., Previdi, S., and J. N. Guichard, "Application-aware Networking (APN) Framework", Work in Progress, Internet-Draft, draft-li-apn-framework-04, 25 October 2021, <<https://www.ietf.org/archive/id/draft-li-apn-framework-04.txt>>.

[I-D.mirsky-ippm-hybrid-two-step]

Mirsky, G., Lingqiang, W., Zhui, G., and H. Song, "Hybrid Two-Step Performance Measurement Method", Work in Progress, Internet-Draft, draft-mirsky-ippm-hybrid-two-step-12, 26 January 2022, <<https://www.ietf.org/archive/id/draft-mirsky-ippm-hybrid-two-step-12.txt>>.

[I-D.song-ippm-ioam-tunnel-mode]

Song, H., Li, Z., Zhou, T., and Z. Wang, "In-situ OAM Processing in Tunnels", Work in Progress, Internet-Draft, draft-song-ippm-ioam-tunnel-mode-00, 27 June 2018, <<https://www.ietf.org/archive/id/draft-song-ippm-ioam-tunnel-mode-00.txt>>.

[I-D.song-ippm-postcard-based-telemetry]

Song, H., Mirsky, G., Filsfils, C., Abdelsalam, A., Zhou, T., Li, Z., Shin, J., and K. Lee, "In-Situ OAM Marking-based Direct Export", Work in Progress, Internet-Draft, draft-song-ippm-postcard-based-telemetry-11, 15 November 2021, <<https://www.ietf.org/archive/id/draft-song-ippm-postcard-based-telemetry-11.txt>>.

[I-D.song-mpls-extension-header]

Song, H., Li, Z., Zhou, T., Andersson, L., and Z. Zhang, "MPLS Extension Header", Work in Progress, Internet-Draft, draft-song-mpls-extension-header-06, 10 January 2022, <<https://www.ietf.org/archive/id/draft-song-mpls-extension-header-06.txt>>.

[I-D.song-spring-siam]

Song, H. and T. Pan, "SRv6 In-situ Active Measurement", Work in Progress, Internet-Draft, draft-song-spring-siam-02, 6 December 2021, <<https://www.ietf.org/archive/id/draft-song-spring-siam-02.txt>>.

[I-D.wwx-netmod-event-yang]

Wu, Q., Bryskin, I., Birkholz, H., Liu, X., and B. Claise, "A YANG Data model for ECA Policy Management", Work in Progress, Internet-Draft, draft-wwx-netmod-event-yang-10, 1 November 2020, <<https://www.ietf.org/archive/id/draft-wwx-netmod-event-yang-10.txt>>.

[I-D.zhou-ippm-enhanced-alternate-marking]

Zhou, T., Fioccola, G., Liu, Y., Lee, S., Cociglio, M., and W. Li, "Enhanced Alternate Marking Method", Work in Progress, Internet-Draft, draft-zhou-ippm-enhanced-alternate-marking-08, 4 January 2022, <<https://www.ietf.org/archive/id/draft-zhou-ippm-enhanced-alternate-marking-08.txt>>.

[passport-postcard]

Handigol, N., Heller, B., Jeyakumar, V., Mazieres, D., and N. McKeown, "Where is the debugger for my software-defined network?", 2012, <<https://doi.org/10.1145/2342441.2342453>>.

[RFC5810]

Doria, A., Ed., Hadi Salim, J., Ed., Haas, R., Ed., Khosravi, H., Ed., Wang, W., Ed., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", RFC 5810, DOI 10.17487/RFC5810, March 2010, <<https://www.rfc-editor.org/info/rfc5810>>.

[RFC7011]

Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.

[RFC8889]

Fioccola, G., Ed., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8889, DOI 10.17487/RFC8889, August 2020, <<https://www.rfc-editor.org/info/rfc8889>>.

[RFC8993] Behringer, M., Ed., Carpenter, B., Eckert, T., Ciavaglia, L., and J. Nobre, "A Reference Model for Autonomic Networking", RFC 8993, DOI 10.17487/RFC8993, May 2021, <<https://www.rfc-editor.org/info/rfc8993>>.

Authors' Addresses

Haoyu Song
Futurewei
2330 Central Expressway
Santa Clara,
United States of America
Email: haoyu.song@futurewei.com

Fengwei Qin
China Mobile
No. 32 Xuanwumenxi Ave., Xicheng District
Beijing, 100032
P.R. China
Email: qinfengwei@chinamobile.com

Huanan Chen
China Telecom
Email: chenhuan6@chinatelecom.cn

Jaehwan Jin
LG U+
South Korea
Email: daenamul@lguplus.co.kr

Jongyoon Shin
SK Telecom
South Korea
Email: jongyoon.shin@sk.com

ippm
Internet-Draft
Intended status: Informational
Expires: August 25, 2022

M. Spiegel
Barefoot Networks, an Intel company
F. Brockners
Cisco
S. Bhandari
Thoughtspot
R. Sivakolundu
Cisco
February 21, 2022

In-situ OAM raw data export with IPFIX
draft-spiegel-ippm-ioam-rawexport-06

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document discusses how In-situ Operations, Administration, and Maintenance (IOAM) information can be exported in raw, i.e. uninterpreted, format from network devices to systems, such as monitoring or analytics systems using IPFIX.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 25, 2022.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

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

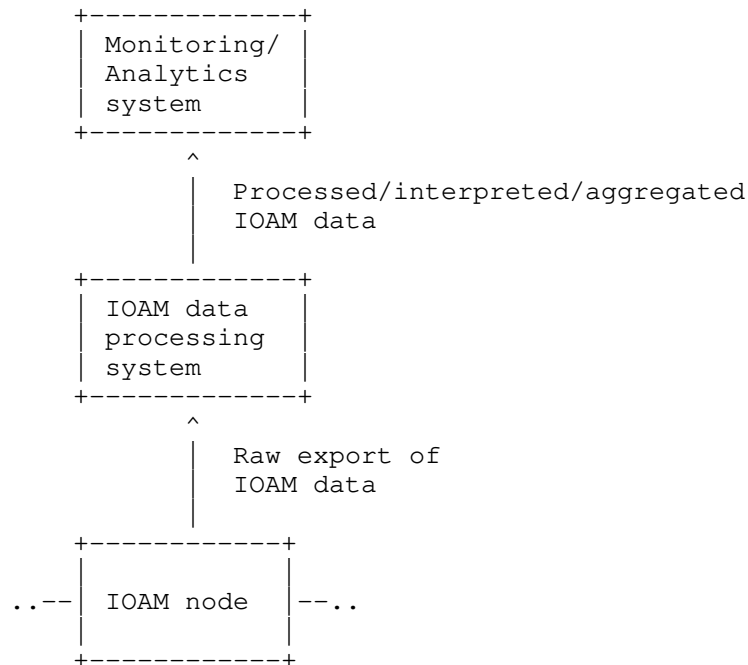
Table of Contents

1. Introduction	3
1.1. Requirements	5
1.2. Scope	5
2. Conventions	6
3. IPFIX for IOAM raw data export	6
3.1. Key IPFIX information elements leveraged for IOAM raw data export	6
3.2. New IPFIX information elements leveraged for IOAM raw data export	7
3.2.1. ioamReportFlags	7
3.2.2. ioamEncapsulationType	8
3.2.3. ioamPreallocatedTraceData	9
3.2.4. ioamIncrementalTraceData	9
3.2.5. ioamE2EData	10
3.2.6. ioamPOTData	10
3.2.7. ioamDirectExportData	11
3.2.8. ipHeaderPacketSectionWithPadding	11
3.2.9. ethernetFrameSection	12
4. Examples	13
4.1. Fixed Length IP Packet	13
4.2. Variable Length IP Packet (length < 255)	14
4.3. Variable Length IP Packet (length > 255)	15
4.4. Variable Length ETHERNET Packet (length < 255)	16
4.5. Variable Length IP Packet with Fixed Length IOAM Incremental Trace Data	17
4.6. Variable Length IP Packet with Variable Length IOAM Incremental Trace Data	18
5. IANA Considerations	19
6. Manageability Considerations	20
7. Security Considerations	20
8. Acknowledgements	20
9. References	20
9.1. Normative References	20
9.2. Informative References	21
Authors' Addresses	22

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. IOAM data fields are defined in [I-D.ietf-ippm-ioam-data]. This document discusses how In-situ Operations, Administration, and Maintenance (IOAM) information can be exported in raw format, i.e. uninterpreted format, from network devices to systems, such as monitoring or analytics systems using IPFIX [RFC7011].

"Raw export of IOAM data" refers to a mode of operation where a node exports the IOAM data as it is received in the packet. The exporting node neither interprets, aggregates nor reformats the IOAM data before it is exported. Raw export of IOAM data is to support an operational model where the processing and interpretation of IOAM data is decoupled from the operation of encapsulating/updating/decapsulating IOAM data, which is also referred to as IOAM data-plane operation. The figure below shows the separation of concerns for IOAM export: Exporting IOAM data is performed by the "IOAM node" which performs IOAM data-plane operation, whereas the interpretation of IOAM data is performed by the IOAM data processing system. The separation of concerns is to off-load interpretation, aggregation and formatting of IOAM data from the node which performs data-plane operations. In other words, a node which is focused on data-plane operations, i.e. forwarding of packets and handling IOAM data will not be tasked to also interpret the IOAM data, but can leave this task to another system. Note that for scalability reasons, a single IOAM node could choose to export IOAM data to several IOAM data processing systems.



IOAM node: IOAM encapsulating, IOAM decapsulating or IOAM transit node.

IOAM data processing system: System that receives raw IOAM data and provides for formatting, aggregation and interpretation of the IOAM data.

Monitoring/Analytics system: System that receives telemetry and other operational information from a variety of sources and provides for correlation and interpretation of the data received.

Raw export of IOAM data is typically generated by network devices at the edges of the network. Deployment and use-case dependent, such as in case of direct export [I-D.ietf-ippm-ioam-direct-export] or in cases where the operator is interested in dropped packets, raw export of IOAM data may be generated by IOAM transit nodes.

1.1. Requirements

Requirements for raw export of IOAM data:

- o Export all IOAM information contained in a packet.
- o Export a specific IOAM data type - Incremental Trace type, Preallocated Trace type, Proof of Transit type, Edge to Edge type, Direct Export type.
- o Export IOAM trace data associated with a packet, even if that data was never included in a transmitted or received packet in the network, for example in case of direct export.
- o Support coalescing of the IOAM data from multiple packets into a single raw export packet.
- o Support export of additional parts of the packet, other than the IOAM data as part of the raw export. This could be parts of the packet header and/or parts of the packet payload. This additional information provides context to the IOAM data (e.g. to be used for flow identification) and is to enable the IOAM data processing system to perform further analysis on the received data.
- o Report the reason why IOAM data was exported. The "reason for export" is to complement the IOAM data retrieved from the packet. For example, if a packet was dropped by a node due to congestion, it could be helpful to export the IOAM data of this dropped packet along with an indication that the packet that the IOAM data belongs to was dropped due to congestion.

1.2. Scope

This document discusses raw export of IOAM data using IPFIX.

The following is considered out of scope for this document:

- o Protocols other than IPFIX for raw export of IOAM data.
- o Interpretation or aggregation of IOAM data prior to exporting.
- o Configuration of network devices so that they can determine when to generate IOAM reports, and what information to include in those reports.
- o Events that trigger generation of IOAM reports.

- o Selection of particular destinations within distributed telemetry monitoring systems, to which IOAM reports will be sent.
- o Export format for flow statistics or processed/interpreted/aggregated IOAM data.

2. Conventions

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].

Abbreviations used in this document:

E2E: Edge to Edge

IOAM: In-situ Operations, Administration, and Maintenance

MTU: Maximum Transmit Unit

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

3. IPFIX for IOAM raw data export

IPFIX, being a generic export protocol, can export any Information Elements as long as they are described in the information model. The IPFIX protocol is well suited for and is defined as the protocol for exporting packet samples in [RFC5476].

IPFIX/PSAMP [RFC7011], [RFC5476] already define many of the information elements needed for exporting sections of packets needed for deriving context and raw IOAM data export. This document specifies extensions of the IPFIX information model for meeting the requirements in Section 1.1.

3.1. Key IPFIX information elements leveraged for IOAM raw data export

The existing IPFIX Information Elements that are required for IOAM raw data export are listed here. Their details are available in IANA's IPFIX registry [IANA-IPFIX].

The existing IPFIX Information Elements used to carry the sections of the packets including IOAM data within it are as follows:

313 - ipHeaderPacketSection

315 - dataLinkFrameSection

The following Information Elements will be used to provide context to the ipHeaderPacketSection and dataLinkFrameSection as described in [IANA-IPFIX]:

408 - dataLinkFrameType

409 - sectionOffset

410 - sectionExportedOctets

The following Information Element will be used to provide forwarding status of the flow and any attached reasons.

89 - forwardingStatus

3.2. New IPFIX information elements leveraged for IOAM raw data export

IOAM data raw export using IPFIX requires a set of new information elements which are described in this section.

3.2.1. ioamReportFlags

Description:

This Information Element describes properties associated with an IOAM report.

The ioamReportFlags data type is an 8-bit field. The following bits are defined here:

Bit 0 Dropped Association - Dropped packet of interest.

Bit 1 Congested Queue Association - Indicates the presence of congestion on a monitored queue.

Bit 2 Tracked Flow Association - Matched a flow of interest.

Bit 3-7 Reserved

IANA is requested to create a new subregistry for IOAM Report Flags and fill it with the initial list from the description. New assignments for IOAM Encapsulation Types are administered by IANA through Expert Review [RFC5226] i.e., review by one of a group of experts designated by an IETF Area Director.

Abstract Data Type: unsigned8

Data Type Semantics: flags

ElementId: TBD1

Status: current

3.2.2. ioamEncapsulationType

Description:

This Information Element specifies the type of encapsulation to interpret ioamPreallocatedTraceData, ioamIncrementalTraceData, ioamE2EData, ioamPOTData, ioamDirectExportData.

The following ioamEncapsulationType values are defined here:

- 0 None : IOAM data follows format defined in [I-D.ietf-ippm-ioam-data]
- 1 GRE : IOAM data follows format defined in [I-D.weis-ippm-ioam-eth]
- 2 IPv6 : IOAM data follows format defined in [I-D.ietf-ippm-ioam-ipv6-options]
- 3 VXLAN-GPE : IOAM data follows format defined in [I-D.brockners-ippm-ioam-vxlan-gpe]
- 4 GENEVE Option: IOAM data follows format defined in [I-D.brockners-ippm-ioam-geneve]
- 5 GENEVE Next Protocol: IOAM data follows format defined in [I-D.weis-ippm-ioam-eth]
- 6 NSH : IOAM data follows format defined in [I-D.ietf-sfc-ioam-nsh]

IANA is requested to create a new subregistry for IOAM Encapsulation Types and fill it with the initial list from the description. New assignments for IOAM Encapsulation Types are administered by IANA through Expert Review [RFC5226] i.e., review by one of a group of experts designated by an IETF Area Director.

Abstract Data Type: unsigned8

Data Type Semantics: identifier

ElementId: TBD2

Status: current

3.2.3. ioamPreallocatedTraceData

Description:

This Information Element carries n octets of IOAM Preallocated Trace data defined in [I-D.ietf-ippm-ioam-data].

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Preallocated Trace Option.

Abstract Data Type: octetArray

ElementId: TBD3

Status: current

3.2.4. ioamIncrementalTraceData

Description:

This Information Element carries n octets of IOAM Incremental Trace data defined in [I-D.ietf-ippm-ioam-data].

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Incremental Trace Option.

Abstract Data Type: octetArray

ElementId: TBD4

Status: current

3.2.5. ioamE2EData

Description:

This Information Element carries n octets of IOAM E2E data defined in [I-D.ietf-ippm-ioam-data].

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Edge-to-Edge Option.

Abstract Data Type: octetArray

ElementId: TBD5

Status: current

3.2.6. ioamPOTData

Description:

This Information Element carries n octets of IOAM POT data defined in [I-D.ietf-ippm-ioam-data].

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Proof of Transit Option.

Abstract Data Type: octetArray

ElementId: TBD6

Status: current

3.2.7. ioamDirectExportData

Description:

This Information Element carries n octets of IOAM Direct Export data defined in [I-D.ietf-ippm-ioam-direct-export].

In addition to the fields from the IOAM Direct Export Option header in the packet, this information element includes all of the trace data from the exporting node, based on the IOAM-Trace-Type value. This data is appended inside ioamDirectExportData following the bit order of the IOAM-Trace-Type field, similar to the way that IOAM encapsulating nodes append trace data in Incremental Trace Option headers.

The format of the data is determined by the ioamEncapsulationType information element, if present. When the ioamEncapsulationType information element is present and has a value other than "None", and with sufficient length, this element may also report octets from subsequent headers and payload. If no ioamEncapsulationType information element is present, then the encapsulation type shall be assumed to be "None" and this information element only contains octets from the IOAM Direct Export Option plus the corresponding trace data.

Abstract Data Type: octetArray

ElementId: TBD7

Status: current

3.2.8. ipHeaderPacketSectionWithPadding

Description:

This Information Element carries a series of n octets from the IP header of a sampled packet, starting sectionOffset octets into the IP header.

However, if no sectionOffset field corresponding to this Information Element is present, then a sectionOffset of zero applies, and the octets MUST be from the start of the IP header.

With sufficient length, this element also reports octets from the IP payload. However, full packet capture of arbitrary packet streams is explicitly out of scope per the Security Considerations sections of [RFC5477] and [RFC2804].

When this Information Element has a fixed length, this MAY include padding octets that are used to fill out that fixed length.

When this information element has a variable length, the variable length MAY include up to 3 octets of padding, used to preserve 4-octet alignment of subsequent Information Elements or subsequent records within the same set.

In either case of fixed or variable length, the amount of populated octets MAY be specified in the sectionExportedOctets field corresponding to this Information Element, in which case the remainder (if any) MUST be padding. If there is no sectionExportedOctets field corresponding to this Information Element, then all octets MUST be populated unless the total length of the IP packet is less than the fixed length of this Information Element, in which case the remainder MUST be padding.

Abstract Data Type: octetArray

ElementId: TBD8

Status: current

3.2.9. ethernetFrameSection

Description:

This Information Element carries a series of n octets from the IEEE 802.3 Ethernet frame of a sampled packet, starting after the preamble and start frame delimiter (SFD), plus sectionOffset octets into the frame if there is a sectionOffset field corresponding to this Information Element.

With sufficient length, this element also reports octets from the Ethernet payload. However, full packet capture of arbitrary packet streams is explicitly out of scope per the Security Considerations sections of [RFC5477] and [RFC2804].

When this Information Element has a fixed length, this MAY include padding octets that are used to fill out that fixed length.

When this information element has a variable length, the variable length MAY include up to 3 octets of padding, used to preserve 4-octet alignment of subsequent Information Elements or subsequent records within the same set.

In either case of fixed or variable length, the amount of populated octets MAY be specified in the sectionExportedOctets field

corresponding to this Information Element, in which case the remainder (if any) MUST be padding. If there is no sectionExportedOctets field corresponding to this Information Element, then all octets MUST be populated unless the total length of the Ethernet frame is less than the fixed length of this Information Element, in which case the remainder MUST be padding.

Abstract Data Type: octetArray

ElementId: TBD9

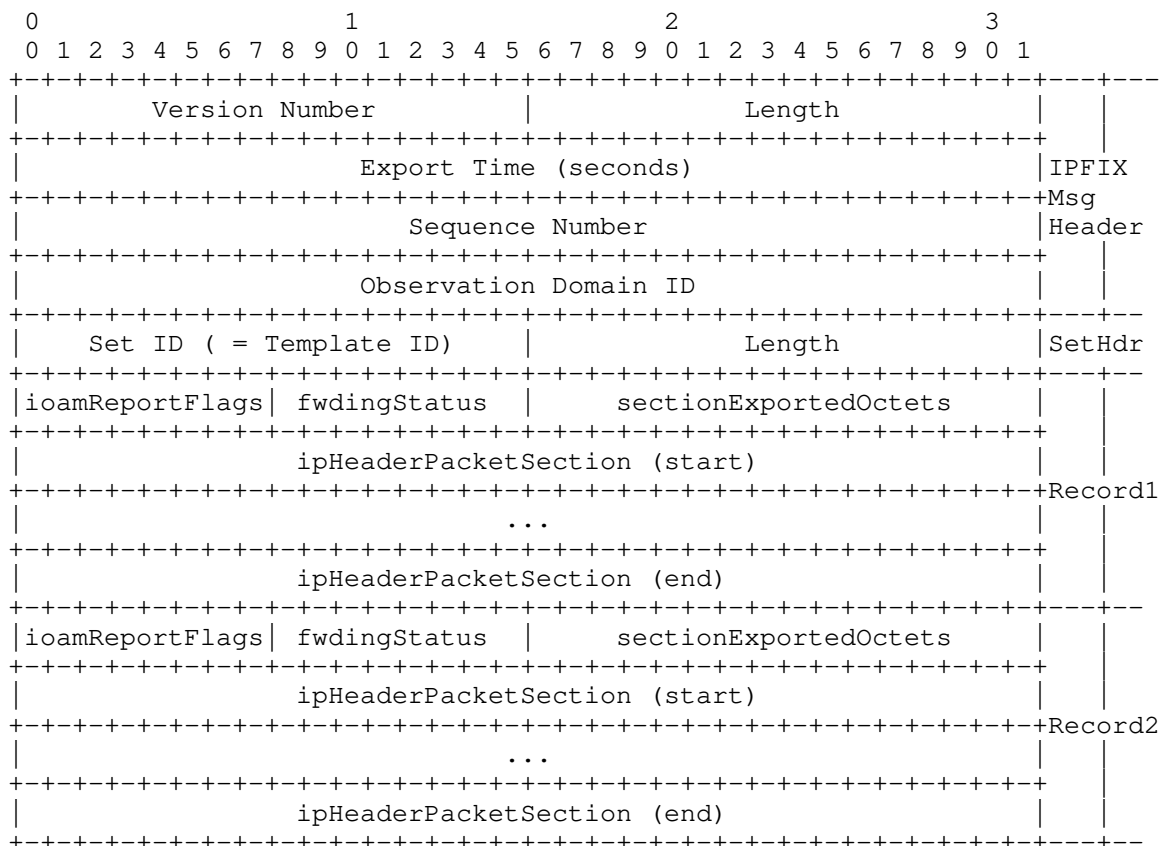
Status: current

4. Examples

This section shows a set of examples of how IOAM information along with other parts of the packet can be carried using IPFIX.

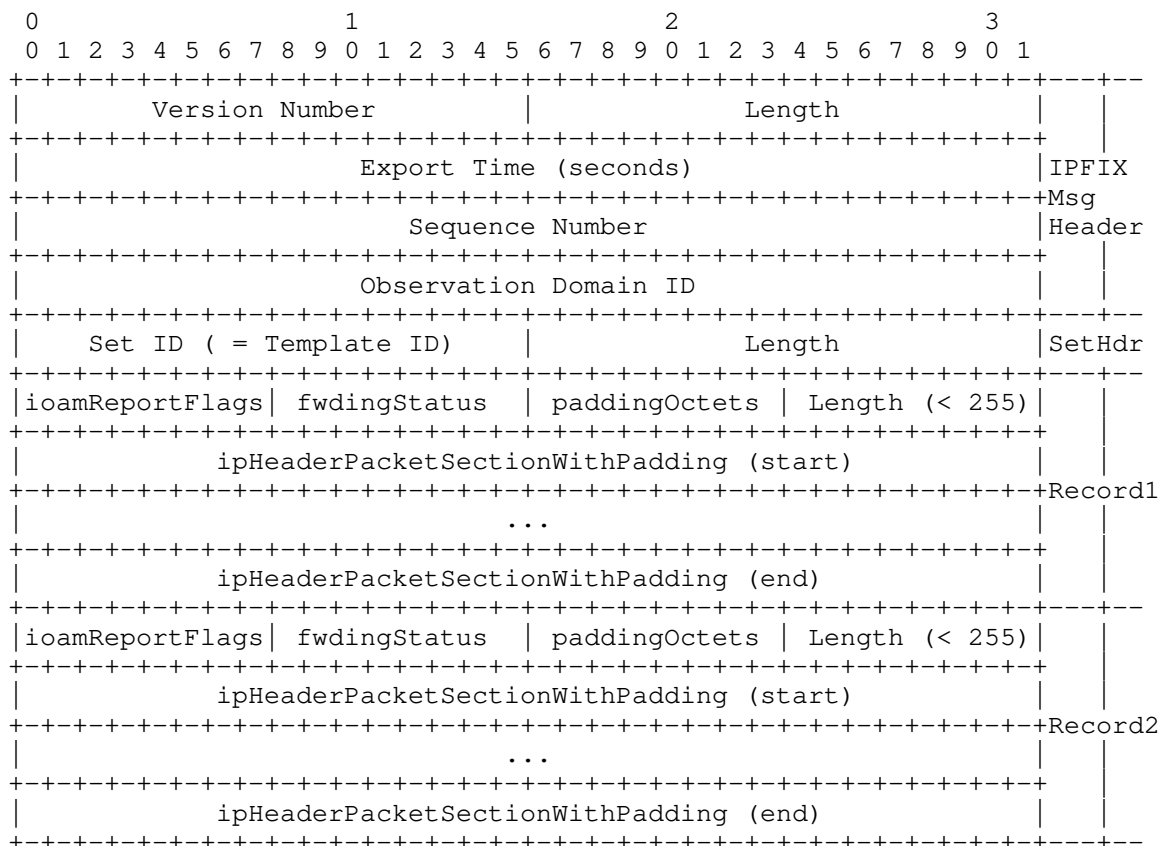
4.1. Fixed Length IP Packet

This example shows a fixed length IP packet. IOAM data is part of the ipHeaderPacketSection.



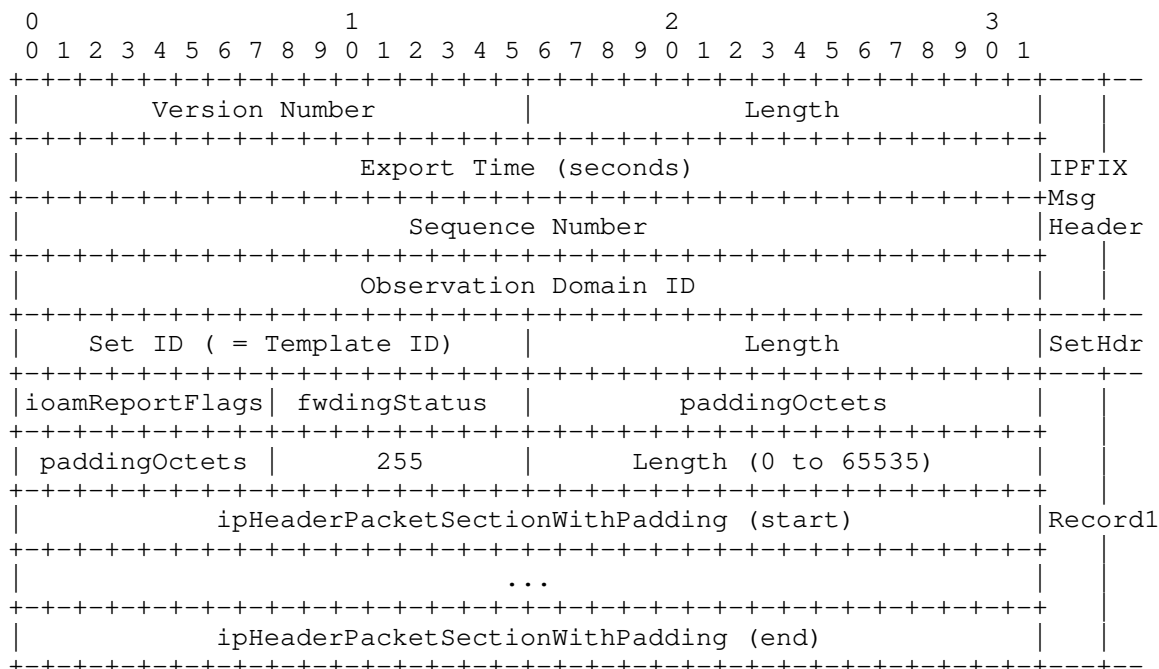
4.2. Variable Length IP Packet (length < 255)

This examples shows a variable length IP packet, with length < 255 bytes. IOAM data is part of the ipHeaderPacketSectionWithPadding.



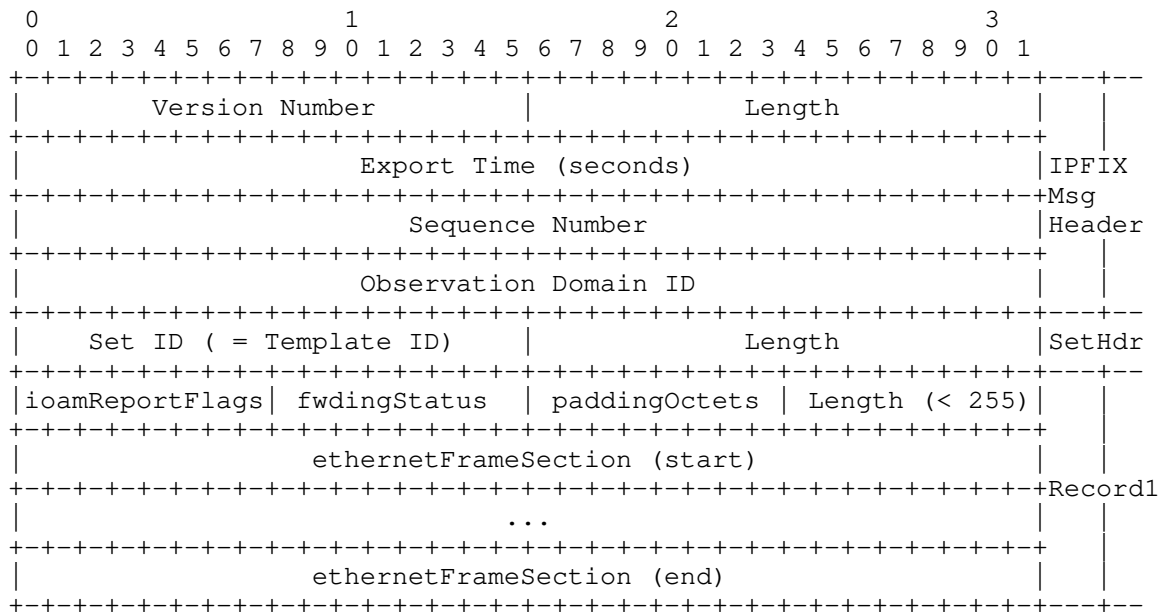
4.3. Variable Length IP Packet (length > 255)

This examples shows a variable length IP packet, with length > 255 bytes. IOAM data is part of the ipHeaderPacketSectionWithPadding.



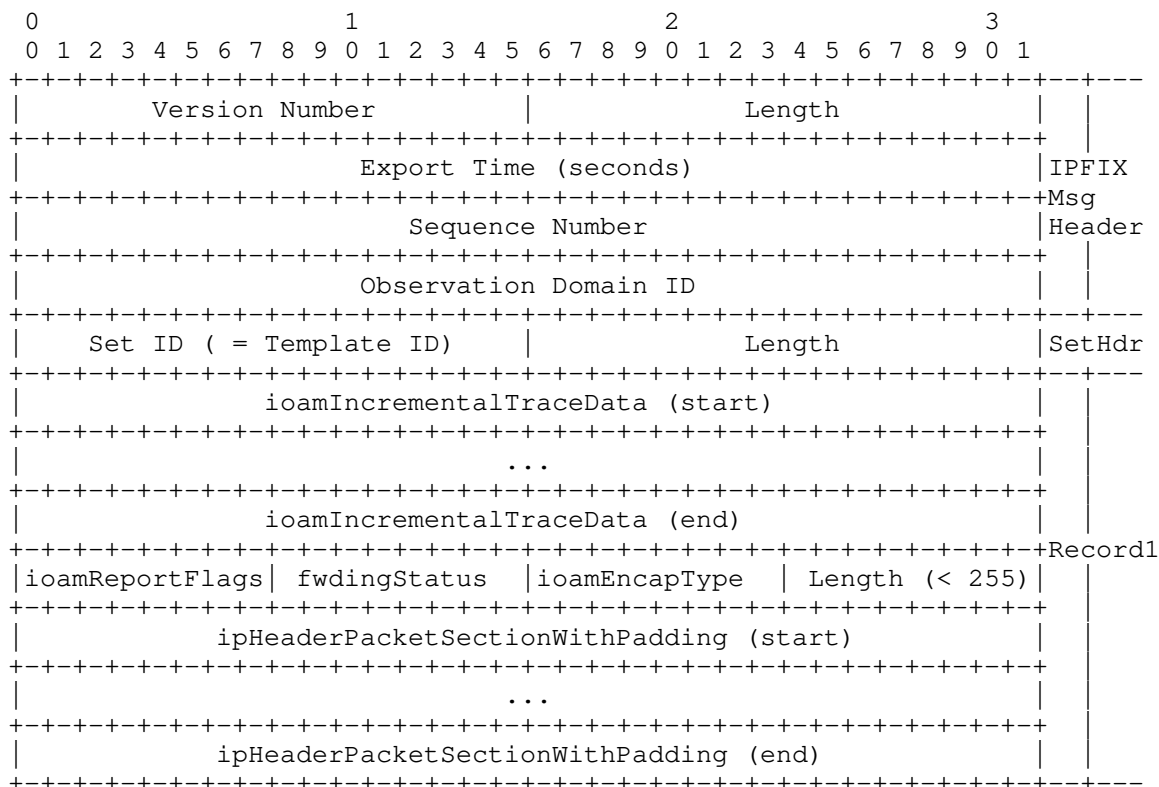
4.4. Variable Length ETHERNET Packet (length < 255)

This examples shows a variable length Ethernet packet, with length < 255 bytes. IOAM data is part of the ethernetFrameSection.



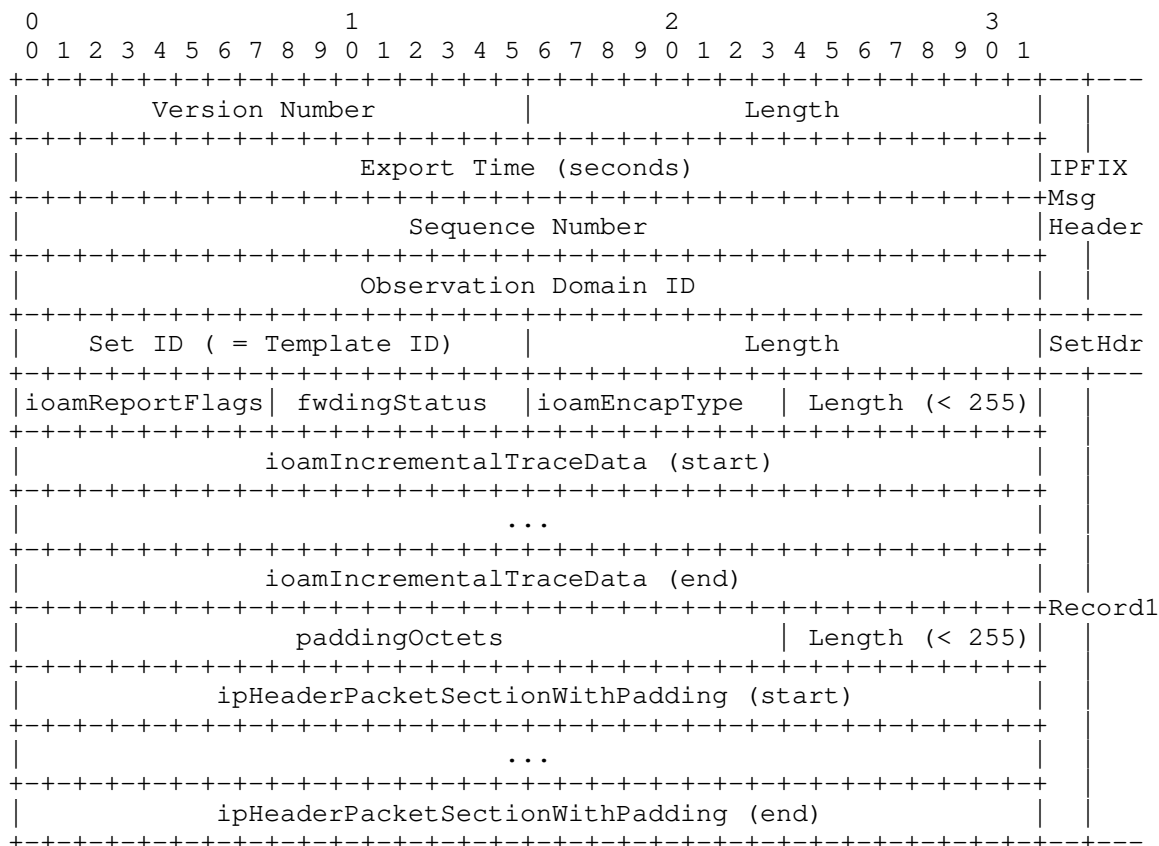
4.5. Variable Length IP Packet with Fixed Length IOAM Incremental Trace Data

This examples shows a variable length IP packet with length < 255 bytes and fixed length ioamIncrementalTraceData carried separately.



4.6. Variable Length IP Packet with Variable Length IOAM Incremental Trace Data

This examples shows a variable length IP packet with length < 255 bytes and variable length ioamIncrementalTraceData with length < 255 bytes carried separately.



5. IANA Considerations

IANA is requested to allocate code points for the following Information Elements in [IANA-IPFIX]:

TBD1 ioamReportFlags

TBD2 ioamEncapsulationType

TBD3 ioamPreallocatedTraceData

TBD4 ioamIncrementalTraceData

TBD5 ioamE2EData

TBD6 ioamPOTData

TBD7 ioamDirectExportData

TBD8 ipHeaderPacketSectionWithPadding

TBD9 ethernetFrameSection

See Section 3.2 for further details.

IANA is requested to create subregistries for ioamReportFlags defined in Section 3.2.1 and ioamEncapsulationType defined in Section 3.2.2.

6. Manageability Considerations

Manageability considerations will be addressed in a later version of this document.

7. Security Considerations

Security considerations will be addressed in a later version of this document.

8. Acknowledgements

The authors would like to thank Barak Gafni, Tal Mizrahi, Jennifer Lemon, and Aviv Kfir for their thoughts and comments on raw IOAM data export.

9. References

9.1. Normative References

- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-12 (work in progress), February 2021.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5476] Claise, B., Ed., Johnson, A., and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, DOI 10.17487/RFC5476, March 2009, <<https://www.rfc-editor.org/info/rfc5476>>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken,
"Specification of the IP Flow Information Export (IPFIX)
Protocol for the Exchange of Flow Information", STD 77,
RFC 7011, DOI 10.17487/RFC7011, September 2013,
<<https://www.rfc-editor.org/info/rfc7011>>.

9.2. Informative References

- [I-D.brockners-ippm-ioam-geneve]
Brockners, F., Bhandari, S., Govindan, V. P., Pignataro, C., Nainar, N. K., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Lapukhov, P., Gafni, B., Kfir, A., and M. Spiegel, "Geneve encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-geneve-05 (work in progress), November 2020.
- [I-D.brockners-ippm-ioam-vxlan-gpe]
Brockners, F., Bhandari, S., Govindan, V. P., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "VXLAN-GPE Encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-vxlan-gpe-03 (work in progress), November 2019.
- [I-D.ietf-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ietf-ippm-ioam-direct-export-07 (work in progress), October 2021.
- [I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S. and F. Brockners, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-07 (work in progress), February 2022.
- [I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-07 (work in progress), January 2022.
- [I-D.weis-ippm-ioam-eth]
Weis, B., Brockners, F., Hill, C., Bhandari, S., Govindan, V. P., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "EtherType Protocol Identification of In-situ OAM Data", draft-weis-ippm-ioam-eth-04 (work in progress), May 2020.

- [IANA-IPFIX] "IP Flow Information Export (IPFIX) Entities",
<<https://www.iana.org/assignments/ipfix/ipfix.xhtml>>.
- [RFC2804] IAB and IESG, "IETF Policy on Wiretapping", RFC 2804,
DOI 10.17487/RFC2804, May 2000,
<<https://www.rfc-editor.org/info/rfc2804>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an
IANA Considerations Section in RFCs", RFC 5226,
DOI 10.17487/RFC5226, May 2008,
<<https://www.rfc-editor.org/info/rfc5226>>.
- [RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G.
Carle, "Information Model for Packet Sampling Exports",
RFC 5477, DOI 10.17487/RFC5477, March 2009,
<<https://www.rfc-editor.org/info/rfc5477>>.

Authors' Addresses

Mickey Spiegel
Barefoot Networks, an Intel company
4750 Patrick Henry Drive
Santa Clara, CA 95054
US

Email: mickey.spiegel@intel.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Thoughtspot
3rd Floor, Indiqube Orion, 24th Main Rd, Garden Layout, HSR Layout
Bangalore, KARNATAKA 560 102
India

Email: shwetha.bhandari@thoughtspot.com

Ramesh Sivakolundu
Cisco Systems, Inc.
170 West Tasman Dr.
SAN JOSE, CA 95134
USA

Email: sramesh@cisco.com

Operations and Management Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2020

Q. Sun
H. Xu
China Telecom
B. Wu, Ed.
Q. Wu, Ed.
Huawei
C. Eckel, Ed.
Cisco Systems
July 3, 2019

A YANG Data Model for SD-WAN Service Delivery
draft-sun-opsawg-sdwan-service-model-04

Abstract

This document provides a YANG data model for an SD-WAN service. An SD-WAN service is a connectivity service offered by a service provider network to provide connectivity across different locations of a customer network or between a customer network and an external network, such as the Internet or a private/public cloud network. This connectivity is provided as an overlay constructed using one of more underlay networks. The model can be used by a service orchestrator of a service provider to request, configure, and manage the components of an SD-WAN service.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 4, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
1.1. Terminology	3
1.2. Definitions	3
2. High Level Overview of SD-WAN Service	4
3. Service Data Model Usage	6
4. Design of the Data Model	7
4.1. SD-WAN connectivity service	8
4.1.1. VPNs	8
4.1.2. Sites	9
4.2. Application based Policy Service	10
5. Modules Tree Structure	12
6. YANG Modules	17
7. Security Considerations	43
8. IANA Considerations	43
9. Appendix 1: Terminology Mapping between MEF SD-WAN Service Attributes and IETF SD-WAN model	44
10. Appendix 2: IETF OSE model vs IETF SD-WAN model	44
11. Acknowledgments	45
12. Contributors	45
13. References	45
13.1. Normative References	45
13.2. Informative References	46
Authors' Addresses	47

1. Introduction

An SD-WAN service is a connectivity service offered by a service provider network to provide connectivity across different locations of a customer network or between a customer network and an external network. Compared to a conventional PE-based connectivity service as defined in Layer 3 VPN Service Model [RFC8299] and Layer 2 VPN Service Model [RFC8466], an SD-WAN service is a CE-based connectivity service that uses the Internet or PE-based connectivity services as underlay connectivity services. More specially, an SD-WAN service is an overlay connectivity service that provides the flexibility of

adding, removing, or moving services without needing to change the underlay networks.

Besides being an overlay service, an SD-WAN Service has the following characteristics:

- o Hybrid WAN access: The CE could connect to a variety of Internet access technologies, including fiber, cable, DSL-based, WiFi, or 4G/Long Term Evolution (LTE), which implies wider reachability and shorter provisioning cycles. It can also use private VPN connectivity services defined in [RFC4364] and [RFC4664], or Operator Ethernet Services, as defined in [MEF51.1], to take advantage of better performance.
- o Application based traffic forwarding: There are diverse applications used in enterprises, such as VoIP calling, video conferencing, streaming media, etc. Application traffic across the WAN will be forwarded based on business priorities, SLA requirements, or other enterprise requirements.
- o Centralized service management: Subscribers of the service need to be provided a single point (such as a web portal) from which to dynamically add or modify services, such as configuring application policies, adding new sites, or adding new underlay connectivity services.

This draft specifies the SD-WAN service YANG model which is modelled from a customer perspective. The model parameters can be used as an input to automated control and configuration applications to manage SD-WAN services.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [RFC2119].

1.2. Definitions

CE Device: Customer Edge Device , as per Provider Provisioned VPN Terminology [RFC4026] .

CE-based VPN: Refers to Provider Provisioned VPN Terminology [RFC4026]

PE Device: Provider Edge Device, as per Provider Provisioned VPN Terminology [RFC4026]

PE-Based VPNs: Refers to Provider Provisioned VPN Terminology [RFC4026]

SD-WAN: An automated, programmatic approach to managing enterprise network connectivity and circuit usage. It extends software-defined networking (SDN) into an application that businesses can use to quickly create a hybrid WAN, which comprises business-grade IP VPN, broadband Internet, and wireless services or multiple WANs of the same or different types. SD-WAN is also deemed as extended CE-based VPN.

SD-WAN Controller: Refers to the abstract entity that combines Control Plane (CP) and Management Plane (MP) defined in SDN: Layers and Architecture Terminology [RFC7426], to configure, manage and control the CEs and other corresponding SD-WAN components.

Underlay network: A network that provides connectivity across SD-WAN sites and over which customer network packets are tunnelled. An underlay network does not need to be aware that it is carrying overlay customer network packets. Addresses on an underlay network appear as "outer addresses" in encapsulated overlay packets. In general, an underlay network can use a completely different protocol (and address family) from that of the overlay network.

Overlay network: A virtual network in which the separation of customer networks is hidden from the underlying physical infrastructure. That is, the underlying transport networks do not need to know about customer separation to correctly forward traffic. IPsec tunnels [RFC6071] are an example of an L3 overlay network.

2. High Level Overview of SD-WAN Service

From a customer perspective, an example of SD-WAN service network is shown in figure 1.

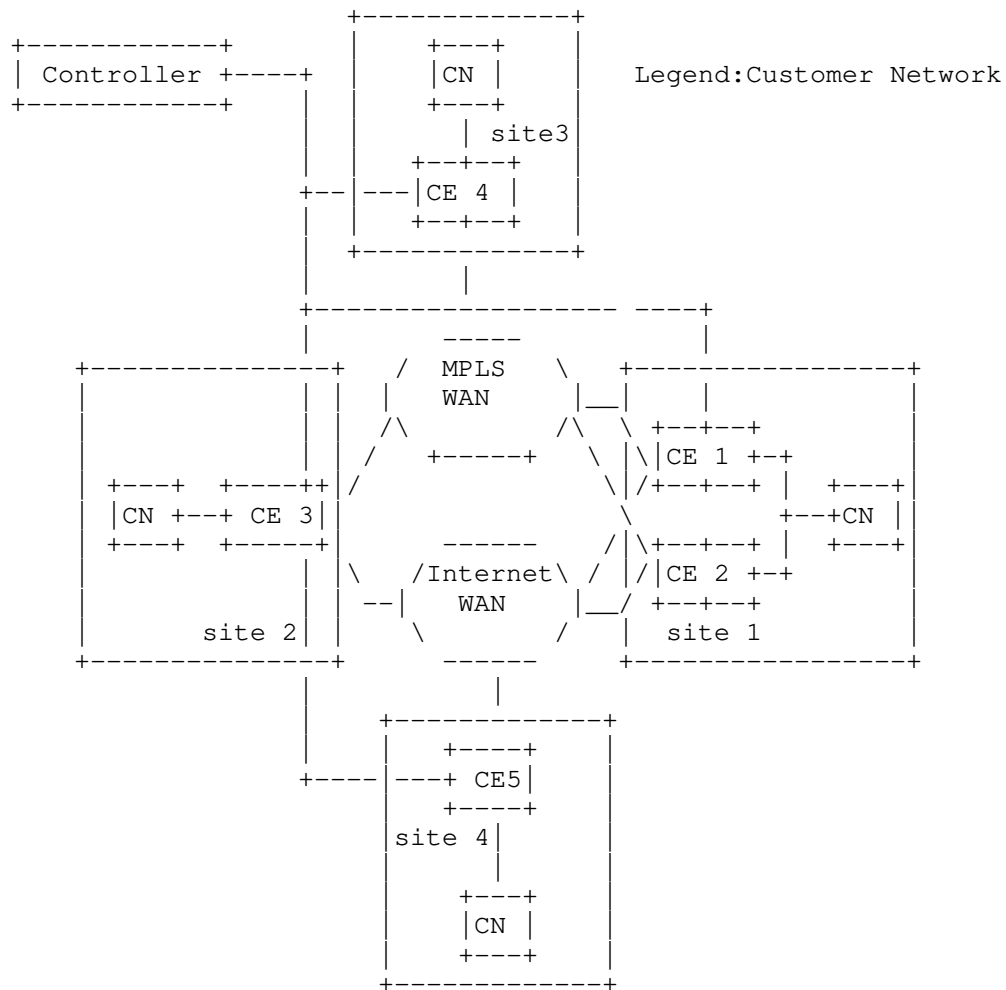


figure 1 SD-WAN network example

As shown in figure 1, the SD-WAN network consists of a number of sites, which are connected through Internet or MPLS VPN.

Within each site, a CE is connected with customer's network on one side, and is also connected to Internet, or to private WAN, or to both on the other side. The customer network could be an L2 or L3 network. For the WAN side, Internet provides ubiquitous IP connectivity via access network like Broadband access or LTE access, while MPLS WAN, like conventional VPN, provides secure and committed connectivity. The boundary between the customer and the service provider is between customer node and the CE device.

Additionally, a site could deploy one or more CEs to improve availability.

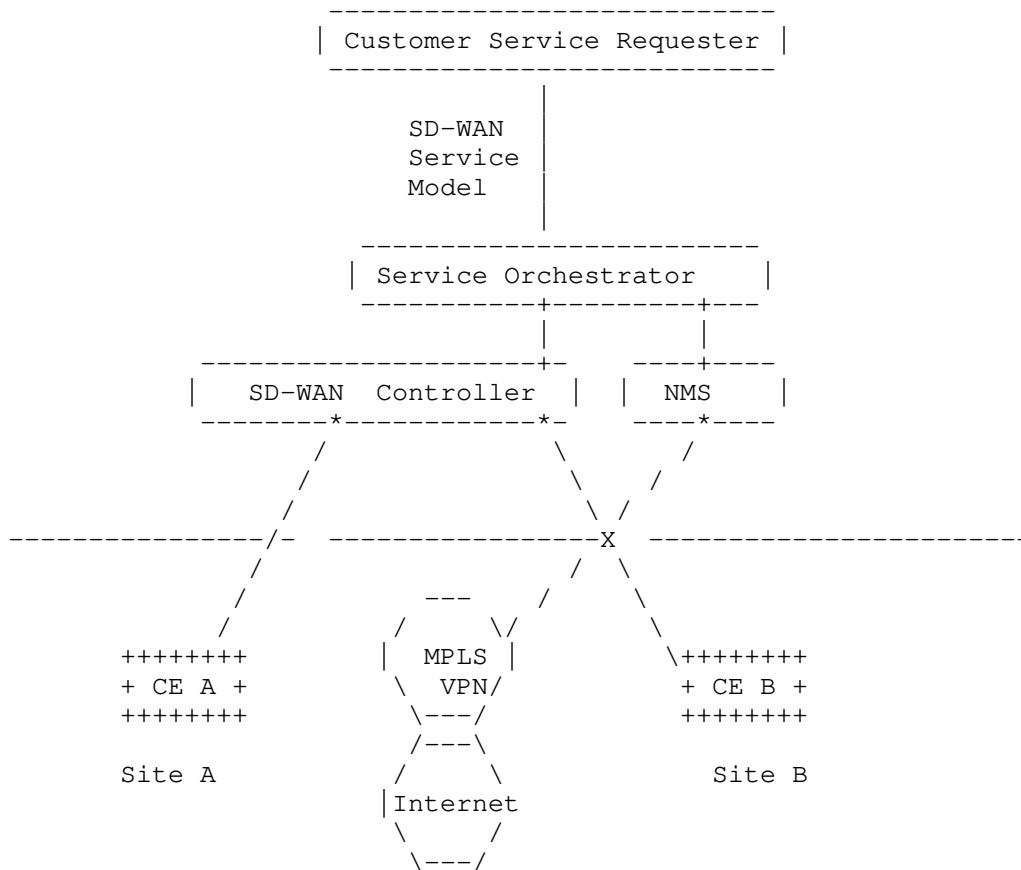
The controller is a centralized entity that manages all the CEs involved in the SD-WAN. The controller could provide bootstrapping of the CEs, ongoing CE configuration, and establishment of secured tunnels between CEs to support the SD-WAN service and application policy enforcement. Various IP tunnelling options (e.g., GRE [RFC2784] and IPSec [RFC6071]), could be used depending on whether traffic from the site is across underlying private VPN or public Internet, and the specific definition is out of scope of this document.

Besides basic connectivity between the sites, the SD-WAN service could be extended by providing direct Internet connectivity, cloud network connectivity, or conventional MPLS VPN interoperability.

3. Service Data Model Usage

The SD-WAN service model provides an abstracted interface to request, configure, and manage the components of an SD-WAN service.

A typical usage for this model is as an input to a service orchestrator that is responsible for service management. Based on the user's service request, the service orchestrator can instruct the SD-WAN controller to add a new site, VPN or application policy in real-time. The orchestrator could orchestrate the other network, such as legacy MPLS VPN network to interconnect with SD-WAN network where Layer 2 VPN Service Mode [RFC8466] or Layer 3 VPN Service Model [RFC8299] could be used.



Reference Architecture for the Use of SD-WAN Service Model Usage

For an SD-WAN to be established under the SP's control, the customer informs the Service Provider of which sites should become part of the requested service and what types of policy will provide. And then the SP configures and updates the service base on the service model and the available resources derived from the SD-WAN controller, and then provisions and manages the customer's service through the SD-WAN controller. How the SD-WAN controller to control and manage the CEs is out of scope of the document.

4. Design of the Data Model

An SD-WAN service consist of two service components:

1. SD-WAN connectivity service

2. SD-WAN application policy service

4.1. SD-WAN connectivity service

SD-WAN connectivity service is the basic component of the SD-WAN service that represents a virtual connection between two or more customer sites. In this model, each virtual connection is defined as a VPN. Each customer can have one or more VPNs, and each VPN can be established between a subset of sites. The association of sites and VPNs is modelled by VPN endpoints.

4.1.1. VPNs

The "sdwan-vpn" list item contains service parameters that apply to an SD-WAN VPN. These parameters are specified as follows:

- o The "vpn-id" leaf is under the vpn-service list, and provides a unique ID for a VPN.
- o The "endpoints" list is under the vpn-service list. Each "endpoint" is a logical point associated with a site. The two main functions of the endpoint are the association of a VPN with a site and per site application based policy enforcement.
- o The "topology" leaf is under the vpn-service list, which refers to a specific topology of the VPN service. Different VPN connection topology can be used. For a VPN with a few sites, simple topologies such as hub-and-spoke or full-mesh can be used. For a large VPN, a hierarchical topology may be taken.
- o The "performance-objectives" container specifies the performance-related properties of an SD-WAN VPN that can be measured. System uptime is the only performance objective defined currently. It indicates the proportion of time, during a given time period that the service is working from the customer perspective. Three parameters are defined, including the start time of the evaluation, the time interval of the evaluation, and the service uptime defined by a percentage.
- o The "reserved-prefixes" container specifies the IP Prefixes that need to be reserved for Service Provider management purposes, such as diagnostics, so as to ensure they are not overlapping with IP Prefixes used by the customer network.

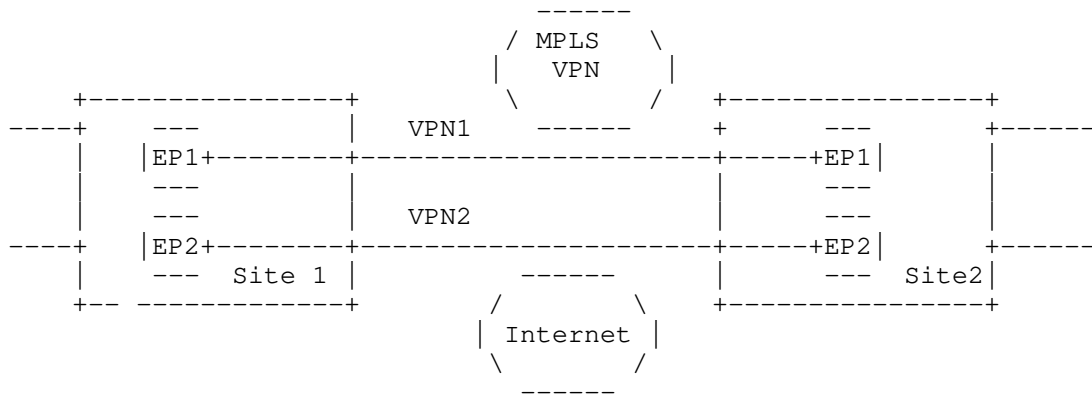


figure 3 SD-WAN VPN example

4.1.2. Sites

A site represents a customer office located at a specific geographic location. The "sites" container specifies the following parameters:

- o "site-id: uniquely identifies the site within the overall network infrastructure.
- o "device" specifies the device type (physical or virtual device) and the number of the devices.
- o "lan-accesses": Specifies the customer network access link parameters. A "site" is composed of at least one "lan-access" where one or more subnets can reside. The "lan-access" consists of the following categories of parameters:
 - * "bearer": defines requirements of the attachment (below Layer 3), bearer type including Ethernet, etc.
 - * IP Connection: defines Layer 3 parameters of the attachment, including IPv4 connection parameters and IPv6 connection parameters.
- o "wan-accesses": Specifies the WAN access link parameters. A "site" is composed of at least one "wan-access". The WAN access can be further specified by access type, service provider name, and bandwidth of the WAN connectivity. The "wan-access" consists of the following categories of parameters:
 - * "access-type": specifies whether the access is Broadband Internet, Wireless Internet or private circuit.

- * "access-provider": specifies the service provider name.
- * bandwidth: specifies the WAN link bandwidth including input and output bandwidth.
- * "bearer": defines requirements of the attachment (below Layer 3), bearer type including Ethernet, etc.
- * IP Connection: defines Layer 3 parameters of the attachment, including IPv4 connection parameters and IPv6 connection parameters.

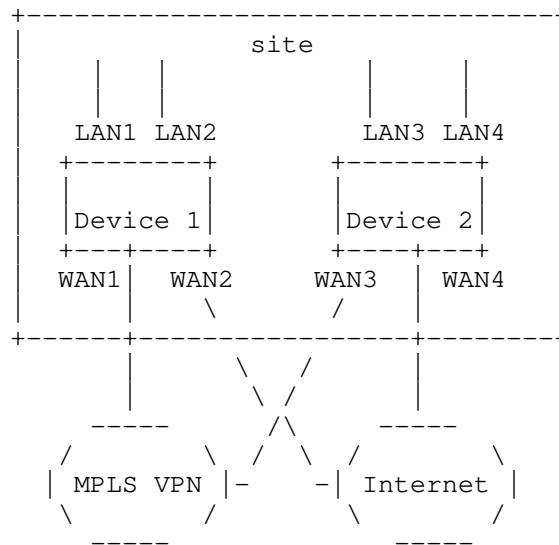


figure 4 Site example

4.2. Application based Policy Service

The connectivity service establishes a virtual connection for the enterprise network, and the Application based Policy Service is designed to ensure business-critical and real-time application experience while also ensuring the security and corporate policies.

Typically, application policies common to each VPN can be defined and then enforced when traffic from a customer's network at a particular site is sent over the WAN.

The application policy assignment is defined under the VPN endpoint container to specify the mapping of application flow name or application group name and their associated policy list names. If an

application flow and the application flow group in which the Application Flow is a member are both assigned a policy at an VPN End Point, the policy assigned to the application flow will supersede the group policy.

The application policy per VPN consists of three lists under the VPN container:

- o application flow list: Describes the characteristics of an enterprise application and is used to identify applications, e.g., based on layer 3 source and destination addresses, layer 4 ports, layer 4 protocol, etc.
- o application group list: Describes application flow aggregation, which is used to deliver aggregation policies, such as bandwidth restrictions for a group of applications.
- o policy list: Defines the application's policy set. Since SD-WAN has more than one WAN connectivity and various encrypted or unencrypted overlay tunnels, there could be multiple tunnel or link selection combination. In this model, different path selection policies are combined to meet different needs based on application SLA, security, cost, and so on. For example, when different applications in a branch need to pass over the WAN, according to the application-aware policy requirements and the IP forwarding table, the Internet application or the SaaS application can be accessed through the Internet, and the data center FTP application can use the Internet encrypted tunnel as the primary path, and the tunnel could only be over broadband Internet instead of wireless internet. This policy combination is not an exhaustive list and could be augmented according to business needs.

An example of a classification of application flows is as follows:

The HTTP traffic from the 192.0.2.0/24 LAN destined for port 80 will be classified in app-id 1.

The FTP traffic from the 192.0.2.0/24 LAN destined for 203.0.113.1/32 will be classified in app-id 2.

An example of a policy list is as follows:

```
"policy": [  
  {  
    "policy-id": "pol-a",  
    "policy-package":  
      {  
        "encryption": "false",  
        "internet-breakout": "true"  
        "public-private": "public",  
        "billing-method": "flat-only"  
        "backup": "false",  
        "bandwidth": "20", "50"  
      }  
    },  
  {  
    "policy-id": "pol-b",  
    "policy-package":  
      {  
        "encryption": "true",  
        "internet-breakout": "false"  
        "public-private": "public",  
        "billing-method": "flat-only"  
        "backup": "false",  
        "bandwidth": "50", "none"  
      }  
    }  
  ]
```

An example of an application policy list is as follows:

```
"app-policy": [  
  {  
    "app-id": "1"  
    "policy-id": "pol-a",  
  },  
  {  
    "app-id": "1"  
    "policy-id": "pol-b",  
  }  
]
```

5. Modules Tree Structure

This document defines an SD-WAN service YANG data model.

```
module: ietf-sdwan-svc  
  +--rw sdwan-svc  
    +--rw vpn-services  
      | +--rw vpn-service* [vpn-id]
```

```

+--rw vpn-id          svc-id
+--rw topology?       identityref
+--rw performance-objective
|   +--rw start-time?   yang:date-and-time
|   +--rw duration?     string
|   +--rw uptime-objective
|       +--rw duration?  decimal64
+--rw reserved-prefixes
|   +--rw prefix*       inet:ip-prefix
+--rw application* [app-id]
|   +--rw app-id        svc-id
|   +--rw ac* [name]
|       +--rw name                string
|       +--rw (match-type)?
|           +--:(match-flow)
|               +--rw match-flow
|                   +--rw ethertype?    uint16
|                   +--rw cvlan?        uint8
|                   +--rw ipv4-src-prefix?  inet:ipv4-prefix
|                   +--rw ipv4-dst-prefix?  inet:ipv4-prefix
|                   +--rw l4-src-port?    inet:port-number
|                   +--rw l4-dst-port?    inet:port-number
|                   +--rw ipv6-src-prefix?  inet:ipv6-prefix
|                   +--rw ipv6-dst-prefix?  inet:ipv6-prefix
|                   +--rw protocol-field?  union
|           +--:(match-application)
|               +--rw match-application?  identityref
+--rw application-group* [app-group-id]
|   +--rw app-group-id    svc-id
|   +--rw app-id*         -> ../../application/app-id
+--rw policy* [policy-id]
|   +--rw policy-id       svc-id
|   +--rw policy-package
|       +--rw encryption?  enumeration
|       +--rw public-private?  enumeration
|       +--rw local-breakout?  boolean
|       +--rw billing-method?  enumeration
|       +--rw backup-path?    enumeration
|       +--rw bandwidth
|           +--rw commit?    uint32
|           +--rw max?       uint32
+--rw endpoints* [endpoint-id]
|   +--rw endpoint-id      svc-id
|   +--rw site-role?       identityref
|   +--rw site-attachment
|       |   +--rw site-id?  -> /sdwan-svc/sites/site/site-id
+--rw endpoint-policy-map
|   +--rw app-group-policy* [app-group-id]

```

```

|         | +--rw app-group-id    leafref
|         | +--rw policy-id?     leafref
+--rw app-policy* [app-id]
|         | +--rw app-id        leafref
|         | +--rw policy-id?    leafref
+--rw sites
  +--rw site* [site-id]
    +--rw site-id      svc-id
    +--rw device* [name]
      | +--rw name      string
      | +--rw type?    identityref
    +--rw lan-access* [name]
      | +--rw name      string
      | +--rw l2-technology
      |   +--rw l2-type? identityref
      |   +--rw untagged-interface
      |     | +--rw speed?  uint32
      |     | +--rw mode?   neg-mode
      |   +--rw tagged-interface
      |     | +--rw type?   identityref
      |     | +--rw dot1q-vlan-tagged
      |     |   | +--rw tg-type? identityref
      |     |   | +--rw cvlan-id  uint16
      |     |   +--rw priority-tagged
      |     |     | +--rw tag-type? identityref
      |     +--rw l2-mtu?      uint32
    +--rw ip-connection
      +--rw ipv4
        | +--rw address-allocation-type? identityref
        | +--rw dhcp
        |   | +--rw primary-subnet
        |   |   | +--rw ip-prefix?
        |   |   |   | inet:ipv4-prefix
        |   |   +--rw default-router? inet:ip-address
        |   |   +--rw provider-addresses*
        |   |   |   | inet:ipv4-address
        |   |   +--rw subscriber-address? inet:ip-address
        |   |   +--rw reserved-ip-prefix* inet:ip-prefix
        |   +--rw secondary-subnet* [ip-prefix]
        |     | +--rw ip-prefix
        |     |   | inet:ipv4-prefix
        |     +--rw provider-addresses*
        |     |   | inet:ipv4-address
        |     +--rw reserved-ip-prefix*
        |           | inet:ipv4-prefix
      +--rw static
        | +--rw primary-subnet
        |   | +--rw ip-prefix?

```

```

|         inet:ipv4-prefix
|         +--rw default-router?          inet:ip-address
|         +--rw provider-addresses*
|         |         inet:ipv4-address
|         +--rw subscriber-address?      inet:ip-address
|         +--rw reserved-ip-prefix*      inet:ip-prefix
+--rw secondary-subnet* [ip-prefix]
|         +--rw ip-prefix
|         |         inet:ipv4-prefix
|         +--rw provider-addresses*
|         |         inet:ipv4-address
|         +--rw reserved-ip-prefix*
|         |         inet:ipv4-prefix
+--rw ipv6
|         +--rw address-allocation-type? identityref
|         +--rw dhcp
|         |         +--rw subnet* [ip-prefix]
|         |         |         +--rw ip-prefix
|         |         |         |         inet:ipv6-prefix
|         |         +--rw provider-addresses*
|         |         |         inet:ipv6-address
|         |         +--rw reserved-ip-prefix*
|         |         |         inet:ipv6-prefix
|         +--rw slaac
|         |         +--rw subnet* [ip-prefix]
|         |         |         +--rw ip-prefix
|         |         |         |         inet:ipv6-prefix
|         |         +--rw provider-addresses*
|         |         |         inet:ipv6-address
|         |         +--rw reserved-ip-prefix*
|         |         |         inet:ipv6-prefix
|         +--rw static
|         |         +--rw subnet* [ip-prefix]
|         |         |         +--rw ip-prefix
|         |         |         |         inet:ipv6-prefix
|         |         +--rw provider-addresses*
|         |         |         inet:ipv6-address
|         |         +--rw reserved-ip-prefix*
|         |         |         inet:ipv6-prefix
|         +--rw subscriber-address?      inet:ipv6-address
+--rw wan-access* [name]
|         +--rw name                      string
|         +--rw access-type?              identityref
|         +--rw access-provider?          string
|         +--rw bandwidth
|         |         +--rw input-bandwidth? uint64
|         |         +--rw output-bandwidth? uint64
+--rw l2-technology

```

```

+--rw l2-type?                identityref
+--rw untagged-interface
|   +--rw speed?      uint32
|   +--rw mode?       neg-mode
+--rw tagged-interface
|   +--rw type?                identityref
|   +--rw dot1q-vlan-tagged
|   |   +--rw tg-type?        identityref
|   |   +--rw cvlan-id        uint16
|   +--rw priority-tagged
|   |   +--rw tag-type?        identityref
+--rw l2-mtu?                  uint32
+--rw ip-connection
+--rw ipv4
|   +--rw address-allocation-type?  identityref
+--rw dhcp
|   +--rw primary-subnet
|   |   +--rw ip-prefix?
|   |   |   inet:ipv4-prefix
|   |   +--rw default-router?      inet:ip-address
|   |   +--rw provider-addresses*
|   |   |   inet:ipv4-address
|   |   +--rw subscriber-address?  inet:ip-address
|   |   +--rw reserved-ip-prefix*  inet:ip-prefix
+--rw secondary-subnet* [ip-prefix]
|   +--rw ip-prefix
|   |   inet:ipv4-prefix
+--rw provider-addresses*
|   inet:ipv4-address
+--rw reserved-ip-prefix*
|   inet:ipv4-prefix
+--rw static
+--rw primary-subnet
|   +--rw ip-prefix?
|   |   inet:ipv4-prefix
+--rw default-router?      inet:ip-address
+--rw provider-addresses*
|   inet:ipv4-address
+--rw subscriber-address?  inet:ip-address
+--rw reserved-ip-prefix*  inet:ip-prefix
+--rw secondary-subnet* [ip-prefix]
|   +--rw ip-prefix
|   |   inet:ipv4-prefix
+--rw provider-addresses*
|   inet:ipv4-address
+--rw reserved-ip-prefix*
|   inet:ipv4-prefix
+--rw ipv6

```



```

+--rw address-allocation-type?  identityref
+--rw dhcp
|   +--rw subnet* [ip-prefix]
|       +--rw ip-prefix
|           |   inet:ipv6-prefix
|       +--rw provider-addresses*
|           |   inet:ipv6-address
|       +--rw reserved-ip-prefix*
|           |   inet:ipv6-prefix
+--rw slaac
|   +--rw subnet* [ip-prefix]
|       +--rw ip-prefix
|           |   inet:ipv6-prefix
|       +--rw provider-addresses*
|           |   inet:ipv6-address
|       +--rw reserved-ip-prefix*
|           |   inet:ipv6-prefix
+--rw static
|   +--rw subnet* [ip-prefix]
|       +--rw ip-prefix
|           |   inet:ipv6-prefix
|       +--rw provider-addresses*
|           |   inet:ipv6-address
|       +--rw reserved-ip-prefix*
|           |   inet:ipv6-prefix
+--rw subscriber-address?  inet:ipv6-address

```

6. YANG Modules

<CODE BEGINS> file "ietf-sdwan-svc@2019-06-06.yang"

```

module ietf-sdwan-svc {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-sdwan-svc";
  prefix sdwan-svc;

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

  organization
    "IETF foo Working Group.";
  contact
    "WG List: foo@ietf.org
     Editor:  ";

```

```
description
  "The YANG module defines a generic service configuration
  model for Managed SD-WAN.";

revision 2019-06-06 {
  description
    "Initial revision";
  reference "A YANG Data Model for SD-WAN service.";
}

typedef svc-id {
  type string;
  description
    "Type definition for service identifier";
}

typedef address-family {
  type enumeration {
    enum ipv4 {
      description
        "IPv4 address family.";
    }
    enum ipv6 {
      description
        "IPv6 address family.";
    }
  }
  description
    "Defines a type for the address family.";
}

typedef neg-mode {
  type enumeration {
    enum full-duplex {
      description
        "Defining Full duplex mode";
    }
    enum auto-neg {
      description
        "Defining Auto negotiation mode";
    }
  }
  description
    "Defining a type of the negotiation mode";
}

typedef device-type {
  type enumeration {
```

```
    enum physical {
        description
            "Physical device";
    }
    enum virtual {
        description
            "Virtual device";
    }
}
description
    "Defines device types.";
}

identity device-type {
    description
        "Base identity for device type.";
}

identity virtual-ce {
    base device-type;
    description
        "Identity for virtual-ce.";
}

identity physical-ce {
    base device-type;
    description
        "Identity for physical-ce.";
}

identity customer-application {
    description
        "Base identity for customer application.";
}

identity web {
    base customer-application;
    description
        "Identity for Web application (e.g., HTTP, HTTPS).";
}

identity mail {
    base customer-application;
    description
        "Identity for mail application.";
}

identity file-transfer {
```

```
    base customer-application;
    description
        "Identity for file transfer application (e.g., FTP, SFTP).";
}

identity database {
    base customer-application;
    description
        "Identity for database application.";
}

identity social {
    base customer-application;
    description
        "Identity for social-network application.";
}

identity games {
    base customer-application;
    description
        "Identity for gaming application.";
}

identity p2p {
    base customer-application;
    description
        "Identity for peer-to-peer application.";
}

identity network-management {
    base customer-application;
    description
        "Identity for management application
        (e.g., Telnet, syslog, SNMP).";
}

identity voice {
    base customer-application;
    description
        "Identity for voice application.";
}

identity video {
    base customer-application;
    description
        "Identity for video conference application.";
}
```

```
identity eth-inf-type {
  description
    "Identity of the Ethernet interface type.";
}

identity tagged {
  base eth-inf-type;
  description
    "Identity of the tagged interface type.";
}

identity untagged {
  base eth-inf-type;
  description
    "Identity of the untagged interface type.";
}

identity lag {
  base eth-inf-type;
  description
    "Identity of the LAG interface type.";
}

identity tag-type {
  description
    "Base identity from which all tag types
    are derived from";
}

identity c-vlan {
  base tag-type;
  description
    "A Customer-VLAN tag, normally using the 0x8100
    Ethertype";
}

identity tagged-inf-type {
  description
    "Identity for the tagged
    interface type.";
}

identity dot1q {
  base tagged-inf-type;
  description
    "Identity for dot1q vlan tagged interface.";
}
```

```
identity priority-tagged {
  base tagged-inf-type;
  description
    "This identity the priority-tagged interface.";
}

identity vpn-topology {
  description
    "Base identity for vpn topology.";
}

identity any-to-any {
  base vpn-topology;
  description
    "Identity for any-to-any VPN topology.";
}

identity hub-spoke {
  base vpn-topology;
  description
    "Identity for Hub-and-Spoke VPN topology.";
}

identity site-role {
  description
    "Site Role in a VPN topology ";
}

identity any-to-any-role {
  base site-role;
  description
    "Site in an any-to-any IP VPN.";
}

identity hub {
  base site-role;
  description
    "Hub Role in Hub-and-Spoke IP VPN.";
}

identity spoke {
  base site-role;
  description
    "Spoke Role in Hub-and-Spoke IP VPN.";
}

identity access-type {
  description
```

```
    "Access type of a site in a connection to different WAN";
}

identity commodity {
    base access-type;
    description
        "Internet access";
}

identity cellular {
    base access-type;
    description
        "Refers to a subset of 3G/4G/LTE and 5G";
}

identity private {
    base access-type;
    description
        "Refers to private circuits such as Ethernet, T1, etc";
}

identity routing-protocol-type {
    description
        "Base identity for routing protocol type.";
}

identity ospf {
    base routing-protocol-type;
    description
        "Identity for OSPF protocol type.";
}

identity bgp {
    base routing-protocol-type;
    description
        "Identity for BGP protocol type.";
}

identity static {
    base routing-protocol-type;
    description
        "Identity for static routing protocol type.";
}

identity address-allocation-type {
    description
        "Base identity for address-allocation-type for PE-CE link.";
}
```

```
identity dhcp {
  base address-allocation-type;
  description
    "Provider network provides DHCP service to customer.";
}

identity static-address {
  base address-allocation-type;
  description
    "Provider-to-customer addressing is static.";
}

identity slaac {
  base address-allocation-type;
  description
    "Use IPv6 SLAAC.";
}

identity ll-only {
  base address-allocation-type;
  description
    "Use IPv6 Link Local.";
}

identity traffic-direction {
  description
    "Base identity for traffic direction";
}

identity inbound {
  base traffic-direction;
  description
    "Identity for inbound";
}

identity outbound {
  base traffic-direction;
  description
    "Identity for outbound";
}

identity both {
  base traffic-direction;
  description
    "Identity for both";
}

identity traffic-action {
```



```
    description
      "Base identity for traffic action";
  }

  identity permit {
    base traffic-action;
    description
      "Identity for permit action";
  }

  identity deny {
    base traffic-action;
    description
      "Identity for deny action";
  }

  identity bd-limit-type {
    description
      "base identity for bd limit type";
  }

  identity percent {
    base bd-limit-type;
    description
      "Identity for percent";
  }

  identity value {
    base bd-limit-type;
    description
      "Identity for value";
  }

  identity protocol-type {
    description
      "Base identity for protocol field type.";
  }

  identity tcp {
    base protocol-type;
    description
      "TCP protocol type.";
  }

  identity udp {
    base protocol-type;
    description
      "UDP protocol type.";
```

```
}

identity icmp {
  base protocol-type;
  description
    "ICMP protocol type.";
}

identity icmp6 {
  base protocol-type;
  description
    "ICMPv6 protocol type.";
}

identity gre {
  base protocol-type;
  description
    "GRE protocol type.";
}

identity ipip {
  base protocol-type;
  description
    "IP-in-IP protocol type.";
}

identity hop-by-hop {
  base protocol-type;
  description
    "Hop-by-Hop IPv6 header type.";
}

identity routing {
  base protocol-type;
  description
    "Routing IPv6 header type.";
}

identity esp {
  base protocol-type;
  description
    "ESP header type.";
}

identity ah {
  base protocol-type;
  description
    "AH header type.";
```

```
}

grouping vpn-endpoint {
  leaf endpoint-id {
    type svc-id;
    description
      "Identity for the vpn endpoint";
  }
  leaf site-role {
    type identityref {
      base site-role;
    }
    default "any-to-any-role";
    description
      "Role of the site in the VPN.";
  }
  container site-attachment {
    leaf site-id {
      type leafref {
        path "/sdwan-svc/sites/site/site-id";
      }
      description
        "Defines site id attached.";
    }
    description
      "Defines site attachment to a vpn endpoint.";
  }
  container endpoint-policy-map {
    list app-group-policy {
      key "app-group-id";
      leaf app-group-id {
        type leafref {
          path "/sdwan-svc/vpn-services/vpn-service"+
            "/application-group/app-group-id";
        }
        description
          "Identity for application";
      }
      leaf policy-id {
        type leafref {
          path "/sdwan-svc/vpn-services/vpn-service/policy/policy-id";
        }
        description
          "Identity for value";
      }
      description
        "list for application group policy";
    }
  }
}
```

```
list app-policy {
  key "app-id";
  leaf app-id {
    type leafref {
      path "/sdwan-svc/vpn-services/vpn-service"+
        "/application/app-id";
    }
    description
      "Identity for application";
  }
  leaf policy-id {
    type leafref {
      path "/sdwan-svc/vpn-services/vpn-service/policy/policy-id";
    }
    description
      "Identity for value";
  }
  description
    "list for application policy";
}
description
  "Identity for policy maps";
}
description
  "grouping for vpn endpoint";
}

grouping flow-definition {
  container match-flow {
    leaf ethertype {
      type uint16;
      description
        "Ethertype value, e.g. 0800 for IPv4.";
    }
    leaf cvlan {
      type uint8 {
        range "0..7";
      }
      description
        "802.1Q matching.";
    }
    leaf ipv4-src-prefix {
      type inet:ipv4-prefix;
      description
        "Match on IPv4 src address.";
    }
    leaf ipv4-dst-prefix {
      type inet:ipv4-prefix;
    }
  }
}
```

```
        description
            "Match on IPv4 dst address.";
    }
    leaf l4-src-port {
        type inet:port-number;
        description
            "Match on Layer 4 src port.";
    }
    leaf l4-dst-port {
        type inet:port-number;
        description
            "Match on Layer 4 dst port.";
    }
    leaf ipv6-src-prefix {
        type inet:ipv6-prefix;
        description
            "Match on IPv6 src address.";
    }
    leaf ipv6-dst-prefix {
        type inet:ipv6-prefix;
        description
            "Match on IPv6 dst address.";
    }
    leaf protocol-field {
        type union {
            type uint8;
            type identityref {
                base protocol-type;
            }
        }
        description
            "Match on IPv4 protocol or IPv6 Next Header field.";
    }
    description
        "Describes flow-matching criteria.";
}
description
    "Grouping for flow definition.";
}

grouping application-criteria {
    list ac {
        key "name";
        ordered-by user;
        leaf name {
            type string;
            description
                "A description identifying application classification";
        }
    }
}
```

```
        criteria.";
    }
    choice match-type {
        default "match-flow";
        case match-flow {
            uses flow-definition;
        }
        case match-application {
            leaf match-application {
                type identityref {
                    base customer-application;
                }
                description
                    "Defines the application to match.";
            }
        }
        description
            "Choice for classification.";
    }
    description
        "List of marking rules.";
}
description
    "This grouping defines QoS parameters for a site.";
}

grouping vpn-service {
    leaf vpn-id {
        type svc-id;
        description
            "Identity for VPN.";
    }
    leaf topology {
        type identityref {
            base vpn-topology;
        }
        description
            "vpn topology: hub-and-spoke or any-to-any";
    }
    container performance-objective {
        leaf start-time {
            type yang:date-and-time;
            description
                "start-time indicates date and time.";
        }
        leaf duration {
            type string;
            description
```

```
        "Time duration.";
    }
    container uptime-objective {
        leaf duration {
            type decimal64 {
                fraction-digits 5;
                range "0..100";
            }
            units "percent";
            description
                "To be used to define the a percentage of the available
                service.";
        }
        description
            "Uptime objective.";
    }
    description
        "The performance objective.";
}
container reserved-prefixes {
    leaf-list prefix {
        type inet:ip-prefix;
        description
            "ip prefix reserved for SP management purpose.";
    }
    description
        "ip prefix list reserved for SP management purpose.";
}
list application {
    key "app-id";
    leaf app-id {
        type svc-id;
        description
            "application name";
    }
    uses application-criteria;
    description
        "list for application";
}
list application-group {
    key "app-group-id";
    leaf app-group-id {
        type svc-id;
        description
            "application name";
    }
    leaf-list app-id {
        type leafref {
```

```
        path "../../application/app-id";
    }
    description
        "application member list in an application group";
}
description
    "list for application group";
}
list policy {
    key "policy-id";
    leaf policy-id {
        type svc-id;
        description
            "Policy names";
    }
    container policy-package {
        leaf encryption {
            type enumeration {
                enum yes {
                    description
                        "Indicates whether or not the application flow requires
                        to send over encrypted overlay tunnel.";
                }
                enum either {
                    description
                        " Either means this policy is not applied";
                }
            }
        }
        description
            "Indicates whether or not the application flow requires
            encryption.";
    }
}
leaf public-private {
    type enumeration {
        enum private-only {
            description
                "The private WAN underlay is specified.";
        }
        enum either {
            description
                "Both public WAN or private WAN could be used";
        }
    }
}
description
    "Indicates whether the Application Flow can traverse
    Public or Private Underlay Connectivity Services
    (or both).Either means this policy is not applied.";
}
```



```
leaf local-breakout {
    type boolean;
    description
        "indicates whether the Application Flow should be
        routed directly to the Internet using Local Internet
        Breakout.It can have values Yes and No.";
}
leaf billing-method {
    type enumeration {
        enum flat-only {
            description
                "Only flat-rate underlay could be used for the
                traffic.";
        }
        enum either {
            description
                "Either flat-rate or usage based underlay could
                be used for the traffic.";
        }
    }
    description
        "billing policy.";
}
leaf backup-path {
    type enumeration {
        enum yes {
            description
                "Only the primary tunnel overlay could be used for
                the traffic.";
        }
        enum no {
            description
                "Either the primary or backup overlay tunnel could be
                used for the traffic.";
        }
    }
    description
        "overlay connection as Primary or both Primary and
        Backup.";
}
container bandwidth {
    leaf commit {
        type uint32;
        description
            "CIR";
    }
    leaf max {
        type uint32;
    }
}
```

```
        description
            "max speed ";
    }
    description
        "Container for the bandwidth policy";
    }
    description
        "Container for policy package";
    }
    description
        "List for policy";
    }
    list endpoints {
        key "endpoint-id";
        uses vpn-endpoint;
        description
            "List of endpoints.";
    }
    description
        "Grouping of vpn service";
    }

    grouping site-l2-technology {
        container l2-technology {
            leaf l2-type {
                type identityref {
                    base eth-inf-type;
                }
                default "untagged";
                description
                    "Defines physical properties of an interface. By default, the
                     Ethernet interface type is set to 'untagged'.";
            }
            container untagged-interface {
                leaf speed {
                    type uint32;
                    units "mbps";
                    default "10";
                    description
                        "Port speed.";
                }
                leaf mode {
                    type neg-mode;
                    default "auto-neg";
                    description
                        "Negotiation mode.";
                }
            }
            description
```

```
        "Container of Untagged Interface Attributes
        configurations.";
    }
    container tagged-interface {
        leaf type {
            type identityref {
                base tagged-inf-type;
            }
            default "dot1q";
            description
                "Tagged interface type. By default,
                the Tagged interface type is dot1q interface. ";
        }
        container dot1q-vlan-tagged {
            leaf tg-type {
                type identityref {
                    base tag-type;
                }
                default "c-vlan";
                description
                    "TAG type.By default, Tag type is Customer-VLAN tag.";
            }
            leaf cvlan-id {
                type uint16;
                mandatory true;
                description
                    "VLAN identifier.";
            }
            description
                "Tagged interface.";
        }
        container priority-tagged {
            leaf tag-type {
                type identityref {
                    base tag-type;
                }
                default "c-vlan";
                description
                    "TAG type.By default, the TAG type is
                    Customer-VLAN tag.";
            }
            description
                "Priority tagged.";
        }
        description
            "Container for tagged Interface.";
    }
    leaf l2-mtu {
```

```
    type uint32;
    units "bytes";
    description
      " L2 Maximum Frame Size MUST be an integer number of bytes
        >= 1522MTU.";
  }
  description
    "Container for l2 technology.";
}
description
  "grouping for l2 technology.";
}

grouping site-ip-connection {
  container ip-connection {
    container ipv4 {
      leaf address-allocation-type {
        type identityref {
          base address-allocation-type;
        }
        description
          "Defines how addresses are allocated.
            If there is no value for address
            allocation type, then the ipv4 is not enabled.";
      }
    }
    container dhcp {
      container primary-subnet {
        leaf ip-prefix {
          type inet:ipv4-prefix;
          description
            "IPv4 address prefix and mask length between 0 and 31,
              in bits.";
        }
      }
      leaf default-router {
        type inet:ip-address;
        description
          "Address of default router.";
      }
      leaf-list provider-addresses {
        type inet:ipv4-address;
        description
          "the Service Provider IPv4 Addresses MUST be within the
            specified IPv4 Prefix.";
      }
      leaf subscriber-address {
        type inet:ip-address;
        description
          "subscriber IPv4 Addresses: Non-empty list
```

```
        of IPv4 addresses";
    }
    leaf-list reserved-ip-prefix {
        type inet:ip-prefix;
        description
            "List of IPv4 Prefixes, possibly empty";
    }
    description
        "Primary Subnet List";
}
list secondary-subnet {
    key "ip-prefix";
    leaf ip-prefix {
        type inet:ipv4-prefix;
        description
            "IPv4 address prefix and mask length between 0 and 31,
            in bits";
    }
    leaf-list provider-addresses {
        type inet:ipv4-address;
        description
            "Service Provider IPv4 Addresses: Non-empty list
            of IPv4 addresses";
    }
    leaf-list reserved-ip-prefix {
        type inet:ipv4-prefix;
        description
            "List of IPv4 Prefixes, possibly empty";
    }
    description
        "Secondary Subnet List";
}
description
    "DHCP allocated addresses related parameters.";
}
container static {
    container primary-subnet {
        leaf ip-prefix {
            type inet:ipv4-prefix;
            description
                "IPv4 address prefix and mask length between 0 and 31,
                in bits.";
        }
    }
    leaf default-router {
        type inet:ip-address;
        description
            "Address of default router.";
    }
}
```

```
    leaf-list provider-addresses {
      type inet:ipv4-address;
      description
        "the Service Provider IPv4 Addresses MUST be within the
        specified IPv4 Prefix.";
    }
    leaf subscriber-address {
      type inet:ip-address;
      description
        "subscriber IPv4 Addresses: Non-empty list
        of IPv4 addresses";
    }
    leaf-list reserved-ip-prefix {
      type inet:ip-prefix;
      description
        "List of IPv4 Prefixes, possibly empty";
    }
    description
      "Primary Subnet List";
  }
  list secondary-subnet {
    key "ip-prefix";
    leaf ip-prefix {
      type inet:ipv4-prefix;
      description
        "IPv4 address prefix and mask length between 0 and 31,
        in bits";
    }
    leaf-list provider-addresses {
      type inet:ipv4-address;
      description
        "Service Provider IPv4 Addresses: Non-empty list
        of IPv4 addresses";
    }
    leaf-list reserved-ip-prefix {
      type inet:ipv4-prefix;
      description
        "List of IPv4 Prefixes, possibly empty";
    }
    description
      "Secondary Subnet List";
  }
  description
    "Static configuration related parameters.";
}
description
  "IPv4-specific parameters.";
}
```

```
container ipv6 {
  leaf address-allocation-type {
    type identityref {
      base address-allocation-type;
    }
    description
      "Defines how addresses are allocated.
      If there is no value for address
      allocation type, then the ipv6 is not enabled.";
  }
  container dhcp {
    list subnet {
      key "ip-prefix";
      leaf ip-prefix {
        type inet:ipv6-prefix;
        description
          "IPv6 address prefix and prefix length between 0 and
          128";
      }
      leaf-list provider-addresses {
        type inet:ipv6-address;
        description
          "Non-empty list of IPv6 addresses";
      }
      leaf-list reserved-ip-prefix {
        type inet:ipv6-prefix;
        description
          "List of IPv6 Prefixes, possibly empty";
      }
      description
        "Subnet List";
    }
    description
      "DHCP allocated addresses related parameters.";
  }
  container slaac {
    list subnet {
      key "ip-prefix";
      leaf ip-prefix {
        type inet:ipv6-prefix;
        description
          "IPv6 address prefix and prefix length of 64 ";
      }
      leaf-list provider-addresses {
        type inet:ipv6-address;
        description
          "Non-empty list of IPv6 addresses";
      }
    }
  }
}
```

```
        leaf-list reserved-ip-prefix {
            type inet:ipv6-prefix;
            description
                "List of IPv6 Prefixes, possibly empty";
        }
        description
            "Subnet List";
    }
    description
        "DHCP allocated addresses related parameters.";
}
container static {
    list subnet {
        key "ip-prefix";
        leaf ip-prefix {
            type inet:ipv6-prefix;
            description
                "IPv6 address prefix and prefix length between 0 and
                128";
        }
        leaf-list provider-addresses {
            type inet:ipv6-address;
            description
                "Non-empty list of IPv6 addresses";
        }
        leaf-list reserved-ip-prefix {
            type inet:ipv6-prefix;
            description
                "List of IPv6 Prefixes, possibly empty";
        }
        description
            "Subnet List";
    }
    leaf subscriber-address {
        type inet:ipv6-address;
        description
            "IPv6 address or Not Specified.";
    }
    description
        "Static configuration related parameters.";
}
description
    "Describes IPv6 addresses used.";
}
description
    "IPv6-specific parameters.";
}
description
```



```
    "This grouping defines IP connection parameters.";
}

container sdwan-svc {
  container vpn-services {
    list vpn-service {
      key "vpn-id";
      uses vpn-service;
      description
        "List for SD-WAN";
    }
    description
      "Container for SD-WAN VPN service";
  }
  container sites {
    list site {
      key "site-id";
      leaf site-id {
        type svc-id;
        description
          "Site Name";
      }
    }
    list device {
      key "name";
      leaf name {
        type string;
        description
          "Device Name";
      }
      leaf type {
        type identityref {
          base device-type;
        }
        description
          "Device Type: virtual or physical CE";
      }
      description
        "List for device";
    }
  }
  list lan-access {
    key "name";
    leaf name {
      type string;
      description
        "lan access link name";
    }
    uses site-l2-technology;
    uses site-ip-connection;
  }
}
```

```
        description
            "container for lan access";
    }
    list wan-access {
        key "name";
        leaf name {
            type string;
            description
                "wan access link name";
        }
        leaf access-type {
            type identityref {
                base access-type;
            }
            description
                "Access type: Internet, private VPN or cellular";
        }
        leaf access-provider {
            type string;
            description
                "Specifies the name of provider";
        }
        container bandwidth {
            leaf input-bandwidth {
                type uint64;
                description
                    "input bandwidth";
            }
            leaf output-bandwidth {
                type uint64;
                description
                    "output bandwidth";
            }
            description
                "Container for bandwidth";
        }
        uses site-l2-technology;
        uses site-ip-connection;
        description
            "container for wan access";
    }
    description
        "List for site";
}
description
    "Container for sites";
}
description
```

```
    "Top-level container for the SD-WAN services.";
  }
}
```

<CODE ENDS>

7. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability.

8. IANA Considerations

IANA has assigned a new URI from the "IETF XML Registry" [RFC3688].

```
URI: urn:ietf:params:xml:ns:yang:ietf-sdwan-svc
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.
```

IANA has recorded a YANG module name in the "YANG Module Names" registry [RFC6020] as follows:

```
Name: ietf-sdwan-svc
Namespace: urn:ietf:params:xml:ns:yang:ietf-sdwan-svc
Prefix: sdwan-svc
Reference: RFC xxxx
```

9. Appendix 1: Terminology Mapping between MEF SD-WAN Service Attributes and IETF SD-WAN model

SD-WAN Service Attributes and Services [MEF70-Draft-R1], defines the SD-WAN service attributes and services for SD-WAN service delivery. These service attributes can be used for communication between subscribers and services to deliver SD-WAN services while this draft defines a YANG data model for SD-WAN service delivery communicated between customer and service provider. The purpose of both work is very similar.

The below table shows the terminology mapping. The YANG model retains most parameter definition name but adjusts some of the structure to reserve space for future augmentation. For example, the model defines "vpn-service" and "lan-access" as a list, which can accommodate the case where the current MEF service attribute restricts only one VPN per customer and one LAN access and future extension to multiple VPN or LAN accesses per customer.

IETF SD-WAN Service model	MEF70 R1 SD-WAN Services Term
SD-WAN VPN	SD-WAN Virtual Connection (SWVC)
SD-WAN VPN Endpoint	SWVC End Point
Site	User Network Interface (UNI)
lan-access	UNI link Attributes
wan-access	TBD(Underlay connectivity)

10. Appendix 2: IETF OSE model vs IETF SD-WAN model

SD-WAN OSE service delivery model [I-D.wood-rtgwg-sdwan-ose-yang] defines two SD-WAN OSE Open SD-WAN Exchange (OSE) service YANG modules to enable the orchestrator in the enterprise network to implement SD-WAN inter-domain reachability and connectivity services and application aware traffic steering services. Although the OSE YANG model is also a service model instead of being a device model, this model is mainly used for interoperability between multiple SD-WAN domains and service consistency. The differences are shown as follows:

IETF OSE service model	IETF SD-WAN Service model
Domain SD-WAN controller facing	customer-facing
Inter OSE GW connectivity service	unaware of SD-WAN domain in one SP network
Inter SD-WAN domain	Inter-SD-WAN Service Provider TBD
SLA aware dynamic Path selection	static Primary/Backup selection

For the SLA based dynamic path selection policy, the OSE service model uses a similar application classification criteria, but at the same time it will collect the relevant status of the traffic SLA profiles and, based on the measurements calculated from the collected information, the primary or secondary path will be selected.

```

+--primary-backup
  +--rw path-values
    +--rw sla-values
      +--rw latency?          uint32
      +--rw jitter?          uint32
      +--rw packet-loss-rate? uint32

```

11. Acknowledgments

This work has benefited from the discussions of with Jack Pugaczewski, Larry S Samberg, and Pascal Menezes from MEF community.

12. Contributors

The authors would like to thank Zitao Wang for his major contributions to the initial modelling.

13. References

13.1. Normative References

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

13.2. Informative References

- [I-D.wood-rtgwg-sdwan-ose-yang]
Wood, S., Bo, W., Wu, Q., and C. Menezes, "YANG Data Model for SD-WAN OSE service delivery", draft-wood-rtgwg-sdwan-ose-yang-00 (work in progress), March 2019.
- [MEF51.1] MEF, Ed., "Operator Ethernet Service Definition", December 2018, <<https://wiki.mef.net/display/CESG/MEF+51.1+-+OVC+Services>>.
- [MEF70-Draft-R1]
MEF, Ed., "SD-WAN Service Attributes and Services", May 2019, <[https://www.mef.net/Assets/Draft-Standards/MEF_70_Draft_\(R1\).pdf](https://www.mef.net/Assets/Draft-Standards/MEF_70_Draft_(R1).pdf)>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, DOI 10.17487/RFC4026, March 2005, <<https://www.rfc-editor.org/info/rfc4026>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, DOI 10.17487/RFC6071, February 2011, <<https://www.rfc-editor.org/info/rfc6071>>.

- [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>>.
- [RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <<https://www.rfc-editor.org/info/rfc7426>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.

Authors' Addresses

Qiong Sun
China Telecom
Beijing
China

Email: sunqiong.bri@chinatelecom.cn

Honglei Xu
China Telecom
Beijing
China

Email: xuhl.bri@chinatelecom.cn

Bo Wu (editor)
Huawei
Nanjing
China

Email: lana.wubo@huawei.com

Qin Wu (editor)
Huawei
Nanjing
China

Email: bill.wu@huawei.com

Charles Eckel (editor)
Cisco Systems
170 W. Tasman Drive
San Jose, CA
United States

Email: eckelcu@cisco.com

Networking Working Group
Internet-Draft
Intended status: Informational
Expires: April 28, 2020

Q. Wu, Ed.
Huawei
M. Boucadair, Ed.
Orange
D. Lopez
Telefonica I+D
C. Xie
China Telecom
L. Geng
China Mobile
October 26, 2019

A Framework for Automating Service and Network Management with YANG
draft-wu-model-driven-management-virtualization-07

Abstract

Data models for service and network management provides a programmatic approach for representing (virtual) services or networks and deriving (1) configuration information that will be communicated to network and service components that are used to build and deliver the service and (2) state information that will be monitored and tracked. Indeed, data models can be used during various phases of the service and network management life cycle, such as service instantiation, service provisioning, optimization, monitoring, and diagnostic. Also, data models are instrumental in the automation of network management. They also provide closed-loop control for the sake of adaptive and deterministic service creation, delivery, and maintenance.

This document provides a framework that describes and discusses an architecture for service and network management automation that takes advantage of YANG modeling technologies. This framework is drawn from a network provider perspective irrespective of the origin of a data module; it can accommodate even modules that are developed outside the IETF.

The document aims to exemplify an approach that specifies the journey from technology-agnostic services to technology-specific actions.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on April 28, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Terminology	5
3. Architectural Concepts & Goals	5
3.1. Data Models: Layering and Representation	5
3.2. Automation of Service Delivery Procedures	6
3.3. Service Fullfillment Automation	7
3.4. YANG Modules Integration	7
4. Architecture Overview	8
4.1. Service Lifecycle Management Procedure	9
4.1.1. Service Exposure	10
4.1.2. Service Creation/Modification	10
4.1.3. Service Optimization	10
4.1.4. Service Diagnosis	11
4.1.5. Service Decommission	11
4.2. Service Fullfillment Management Procedure	11
4.2.1. Intended Configuration Provision	11
4.2.2. Configuration Validation	12
4.2.3. Performance Monitoring	12
4.2.4. Fault Diagnostic	13
4.3. Multi-layer/Multi-domain Service Mapping	13
4.4. Service Decomposing	13

5. YANG Data Model Integration Examples	13
5.1. L3VPN Service Delivery	13
5.2. VN Lifecycle Management Example	15
6. Security Considerations	16
7. IANA Considerations	16
8. Acknowledgements	16
9. Contributors	16
10. Informative References	17
Appendix A. Layered YANG Modules Example Overview	25
A.1. Service Models: Definition and Samples	25
A.2. Network Models: Definitions and Samples	26
A.3. Device Models: Definitions and Samples	29
A.3.1. Model Composition	30
A.3.2. Device Models: Definitions and Samples	30
Authors' Addresses	33

1. Introduction

The service management system usually comprises service activation/provision and service operation. Current service delivery procedures, from the processing of customer's requirements and order to service delivery and operation, typically assume the manipulation of data sequentially into multiple OSS/BSS applications that may be managed by different departments within the service provider's organization (e.g., billing factory, design factory, network operation center, etc.). In addition, many of these applications have been developed in-house over the years and operating in a silo mode:

- o The lack of standard data input/output (i.e., data model) also raises many challenges in system integration and often results in manual configuration tasks.
- o Secondly, many current service fulfillment system might have limited visibility to the network and therefore have slow response to the network changes.

Software Defined Networking (SDN) becomes crucial to address these challenges. SDN techniques [RFC7149] are meant to automate the overall service delivery procedures and typically rely upon (standard) data models that are used to not only reflect service providers' savoir-faire but also to dynamically instantiate and enforce a set of (service-inferred) policies that best accommodate what has been (contractually) defined (and possibly negotiated) with the customer. [RFC7149] provides a first tentative to rationalize that service provider's view on the SDN space by identifying concrete technical domains that need to be considered and for which solutions can be provided:

- o Techniques for the dynamic discovery of topology, devices, and capabilities, along with relevant information and data models that are meant to precisely document such topology, devices, and their capabilities.
- o Techniques for exposing network services [RFC8309] and their characteristics.
- o Techniques used by service-requirement-derived dynamic resource allocation and policy enforcement schemes, so that networks can be programmed accordingly.
- o Dynamic feedback mechanisms that are meant to assess how efficiently a given policy (or a set thereof) is enforced from a service fulfillment and assurance perspective.

Models are key for each of these technical items. Service and network management automation is an important step to improve the agility of network operations and infrastructures. Models are also important to ease integrating multi-vendor solutions.

YANG module developers have taken both top-down and bottom-up approaches to develop modules [RFC8199], and also to establish a mapping between network technology and customer requirements on the top or abstracting common construct from various network technologies on the bottom. At the time of writing this document (2019), there are many data models including configuration and service models that have been specified or are being specified by the IETF. They cover many of the networking protocols and techniques. However, how these models work together to configure a device, manage a set of devices involved in a service, or even provide a service is something that is not currently documented either within the IETF or other SDOs (e.g., MEF).

This document provides a framework that describes and discusses an architecture for service and network management automation that takes advantage of YANG modeling technologies and investigates how different layer YANG data models interact with each other (e.g., service mapping, model composing) in the context of service delivery and fulfillment.

This framework is drawn from a network provider perspective irrespective of the origin of a data module; it can accommodate even modules that are developed outside the IETF.

The document also identifies a list of use cases to exemplify the proposed approach, but it does not claim to be exhaustive.

2. Terminology

The following terms are defined in [RFC8309][RFC8199] and are not redefined here:

- o Network Operator
- o Customer
- o Service
- o Data Model
- o Service Model
- o Network Element Module

The document makes use of the following terms:

Network Model: The Network Model describes network level abstraction or various aspects of a network infrastructure, including devices and their subsystems, and relevant protocols operating at the link and network layers across multiple devices. It can be used by a network operator to allocate the resource(e.g., tunnel resource, topology resource) for the service or schedule the resource to meet the service requirements define in the Service Model.

Device Model: Network Element YANG data module described in [RFC8199].

3. Architectural Concepts & Goals

3.1. Data Models: Layering and Representation

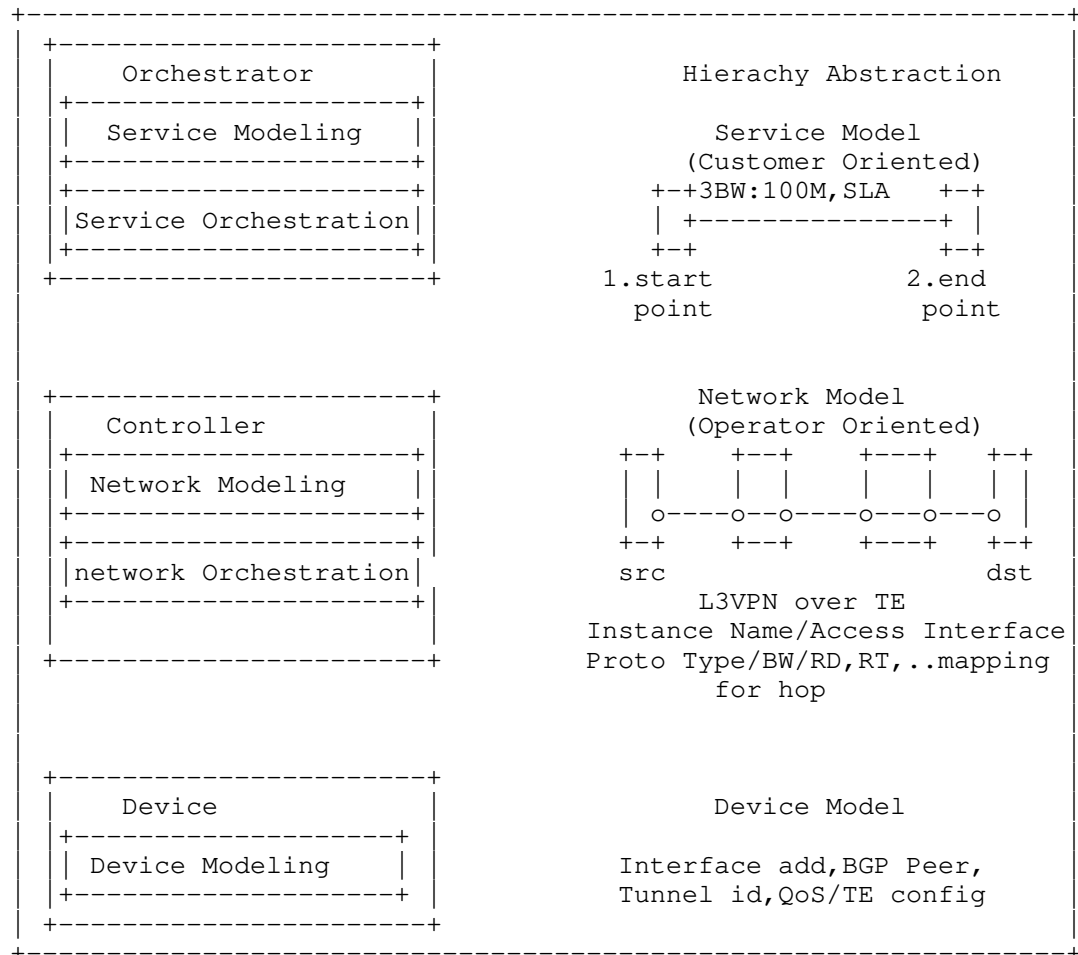
As described in [RFC8199], layering of modules allows for better reusability of lower-layer modules by higher-level modules while limiting duplication of features across layers.

The data modules developed by IETF can be classified into service level, network level and device level modules. Different service model at service level may rely on the same set of network level or device level models. Service models usually follow top down approach and are mostly customer-facing modules providing a common model construct for higher level network services, which can be further mapped to network technology-specific modules at lower layer.

Network level modules are mainly network resource-facing modules and describe various aspects of a network infrastructure, including

devices and their subsystems, and relevant protocols operating at the link and network layers across multiple devices (e.g., Network topology and TE Tunnel modules).

Device level modules usually follow a bottom-up approach and are mostly technology-specific modules used to realize a service.



Layering and representation

3.2. Automation of Service Delivery Procedures

To dynamically offer and deliver service offerings, Service level modules can be used by an operator. One or more monolithic Service modules can be used in the context of a composite service activation

request (e.g., delivery of a caching infrastructure over a VPN). Such modules are used to feed a decision-making intelligence to adequately accommodate customer's needs.

Also, such modules may be used jointly with services that require dynamic invocation. An example is provided by the service modules defined by the DOTS WG to dynamically trigger requests to handle DDoS attacks [I-D.ietf-dots-signal-channel][I-D.ietf-dots-data-channel].

Network level modules can be derived from service level modules and used to provision, monitor, instantiate the service, and provide lifecycle management of network resources (e.g., expose network resources to customers or operators to provide service fulfillment and assurance and allow customers or operators to dynamically adjust the network resources based on service requirements as described in service level modules and the current network performance information described in the telemetry modules).

3.3. Service Fullfillment Automation

To operate the service, Device level modules derived from Service level modules or Network level modules can be used to provision each involved network function/device with the proper configuration information, and operate the network based on service requirements as described in the Service level module(s).

In addition, the operational state including configuration that is in effect together with statistics should be exposed to upper layers to provide better network visibility (and assess to what extent the derived low level modules are consistent with the upper level inputs).

Note that it is important to correlate telemetry data with configuration data to be used for closed loops at the different stages of service delivery, from resource allocation to service operation, in particular.

3.4. YANG Modules Integration

To support top-down service delivery, YANG modules at different level or at the same level need to be integrated together to enable function, feature in the network device and get network setup. For example, the service parameters captured in service level modules need to be decomposed into a set of (configuration/notification) parameters that may be specific to one or more technologies; these technology-specific parameters are grouped together to define technology-specific device level models or network level models.

In addition, these technology-specific device level models or network level models can be further integrated with each other using schema mount mechanism [RFC8528] to provision each involved network function/device or each involved administrative domain to support newly added module or features. A collection of device models integrated together can be loaded and validated during implementation time.

Policies provide a higher layer of abstraction. Policy models can be defined at service level, network level, or device level to provide policy-based management and telemetry automation, e.g., telemetry data can trigger a new policy that captures new network service requirements.

Performance measurement telemetry can be used to provide service assurance at service level or at the network level. Performance measurement telemetry model can tie with network level model or service level model to monitor network performance or service level agreement.

4. Architecture Overview

The architectural considerations described in Section 3 lead to the architecture described in this section and illustrated in Figure 1.

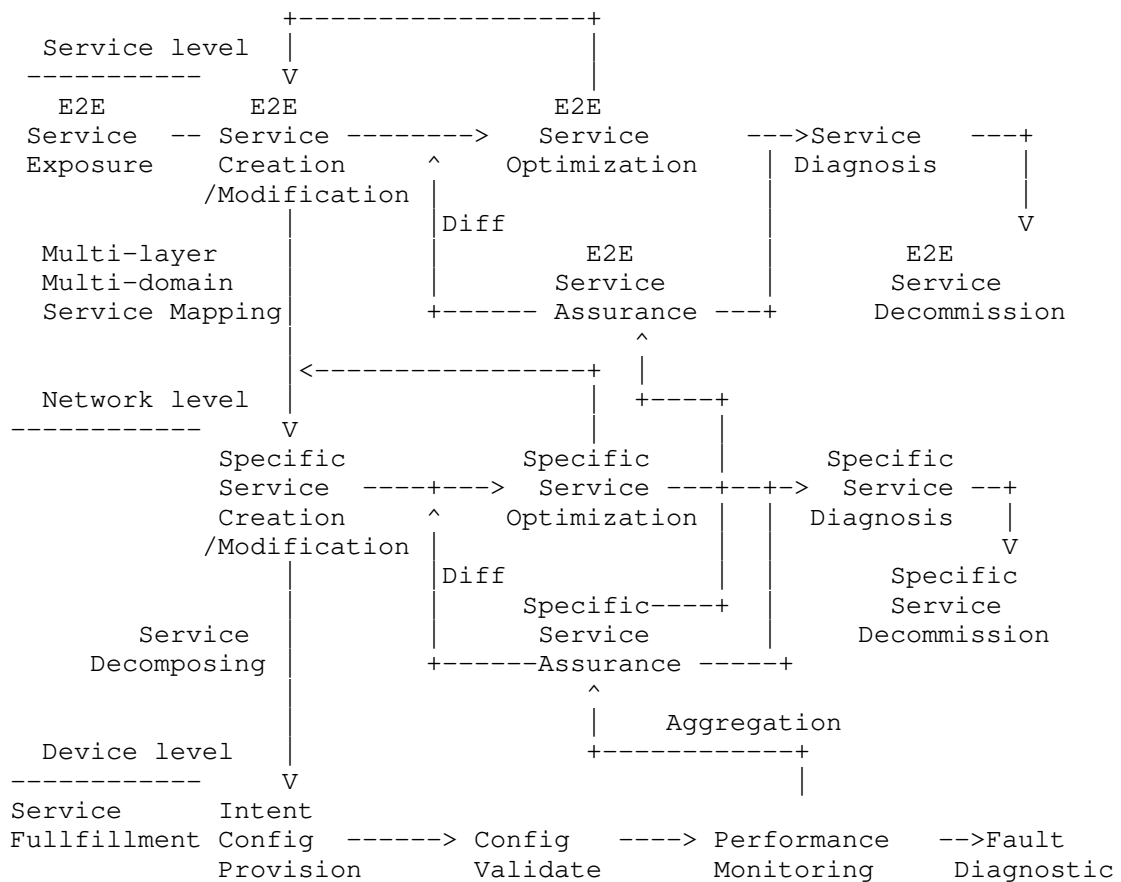


Figure 1: Service and Network Lifecycle Management

4.1. Service Lifecycle Management Procedure

Service lifecycle management includes end to end service lifecycle management at the service level and specific network lifecycle management at the network level. The end-to-end service lifecycle management is multi-domain or multi-layer service management while specific service lifecycle management is domain specific or layer specific service lifecycle management.

- o Note: Clarify what is meant by "domain".

4.1.1.1. Service Exposure

A service in the context of this document (sometimes called a Network Service) is some form of connectivity between customer sites and the Internet or between customer sites across the network operator's network and across the Internet.

Service exposure is used to capture services offered to customers (ordering and order handling). One typical example is that a customer can use a L3SM service model to request L3VPN service by providing the abstract technical characterization of the intended service between customer sites.

Service model catalogs can be created along to expose the various services and the information needed to invoke/order a given service.

4.1.1.2. Service Creation/Modification

A customer is (usually) unaware of the technology that the network operator has available to deliver the service, so the customer does not make requests specific to the underlying technology but is limited to making requests specific to the service that is to be delivered. This service request can be issued using the service model.

The service orchestrator/management system maps such service request to its view. This view can be described as a network model and this mapping may include a choice of which networks and technologies to use depending on which service features have been requested.

In addition, a customer may require to change underlying network infrastructure to adapt to new customer's needs and service requirements. This service modification can be issued in the same service model used by the service request.

4.1.1.3. Service Optimization

Service optimization is a technique that gets the configuration of the network updated due to network change, incident mitigation, or new service requirements. One typical example is once the tunnel or the VPN is setup, Performance monitoring information or telemetry information per tunnel or per VPN can be collected and fed into the management system, if the network performance doesn't meet the service requirements, the management system can create new VPN policies capturing network service requirements and populate them into the network.

Both network performance information and policies can be modelled using YANG. With Policy-based management, self-configuration and self-optimization behavior can be specified and implemented.

4.1.4. Service Diagnosis

Operations, Administration, and Maintenance (OAM) are important networking functions for service diagnosis that allow operators to:

- o monitor network communications (i.e., reachability verification and Continuity Check)
- o troubleshoot failures (i.e., fault verification and localization)
- o monitor service-level agreements and performance (i.e., performance management)

When the network is down, service diagnosis should be in place to pinpoint the problem and provide recommendation (or instructions) for the network recovery.

The service diagnosis information can be modelled as technology-independent RPC operations for OAM protocols and technology-independent abstraction of key OAM constructs for OAM protocols [RFC8531][RFC8533]. These models can provide consistent configuration, reporting, and presentation for the OAM mechanisms used to manage the network.

4.1.5. Service Decommission

Service decommission allow the customer to stop the service and remove the service from active status and release the network resource that is allocated to the service. Customer can also use the service model to withdraw the subscription to a service.

4.2. Service Fullfillment Management Procedure

4.2.1. Intended Configuration Provision

Intended configuration at the device level is derived from network model at the network level or service model at the service level and represents the configuration that the system attempts to apply. Take L3SM service model as an example, to deliver a L3VPN service, we need to map L3VPN service view defined in Service model into detailed intended configuration view defined by specific configuration models for network elements, configuration information includes:

- o VRF definition, including VPN Policy expression

- o Physical Interface
- o IP layer (IPv4, IPv6)
- o QoS features such as classification, profiles, etc.
- o Routing protocols: support of configuration of all protocols listed in the document, as well as routing policies associated with those protocols.
- o Multicast Support
- o NAT or address sharing
- o Security function

This specific configuration models can be used to configure PE and CE devices within the site, e.g., A BGP policy model can be used to establish VPN membership between sites and VPN Service Topology.

4.2.2. Configuration Validation

Configuration validation is used to validate intended configuration and ensure the configuration take effect. For example, a customer creates an interface "et-0/0/0" but the interface does not physically exist at this point, then configuration data appears in the <intended> status but does not appear in <operational> datastore.

4.2.3. Performance Monitoring

When configuration is in effect in the device, <operational> datastore holds the complete operational state of the device including learned, system, default configuraton and system state. However the configurations and state of a particular device does not have the visibility to the whole network or information of the flow packets are going to take through the entire network. Therefore it becomes more difficult to operate the network without understanding the current status of the network.

The management system should subscribe to updates of a YANG datastore in all the network devices for performance monitoring purpose and build full topological visibility to the network by aggregating and filtering these operational state from different sources.

4.2.4. Fault Diagnostic

When configuration is in effect in the device, some device may be misconfigured (e.g., device links are not consistent on both sides of the network connection), network resources be misallocated and services may be negatively affected without knowing what is going on in the network.

Technology-dependent nodes and remote procedure call (RPC) commands are defined in technology-specific YANG data models which can use and extend the base model described in Section 4.1.4 can be used to deal with these challenges.

These RPC command received in the technology dependent node can be used to trigger technology specific OAM message exchange for fault verification and fault isolation, e.g., TRILL Multicast Tree Verification (MTV) RPC command [I-D.ietf-trill-yang-oam] can be used to trigger Multi-Destination Tree Verification Message defined in [RFC7455] to verify TRILL distribution tree integrity.

4.3. Multi-layer/Multi-domain Service Mapping

Multi-layer/Multi-domain Service Mapping allow you map end to end abstract view of the service segmented at different layer or different administrative domain into domain specific view. One example is to map service parameters in L3VPN service model into configuration parameters such as RD, RT, and VRF in L3VPN network model. Another example is to map service parameters in L3VPN service model into TE tunnel parameter (e.g., Tunnel ID) in TE model and VN parameters (e.g., AP list, VN member) in TEAS VN model [I-D.ietf-teas-actn-vn-yang].

4.4. Service Decomposing

Service Decomposing allows to decompose service model at the service level or network model at the network level into a set of device/function models at the device level. These device models may be tied to specific device type or classified into a collection of related YANG modules based on service type and feature offered and load at the implementation time before configuration is loaded and validated.

5. YANG Data Model Integration Examples

5.1. L3VPN Service Delivery

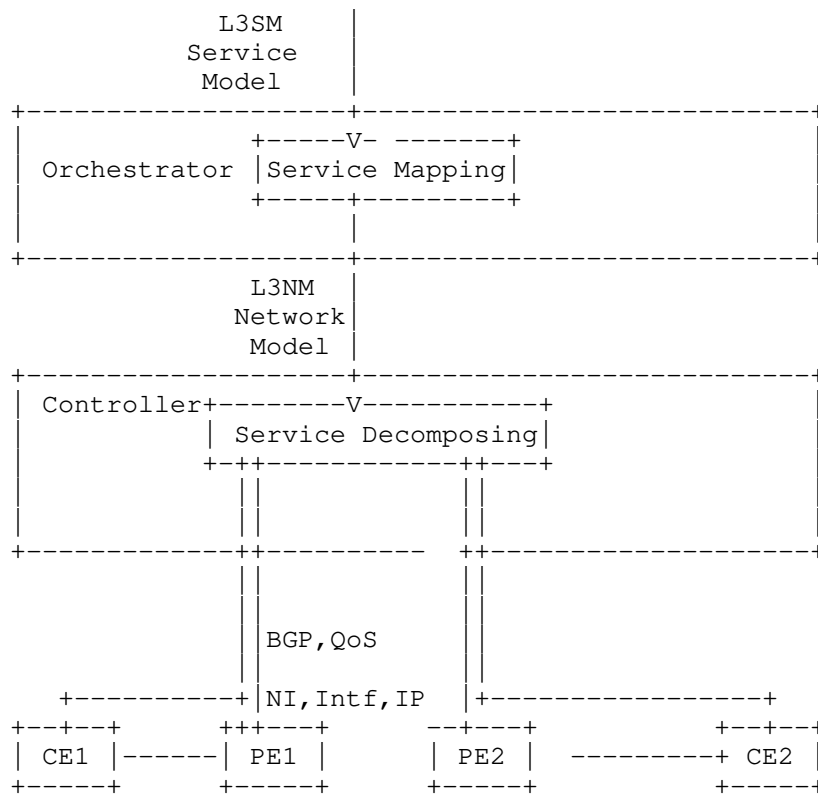


Figure 2: L3VPN Service Delivery Example

In reference to Figure 2, the following steps are performed to deliver the L3VPN service within the network management automation architecture defined in this document:

1. Customer Requests to create two sites based on L3SM Service model with each having one network access connectivity:

Site A: Network-Access A, Bandwidth=20M, for class "foo",
guaranteed-bw-percent = 10, One-Way-Delay=70 msec

Site B: Network-Access B, Bandwidth=30M, for class "foo1",
guaranteed-bw-percent = 15, One-Way-Delay=60 msec

2. The Orchestrator extracts the service parameters from the L3SM model. Then, it uses them as input to translate them into an orchestrated configuration of network elements (e.g., RD, RT, VRF, etc.) that is part of the L3NM network model.

3. The Controller takes orchestrated configuration parameters in the L3NM network model and translates them into orchestrated configuration of network elements that is part of BGP model, QoS model, Network Instance model, IP management model, interface model, etc.

5.2. VN Lifecycle Management Example

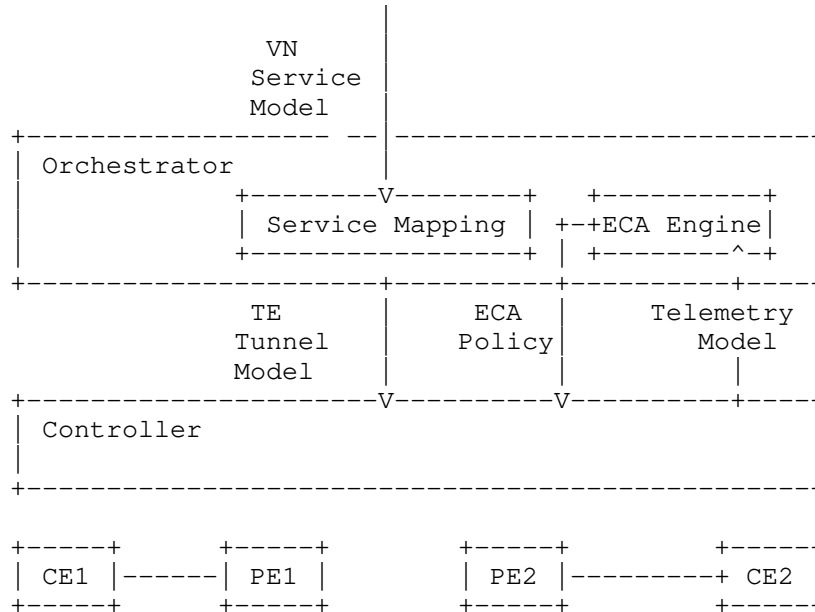


Figure 3

In reference to Figure 3, the following steps are performed to deliver the VN service within the network management automation architecture defined in this document:

1. Customer requests to create 'VN' based on Access point, association between VN and Access point, VN member defined in the VN YANG module.
2. The orchestrator creates the single abstract node topology based on the information captured in an VN YANG module.
3. The Customer exchanges connectivity-matrix on abstract node and explicit path using TE topology model with the orchestrator. This information can be used to instantiate VN and setup tunnels between source and destination endpoints.

4. The telemetry which augments the TEAS VN model and corresponding TE Tunnel model can be used to notify all the parameter changes and network performance change related to VN topology or Tunnel [I-D.ietf-teas-actn-pm-telemetry-autonomics]. This information can be further used as input to ECA engine in the orchestrator and generate ECA policy model to optimize the network.

6. Security Considerations

Security considerations specific to each of the technologies and protocols listed in the document are discussed in the specification documents of each of these techniques.

(Potential) security considerations specific to this document are listed below:

- o Create forwarding loops by mis-configuring the underlying network.
- o Leak sensitive information: special care should be considered when translating between the various layers introduced in the document.
- o ...tbc

7. IANA Considerations

There are no IANA requests or assignments included in this document.

8. Acknowledgements

Thanks to Joe Clark, Greg Mirsky, and Shunsuke Homma for the review.

9. Contributors

Christian Jacquenet
Orange
Rennes, 35000
France
Email: Christian.jacquenet@orange.com

Luis Miguel Contreras Murillo
Telifonica

Email: luismiguel.contrerasmurillo@telefonica.com

Oscar Gonzalez de Dios
Telefonica
Madrid
ES

Email: oscar.gonzalezdedios@telefonica.com

Chongfeng Xie
China Telecom
Beijing
China

Email: xiechf.bri@chinatelecom.cn

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Young Lee
Sung Kyun Kwan University

Email: younglee.tx@gmail.com

10. Informative References

[I-D.arkko-arch-virtualization]

Arkko, J., Tantsura, J., Halpern, J., and B. Varga,
"Considerations on Network Virtualization and Slicing",
draft-arkko-arch-virtualization-01 (work in progress),
March 2018.

[I-D.asechoud-netmod-diffserv-model]

Choudhary, A., Shah, S., Jethanandani, M., Liu, B., and N.
Strahle, "YANG Model for Diffserv", draft-asechoud-netmod-
diffserv-model-03 (work in progress), June 2015.

- [I-D.claccla-netmod-model-catalog]
Clarke, J. and B. Claise, "YANG module for yangcatalog.org", draft-claccla-netmod-model-catalog-03 (work in progress), April 2018.
- [I-D.homma-slice-provision-models]
Homma, S., Nishihara, H., Miyasaka, T., Galis, A., OV, V., Lopez, D., Contreras, L., Ordonez-Lucena, J., Martinez-Julia, P., Qiang, L., Rokui, R., Ciavaglia, L., and X. Foy, "Network Slice Provision Models", draft-homma-slice-provision-models-01 (work in progress), July 2019.
- [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-07 (work in progress), March 2019.
- [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-10 (work in progress), July 2019.
- [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-04 (work in progress), October 2018.
- [I-D.ietf-bfd-yang]
Rahman, R., Zheng, L., Jethanandani, M., Networks, J., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-17 (work in progress), August 2018.
- [I-D.ietf-ccamp-alarm-module]
Vallin, S. and M. Bjorklund, "YANG Alarm Module", draft-ietf-ccamp-alarm-module-09 (work in progress), April 2019.
- [I-D.ietf-ccamp-flexigrid-media-channel-yang]
Madrid, U., Perdices, D., Lopezalvarez, V., Dios, O., King, D., Lee, Y., and G. Galimberti, "YANG data model for Flexi-Grid media-channels", draft-ietf-ccamp-flexigrid-media-channel-yang-02 (work in progress), March 2019.

- [I-D.ietf-ccamp-flexigrid-yang]
Madrid, U., Perdices, D., Lopezalvarez, V., King, D., and Y. Lee, "YANG data model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang-04 (work in progress), July 2019.
- [I-D.ietf-ccamp-llcsm-yang]
Lee, Y., Lee, K., Zheng, H., Dhody, D., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-llcsm-yang-10 (work in progress), September 2019.
- [I-D.ietf-ccamp-mw-yang]
Ahlberg, J., Ye, M., Li, X., Spreafico, D., and M. Vaupotic, "A YANG Data Model for Microwave Radio Link", draft-ietf-ccamp-mw-yang-13 (work in progress), November 2018.
- [I-D.ietf-ccamp-otn-topo-yang]
Zheng, H., Guo, A., Busi, I., Sharma, A., Liu, X., Belotti, S., Xu, Y., Wang, L., and O. Dios, "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang-08 (work in progress), September 2019.
- [I-D.ietf-ccamp-otn-tunnel-model]
Zheng, H., Busi, I., Belotti, S., Lopezalvarez, V., and Y. Xu, "OTN Tunnel YANG Model", draft-ietf-ccamp-otn-tunnel-model-08 (work in progress), October 2019.
- [I-D.ietf-ccamp-wson-tunnel-model]
Lee, Y., Zheng, H., Guo, A., Lopezalvarez, V., King, D., Yoon, B., and R. Vilata, "A Yang Data Model for WSON Tunnel", draft-ietf-ccamp-wson-tunnel-model-04 (work in progress), September 2019.
- [I-D.ietf-dots-data-channel]
Boucadair, M. and R. K, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel Specification", draft-ietf-dots-data-channel-31 (work in progress), July 2019.
- [I-D.ietf-dots-signal-channel]
K, R., Boucadair, M., Patil, P., Mortensen, A., and N. Teague, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification", draft-ietf-dots-signal-channel-37 (work in progress), July 2019.

- [I-D.ietf-idr-bgp-model]
Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP YANG Model for Service Provider Networks", draft-ietf-idr-bgp-model-07 (work in progress), October 2019.
- [I-D.ietf-ippm-stamp-yang]
Mirsky, G., Xiao, M., and W. Luo, "Simple Two-way Active Measurement Protocol (STAMP) Data Model", draft-ietf-ippm-stamp-yang-05 (work in progress), October 2019.
- [I-D.ietf-ippm-twamp-yang]
Civil, R., Morton, A., Rahman, R., Jethanandani, M., and K. Pentikousis, "Two-Way Active Measurement Protocol (TWAMP) Data Model", draft-ietf-ippm-twamp-yang-13 (work in progress), July 2018.
- [I-D.ietf-mpls-base-yang]
Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", draft-ietf-mpls-base-yang-11 (work in progress), September 2019.
- [I-D.ietf-pim-igmp-ml-d-snooping-yang]
Zhao, H., Liu, X., Liu, Y., Sivakumar, M., and A. Peter, "A Yang Data Model for IGMP and MLD Snooping", draft-ietf-pim-igmp-ml-d-snooping-yang-08 (work in progress), June 2019.
- [I-D.ietf-pim-igmp-ml-d-yang]
Liu, X., Guo, F., Sivakumar, M., McAllister, P., and A. Peter, "A YANG Data Model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", draft-ietf-pim-igmp-ml-d-yang-15 (work in progress), June 2019.
- [I-D.ietf-pim-yang]
Liu, X., McAllister, P., Peter, A., Sivakumar, M., Liu, Y., and f. hu, "A YANG Data Model for Protocol Independent Multicast (PIM)", draft-ietf-pim-yang-17 (work in progress), May 2018.
- [I-D.ietf-rtgwg-device-model]
Lindem, A., Berger, L., Bogdanovic, D., and C. Hopps, "Network Device YANG Logical Organization", draft-ietf-rtgwg-device-model-02 (work in progress), March 2017.

- [I-D.ietf-rtgwg-policy-model]
Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data Model for Routing Policy Management", draft-ietf-rtgwg-policy-model-07 (work in progress), September 2019.
- [I-D.ietf-software-iftunnel]
Boucadair, M., Farrer, I., and R. Asati, "Tunnel Interface Types YANG Module", draft-ietf-software-iftunnel-07 (work in progress), June 2019.
- [I-D.ietf-software-yang]
Farrer, I. and M. Boucadair, "YANG Modules for IPv4-in-IPv6 Address plus Port (A+P) Softwires", draft-ietf-software-yang-16 (work in progress), January 2019.
- [I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-13 (work in progress), July 2019.
- [I-D.ietf-supra-generic-policy-data-model]
Halpern, J. and J. Strassner, "Generic Policy Data Model for Simplified Use of Policy Abstractions (SUPA)", draft-ietf-supra-generic-policy-data-model-04 (work in progress), June 2017.
- [I-D.ietf-teas-actn-vn-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Yoon, "A Yang Data Model for VN Operation", draft-ietf-teas-actn-vn-yang-06 (work in progress), July 2019.
- [I-D.ietf-teas-sf-aware-topo-model]
Bryskin, I., Liu, X., Lee, Y., Guichard, J., Contreras, L., Ceccarelli, D., and J. Tantsura, "SF Aware TE Topology YANG Model", draft-ietf-teas-sf-aware-topo-model-03 (work in progress), March 2019.
- [I-D.ietf-teas-te-service-mapping-yang]
Lee, Y., Dhody, D., Fioccola, G., WU, Q., Ceccarelli, D., and J. Tantsura, "Traffic Engineering (TE) and Service Mapping Yang Model", draft-ietf-teas-te-service-mapping-yang-02 (work in progress), September 2019.
- [I-D.ietf-teas-yang-l3-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Layer 3 TE Topologies", draft-ietf-teas-yang-l3-te-topo-05 (work in progress), July 2019.

- [I-D.ietf-teas-yang-path-computation]
Busi, I. and S. Belotti, "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation-06 (work in progress), July 2019.
- [I-D.ietf-teas-yang-rsvp-te]
Beeram, V., Saad, T., Gandhi, R., Liu, X., Bryskin, I., and H. Shah, "A YANG Data Model for RSVP-TE Protocol", draft-ietf-teas-yang-rsvp-te-07 (work in progress), July 2019.
- [I-D.ietf-teas-yang-sr-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and S. Litkowski, "YANG Data Model for SR and SR TE Topologies", draft-ietf-teas-yang-sr-te-topo-05 (work in progress), July 2019.
- [I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-21 (work in progress), April 2019.
- [I-D.ietf-teas-yang-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.
- [I-D.ietf-trill-yang-oam]
Kumar, D., Senevirathne, T., Finn, N., Salam, S., Xia, L., and H. Weiguo, "YANG Data Model for TRILL Operations, Administration, and Maintenance (OAM)", draft-ietf-trill-yang-oam-05 (work in progress), March 2017.
- [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>>.
- [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>>.
- [RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <<https://www.rfc-editor.org/info/rfc4761>>.

- [RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <<https://www.rfc-editor.org/info/rfc4762>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <<https://www.rfc-editor.org/info/rfc7149>>.
- [RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <<https://www.rfc-editor.org/info/rfc7276>>.
- [RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <<https://www.rfc-editor.org/info/rfc7297>>.
- [RFC7455] Senevirathne, T., Finn, N., Salam, S., Kumar, D., Eastlake 3rd, D., Aldrin, S., and Y. Li, "Transparent Interconnection of Lots of Links (TRILL): Fault Management", RFC 7455, DOI 10.17487/RFC7455, March 2015, <<https://www.rfc-editor.org/info/rfc7455>>.
- [RFC8077] Martini, L., Ed. and G. Heron, Ed., "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", STD 84, RFC 8077, DOI 10.17487/RFC8077, February 2017, <<https://www.rfc-editor.org/info/rfc8077>>.
- [RFC8194] Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for LMAP Measurement Agents", RFC 8194, DOI 10.17487/RFC8194, August 2017, <<https://www.rfc-editor.org/info/rfc8194>>.
- [RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <<https://www.rfc-editor.org/info/rfc8199>>.
- [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>>.
- [RFC8328] Liu, W., Xie, C., Strassner, J., Karagiannis, G., Klyus, M., Bi, J., Cheng, Y., and D. Zhang, "Policy-Based Management Framework for the Simplified Use of Policy Abstractions (SUPA)", RFC 8328, DOI 10.17487/RFC8328, March 2018, <<https://www.rfc-editor.org/info/rfc8328>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.
- [RFC8512] Boucadair, M., Ed., Sivakumar, S., Jacquenet, C., Vinapamula, S., and Q. Wu, "A YANG Module for Network Address Translation (NAT) and Network Prefix Translation (NPT)", RFC 8512, DOI 10.17487/RFC8512, January 2019, <<https://www.rfc-editor.org/info/rfc8512>>.
- [RFC8513] Boucadair, M., Jacquenet, C., and S. Sivakumar, "A YANG Data Model for Dual-Stack Lite (DS-Lite)", RFC 8513, DOI 10.17487/RFC8513, January 2019, <<https://www.rfc-editor.org/info/rfc8513>>.
- [RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <<https://www.rfc-editor.org/info/rfc8519>>.

- [RFC8528] Bjorklund, M. and L. Lhotka, "YANG Schema Mount", RFC 8528, DOI 10.17487/RFC8528, March 2019, <<https://www.rfc-editor.org/info/rfc8528>>.
- [RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <<https://www.rfc-editor.org/info/rfc8529>>.
- [RFC8530] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", RFC 8530, DOI 10.17487/RFC8530, March 2019, <<https://www.rfc-editor.org/info/rfc8530>>.
- [RFC8531] Kumar, D., Wu, Q., and Z. Wang, "Generic YANG Data Model for Connection-Oriented Operations, Administration, and Maintenance (OAM) Protocols", RFC 8531, DOI 10.17487/RFC8531, April 2019, <<https://www.rfc-editor.org/info/rfc8531>>.
- [RFC8532] Kumar, D., Wang, Z., Wu, Q., Ed., Rahman, R., and S. Raghavan, "Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8532, DOI 10.17487/RFC8532, April 2019, <<https://www.rfc-editor.org/info/rfc8532>>.
- [RFC8533] Kumar, D., Wang, M., Wu, Q., Ed., Rahman, R., and S. Raghavan, "A YANG Data Model for Retrieval Methods for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8533, DOI 10.17487/RFC8533, April 2019, <<https://www.rfc-editor.org/info/rfc8533>>.

Appendix A. Layered YANG Modules Example Overview

It is not the intent of this document to provide an inventory of tools and mechanisms used in specific network and service management domains; such inventory can be found in documents such as [RFC7276].

A.1. Service Models: Definition and Samples

As described in [RFC8309], the service is "some form of connectivity between customer sites and the Internet and/or between customer sites across the network operator's network and across the Internet". More concretely, an IP connectivity service can be defined as the IP transfer capability characterized by a (Source Nets, Destination Nets, Guarantees, Scope) tuple where "Source Nets" is a group of

unicast IP addresses, "Destination Nets" is a group of IP unicast and/or multicast addresses, and "Guarantees" reflects the guarantees (expressed in terms of Quality Of Service (QoS), performance, and availability, for example) to properly forward traffic to the said "Destination" [RFC7297].

For example:

- o L3SM model [RFC8299] defines the L3VPN service ordered by a customer from a network operator.
- o L2SM model [RFC8466] defines the L2VPN service ordered by a customer from a network operator.
- o VN model [I-D.ietf-teas-actn-vn-yang] provides a YANG data model generally applicable to any mode of Virtual Network (VN) operation.

A.2. Network Models: Definitions and Samples

Figure 4 depicts a set of Network models such as topology models or tunnel models:

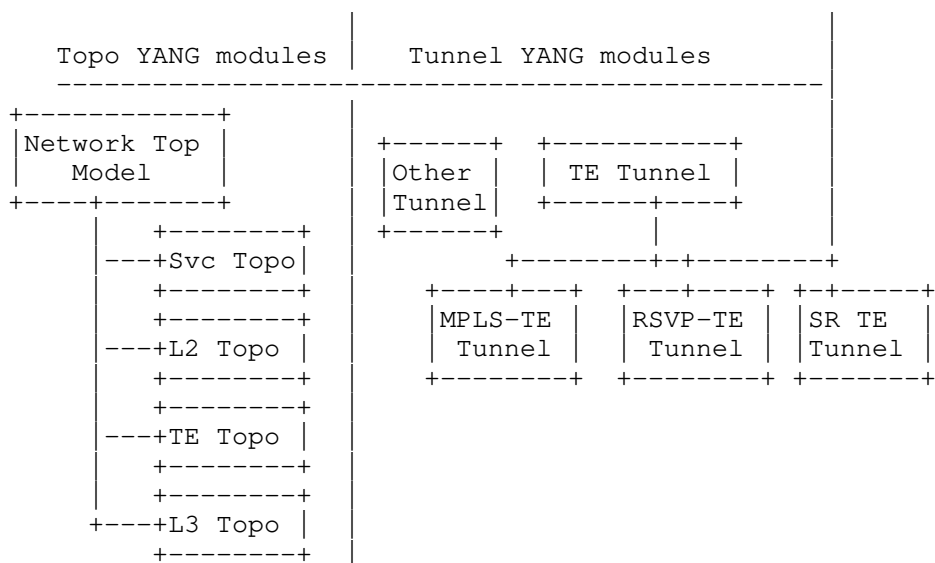


Figure 4: Sample Resource Facing Network Models

Topology YANG module Examples:

- o Network Topology Models: [RFC8345] defines a base model for network topology and inventories. Network topology data include link resource, node resource, and terminate-point resources.
- o TE Topology Models: [I.D-ietf-teas-yang-te-topo] defines a data model for representing and manipulating TE topologies.

This module is extended from network topology model defined in [RFC8345] with TE topologies specifics. This model contains technology-agnostic TE Topology building blocks that can be augmented and used by other technology-specific TE Topology models.

- o L3 Topology Models

[RFC8346] defines a data model for representing and manipulating L3 Topologies. This model is extended from the network topology model defined in [RFC8345] with L3 topologies specifics.

- o L2 Topology Models

[I.D-ietf-i2rs-yang-l2-topology] defines a data model for representing and manipulating L2 Topologies. This model is extended from the network topology model defined in [RFC8345] with L2 topologies specifics.

Tunnel YANG module Examples:

- o Tunnel identities [I-D.ietf-softwire-iftunnel] to ease manipulating extensions to specific tunnels.
- o TE Tunnel Model

[I.D-ietf-teas-yang-te] defines a YANG module for the configuration and management of TE interfaces, tunnels and LSPs.

- o SR TE Tunnel Model

[I.D-ietf-teas-yang-te] augments the TE generic and MPLS-TE model(s) and defines a YANG module for Segment Routing (SR) TE specific data.

- o MPLS TE Model

[I.D-ietf-teas-yang-te] augments the TE generic and MPLS-TE model(s) and defines a YANG module for MPLS TE configurations, state, RPC and notifications.

- o RSVP-TE MPLS Model

[I.D-ietf-teas-yang-rsvp-te] augments the RSVP-TE generic module with parameters to configure and manage signaling of MPLS RSVP-TE LSPs.

Other Network Models:

- o Path Computation API Model

[I.D-ietf-teas-path-computation] YANG module for a stateless RPC which complements the stateful solution defined in [I.D-ietf-teas-yang-te].

- o OAM Models (including Fault Management (FM) and Performance Monitoring)

[RFC8532] defines a base YANG module for the management of OAM protocols that use Connectionless Communications. [RFC8533] defines a retrieval method YANG module for connectionless OAM protocols. [RFC8531] defines a base YANG module for connection oriented OAM protocols. These three models are intended to provide consistent reporting, configuration and representation for connection-less OAM and Connection oriented OAM separately.

Alarm monitoring is a fundamental part of monitoring the network. Raw alarms from devices do not always tell the status of the network services or necessarily point to the root cause. [I.D-ietf-ccamp-alarm-module] defines a YANG module for alarm management.

- o Generic Policy Model

The Simplified Use of Policy Abstractions (SUPA) policy-based management framework [RFC8328] defines base YANG modules [I-D.ietf-sup-generic-policy-data-model] to encode policy. These models point to device-, technology-, and service-specific YANG modules developed elsewhere. Policy rules within an operator's environment can be used to express high-level, possibly network-wide, policies to a network management function (within a controller, an orchestrator, or a network element). The network management function can then control the configuration and/or monitoring of network elements and services. This document describes the SUPA basic framework, its elements, and interfaces.

A.3. Device Models: Definitions and Samples

Network Element models (Figure 5) are used to describe how a service can be implemented by activating and tweaking a set of functions (enabled in one or multiple devices, or hosted in cloud infrastructures) that are involved in the service delivery. The following figure uses IETF defined models as an example.

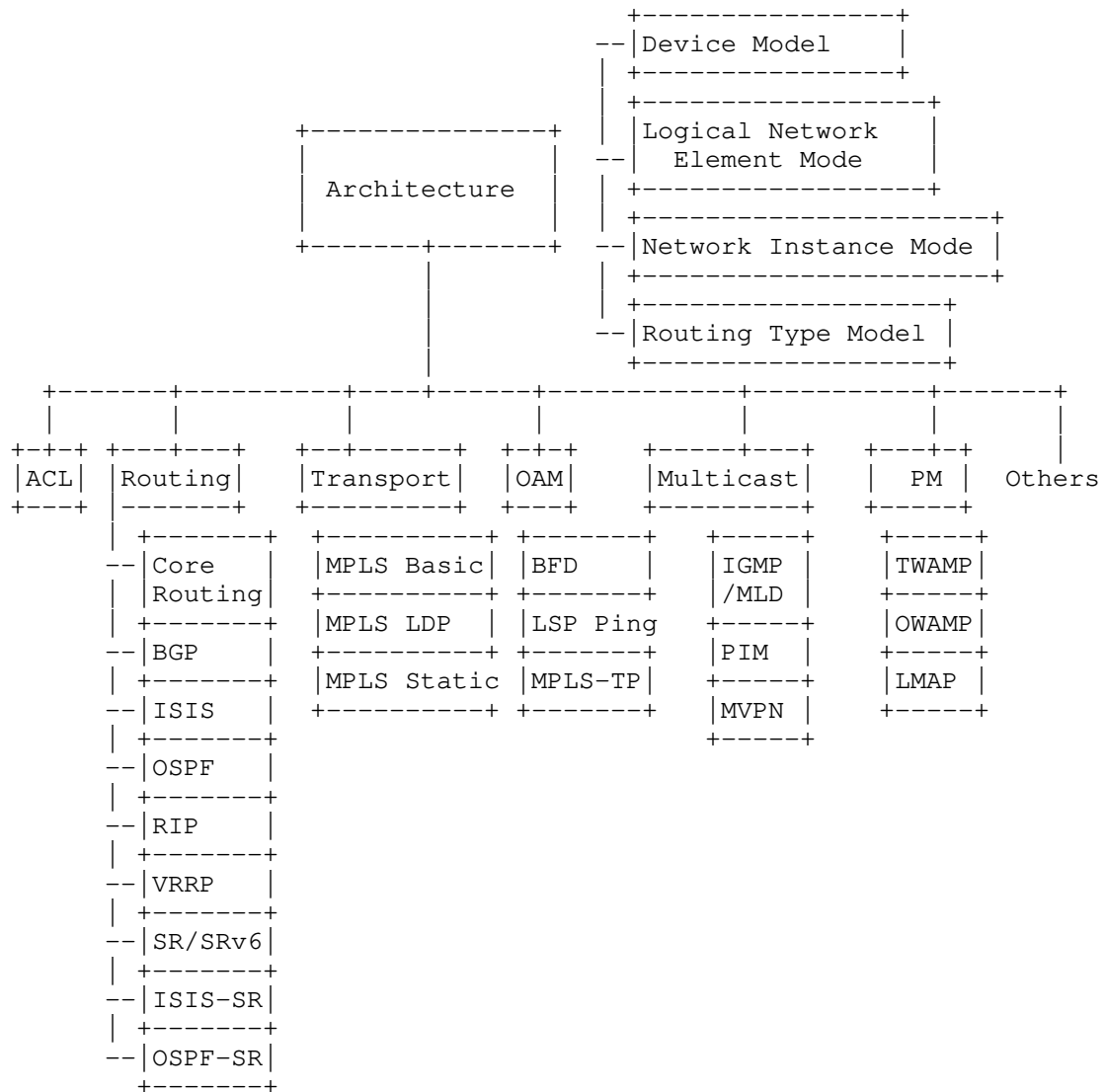


Figure 5: Network Element Modules Overview

A.3.1. Model Composition

- o Device Model

[I.D-ietf-rtgwg-device-model] presents an approach for organizing YANG modules in a comprehensive logical structure that may be used to configure and operate network devices. The structure is itself represented as an example YANG module, with all of the related component models logically organized in a way that is operationally intuitive, but this model is not expected to be implemented.

- o Logical Network Element Model

[RFC8530] defines a logical network element module which can be used to manage the logical resource partitioning that may be present on a network device. Examples of common industry terms for logical resource partitioning are Logical Systems or Logical Routers.

- o Network Instance Model

[RFC8529] defines a network instance module. This module can be used to manage the virtual resource partitioning that may be present on a network device. Examples of common industry terms for virtual resource partitioning are Virtual Routing and Forwarding (VRF) instances and Virtual Switch Instances (VSIs).

A.3.1.1. Schema Mount

Modularity and extensibility were among the leading design principles of the YANG data modeling language. As a result, the same YANG module can be combined with various sets of other modules and thus form a data model that is tailored to meet the requirements of a specific use case. [RFC8528] defines a mechanism, denoted schema mount, that allows for mounting one data model consisting of any number of YANG modules at a specified location of another (parent) schema.

That capability does not cover design time.

A.3.2. Device Models: Definitions and Samples

BGP: [I-D.ietf-idr-bgp-yang-model] defines a YANG module for configuring and managing BGP, including protocol, policy, and operational aspects based on data center, carrier and content provider operational requirements.

- MPLS: [I-D.ietf-mpls-base-yang] defines a base model for MPLS which serves as a base framework for configuring and managing an MPLS switching subsystem. It is expected that other MPLS technology YANG modules (e.g. MPLS LSP Static, LDP or RSVP-TE models) will augment the MPLS base YANG module.
- QoS: [I-D.asechoud-netmod-diffserv-model] describes a YANG module of Differentiated Services for configuration and operations.
- ACL: Access Control List (ACL) is one of the basic elements used to configure device forwarding behavior. It is used in many networking technologies such as Policy Based Routing, Firewalls, etc. [RFC8519] describes a data model of Access Control List (ACL) basic building blocks.
- NAT: For the sake of network automation and the need for programming Network Address Translation (NAT) function in particular, a data model for configuring and managing the NAT is essential. [RFC8512] defines a YANG module for the NAT function covering a variety of NAT flavors such as Network Address Translation from IPv4 to IPv4 (NAT44), Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers (NAT64), customer-side translator (CLAT), Stateless IP/ICMP Translation (SIIT), Explicit Address Mappings (EAM) for SIIT, IPv6-to-IPv6 Network Prefix Translation (NPTv6), and Destination NAT. [RFC8513] specifies a YANG module for the DS-Lite AFTR.
- Stateless Address Sharing: [I-D.ietf-softwire-yang] specifies a YANG module for A+P address sharing, including Lightweight 4over6, Mapping of Address and Port with Encapsulation (MAP-E), and Mapping of Address and Port using Translation (MAP-T) softwire mechanisms.
- Multicast: [I-D.ietf-pim-yang] defines a YANG module that can be used to configure and manage Protocol Independent Multicast (PIM) devices. [I-D.ietf-pim-igmp-mld-yang] defines a YANG module that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) devices. [I-D.ietf-pim-igmp-mld-snooping-yang] defines a YANG module that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping devices.

EVPN: [I-D.ietf-bess-evpn-yang] defines a YANG module for Ethernet VPN services. The model is agnostic of the underlay. It apply to MPLS as well as to VxLAN encapsulation. The model is also agnostic of the services including E-LAN, E-LINE and E-TREE services. This document mainly focuses on EVPN and Ethernet-Segment instance framework.

L3VPN: [I-D.ietf-bess-l3vpn-yang] defines a YANG module that can be used to configure and manage BGP L3VPNs [RFC4364]. It contains VRF specific parameters as well as BGP specific parameters applicable for L3VPNs.

L2VPN: [I-D.ietf-bess-l2vpn-yang] defines a YANG module for MPLS based Layer 2 VPN services (L2VPN) [RFC4664] and includes switching between the local attachment circuits. The L2VPN model covers point-to-point VPWS and Multipoint VPLS services. These services use signaling of Pseudowires across MPLS networks using LDP [RFC8077][RFC4762] or BGP [RFC4761].

Routing Policy: [I-D.ietf-rtgwg-policy-model] defines a YANG module for configuring and managing routing policies in a vendor-neutral way and based on actual operational practice. The model provides a generic policy framework which can be augmented with protocol-specific policy configuration.

BFD: [I-D.ietf-bfd-yang] defines a YANG module that can be used to configure and manage Bidirectional Forwarding Detection (BFD) [RFC5880]. BFD is a network protocol which is used for liveness detection of arbitrary paths between systems.

SR/SRv6: [I-D.ietf-spring-sr-yang] a YANG module for segment routing configuration and operation. [I-D.raza-spring-srv6-yang] defines a YANG module for Segment Routing IPv6 (SRv6) base. The model serves as a base framework for configuring and managing an SRv6 subsystem and expected to be augmented by other SRv6 technology models accordingly.

Core Routing: [RFC8349] defines the core routing data model, which is intended as a basis for future data model development covering more-sophisticated routing systems. It is expected that other Routing technology YANG modules (e.g., VRRP, RIP, ISIS, OSPF models) will augment the Core Routing base YANG module.

PM:

[I.D-ietf-ippm-twamp-yang] defines a data model for client and server implementations of the Two-Way Active Measurement Protocol (TWAMP).

[I.D-ietf-ippm-stamp-yang] defines the data model for implementations of Session-Sender and Session-Reflector for Simple Two-way Active Measurement Protocol (STAMP) mode using YANG.

[RFC8194] defines a data model for Large-Scale Measurement Platforms (LMAPs).

Authors' Addresses

Qin Wu (editor)
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Mohamed Boucadair (editor)
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Diego R. Lopez
Telefonica I+D
Spain

Email: diego.r.lopez@telefonica.com

Chongfeng Xie
China Telecom
Beijing
China

Email: xiechf.bri@chinatelecom.cn

Liang Geng
China Mobile

Email: gengliang@chinamobile.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 22, 2019

G. Zheng
M. Wang
B. Wu
Huawei
June 20, 2019

Yang data model for TACACS+
draft-zheng-opsawg-tacacs-yang-02

Abstract

This document defines a YANG modules that augment the System data model defined in the RFC 7317 with TACACS+ client model. The data model of Terminal Access Controller Access Control System Plus (TACACS+) client allows the configuration of TACACS+ servers for centralized Authentication, Authorization and Accounting.

The YANG modules in this document conforms to the Network Management Datastore Architecture (NMDA) defined in RFC 8342.

Status of This Memo

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

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

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

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

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Tree Diagrams	3
3. TACACS+ Client Model	3
4. TACACS+ Client Module	5
5. Security Considerations	11
6. IANA Considerations	11
7. Acknowledgments	12
8. References	12
8.1. Normative References	12
8.2. Informative References	13
Authors' Addresses	13

1. Introduction

This document defines a YANG modules that augment the System data model defined in the [RFC7317] with TACACS+ client model.

TACACS+ provides Device Administration for routers, network access servers and other networked computing devices via one or more centralized servers which is defined in the TACACS+ Protocol. [I-D.ietf-opsawg-tacacs]

The System Management Model [RFC7317] defines two YANG features to support local or RADIUS authentication:

- o User Authentication Model: Define a list of usernames and passwords and control the order in which local or RADIUS authentication is used.
- o RADIUS Client Model: Defines a list of RADIUS server that a device used.

Since TACACS+ is also used for device management and the feature is not contained in the system model, this document defines a YANG data model that allows users to configure TACACS+ client functions on a device for centralized Authentication, Authorization and Accounting provided by TACACS+ servers.

The YANG models can be used with network management protocols such as NETCONF[RFC6241] to install, manipulate, and delete the configuration of network devices.

The YANG data model in this document conforms to the Network Management Datastore Architecture (NMDA) defined in [RFC8342].

2. Conventions used in this document

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC6241] and are used in this specification:

- o client
- o configuration data
- o server
- o state data

The following terms are defined in [RFC7950] and are used in this specification:

- o augment
- o data model
- o data node

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

2.1. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

3. TACACS+ Client Model

This model is used to configure TACACS+ client on the device to support deployment scenarios with centralized authentication, authorization, and accounting servers. Authentication is used to

validates a user's name and password, authorization allows the user to access and execute commands at various command levels assigned to the user and accounting keeps track of the activity of a user who has accessed the device.

The `ietf-system-tacacsplus` module is intended to augment the `/sys:system` path defined in the `ietf-system` module with `"tacacsplus"` grouping. Therefore, a device can use local, Remote Authentication Dial In User Service (RADIUS), or Terminal Access Controller Access Control System Plus (TACACS+) to validate users who attempt to access the router by several mechanisms, e.g. a command line interface or a web-based user interface.

The `"server"` list is directly under the `"tacacsplus"` container, which is to hold a list of different TACACS+ server and use `server-type` to distinguish the three protocols. The list of servers is for redundancy purpose.

Most of the parameters in the `"server"` list are taken directly from the TACACS+ protocol [I-D.ietf-opsawg-tacacs], and some are derived from the wide implementation of network equipment manufacturers. For example, when there are multiple interfaces connected to the TACACS+ server, the source address of outgoing TACACS+ packets could be specified, or the source address could be specified through the interface setting. For the TACACS+ server located in a private network, a VRF instance needs to be specified.

The `"statistics"` container under the `"server list"` is to record session statistics and usage information during user access which include the amount of data a user has sent and/or received during a session.

The data model for TACACS+ client has the following structure:

```

module: ietf-system-tacacsplus
augment /sys:system:
  +--rw tacacsplus {tacacsplus}?
    +--rw server* [name]
      +--rw name string
      +--rw server-type? enumeration
      +--rw address inet:host
      +--rw port? inet:port-number
      +--rw shared-secret string
      +--rw (source-type)?
        | +--:(source-ip)
        | | +--rw source-ip? inet:ip-address
        | +--:(source-interface)
        | | +--rw source-interface? if:interface-ref
      +--rw single-connection? boolean
      +--rw timeout? uint16
      +--rw vrf-instance?
        | -> /ni:network-instances/network-instance/name
      +--ro statistics
        +--ro connection-opens? yang:counter64
        +--ro connection-closes? yang:counter64
        +--ro connection-aborts? yang:counter64
        +--ro connection-failures? yang:counter64
        +--ro connection-timeouts? yang:counter64
        +--ro messages-sent? yang:counter64
        +--ro messages-received? yang:counter64
        +--ro errors-received? yang:counter64

```

4. TACACS+ Client Module

<CODE BEGINS> file "ietf-system-tacacsplus@2019-06-20.yang"

```

module ietf-system-tacacsplus {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-system-tacacsplus";
  prefix sys-tacsplus;

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

```

```
    "RFC 8529: YANG Data Model for Network Instances";
}
import ietf-interfaces {
  prefix if;
  reference
    "RFC 8343: A YANG Data Model for Interface Management";
}
import ietf-system {
  prefix sys;
  reference "RFC 7317: A YANG Data Model for System Management";
}
import ietf-netconf-acm {
  prefix nacm;
  reference "RFC 8341: Network Configuration Access Control Model";
}

organization
  "IETF Opsawg (Operations and Management Area Working Group)";
contact
  "WG Web:    <http://tools.ietf.org/wg/opsawg/>
  WG List:    <mailto:opsawg@ietf.org>

  Editor:     Guangying Zheng
              <mailto:zhengguangying@huawei.com>";
description
  "This module provides configuration of TACACS+ client.

  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).

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

revision 2019-06-20 {
  description
    "Initial revision.";
  reference "foo";
}

feature tacacsplus {
  description
```



```
    "Indicates that the device can be configured as a TACACS+
      client.";
    reference "draft-ietf-opsawg-tacacs-11: The TACACS+ Protocol";
  }

  grouping statistics {
    description
      "Grouping for TACACS+ packets statistics attributes";
    container statistics {
      config false;
      description
        "A collection of server-related statistics objects";
      leaf connection-opens {
        type yang:counter64;
        description
          "Number of new connection requests sent to the server, e.g.
            socket open";
      }
      leaf connection-closes {
        type yang:counter64;
        description
          "Number of connection close requests sent to the server, e.g.
            socket close";
      }
      leaf connection-aborts {
        type yang:counter64;
        description
          "Number of aborted connections to the server. These do
            not include connections that are close gracefully.";
      }
      leaf connection-failures {
        type yang:counter64;
        description
          "Number of connection failures to the server";
      }
      leaf connection-timeouts {
        type yang:counter64;
        description
          "Number of connection timeouts to the server";
      }
      leaf messages-sent {
        type yang:counter64;
        description
          "Number of messages sent to the server";
      }
      leaf messages-received {
        type yang:counter64;
        description
```

```
        "Number of messages received by the server";
    }
    leaf errors-received {
        type yang:counter64;
        description
            "Number of error messages received from the server";
    }
}

grouping tacacsplus {
    description
        "Grouping for TACACS+ attributes";
    container tacacsplus {
        if-feature "tacacsplus";
        description
            "Container for TACACS+ configurations and operations.";
        list server {
            key "name";
            ordered-by user;
            description
                "List of TACACS+ servers used by the device

                When the TACACS+ client is invoked by a calling
                application, it sends the query to the first server in
                this list.  If no response has been received within
                'timeout' seconds, the client continues with the next
                server in the list.  If no response is received from any
                server, the client continues with the first server again.
                When the client has traversed the list 'attempts' times
                without receiving any response, it gives up and returns an
                error to the calling application.";

            leaf name {
                type string;
                description
                    "An arbitrary name for the TACACS+ server.";
            }
            leaf server-type {
                type enumeration {
                    enum authentication {
                        description
                            "The server is an authentication server.";
                    }
                    enum authorization {
                        description
                            "The server is an authorization server.";
                    }
                    enum accounting {
```

```
        description
            "The server is an accounting server.";
    }
}
description
    "Server type: authentication/authorization/accounting.";
}
leaf address {
    type inet:host;
    mandatory true;
    description
        "The address of the TACACS+ server.";
}
leaf port {
    type inet:port-number;
    default "49";
    description
        "The port number of TACACS+ Server port.";
}
leaf shared-secret {
    type string;
    mandatory true;
    nacm:default-deny-all;
    description
        "The shared secret, which is known to both the
        TACACS+ client and server. TACACS+ server administrators
        SHOULD configure secret keys of minimum
        16 characters length.";
    reference "TACACS+ protocol:";
}
choice source-type {
    description
        "The source address type for outbound TACACS+ packets.";
    case source-ip {
        leaf source-ip {
            type inet:ip-address;
            description
                "Specifies source IP address for TACACS+ outbound
                packets.";
        }
    }
    case source-interface {
        leaf source-interface {
            type if:interface-ref;
            description
                "Specifies the interface from which the IP address is
                derived for use as the source for the outbound TACACS+
                packet";
        }
    }
}
```

```
    }
  }
}
leaf single-connection {
  type boolean;
  default "false";
  description
    "Whether the single connection mode is enabled for the
    server. By default, the single connection mode is
    disabled.";
}
leaf timeout {
  type uint16 {
    range "1..300";
  }
  units "seconds";
  default "5";
  description
    "The number of seconds the device will wait for a
    response from each TACACS+ server before trying with a
    different server.";
}
leaf vrf-instance {
  type leafref {
    path "/ni:network-instances/ni:network-instance/ni:name";
  }
  description
    "Specifies the VPN Routing and Forwarding (VRF) instance to
    use to communicate with the TACACS+ server.";
}

uses statistics;
}
}
}

augment "/sys:system" {
  description
    "Augment the system model with authorization and accounting
    attributes
    Augment the system model with the tacacsplus model";
  uses tacacsplus;
}
}

<CODE ENDS>
```

5. Security Considerations

The YANG module defined in this document is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

This document describes the use of TACACS+ for purposes of authentication, authorization and accounting, it is vulnerable to all of the threats that are present in TACACS+ applications. For a discussion of such threats, see Section 9 of the TACACS+ Protocol [I-D.ietf-opsawg-tacacs].

6. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-system-tacacsplus
Registrant Contact: The IESG.
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-system-tacacsplus
Namespace: urn:ietf:params:xml:ns:yang:ietf-tacacsplus
Prefix: sys-tacsplus
Reference: RFC XXXX

7. Acknowledgments

The authors wish to thank Alex Campbell and Ebben Aries, Alan DeKok, Joe Clarke, many others for their helpful comments.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", RFC 7317, DOI 10.17487/RFC7317, August 2014, <<https://www.rfc-editor.org/info/rfc7317>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

8.2. Informative References

- [I-D.ietf-opsawg-tacacs]
Dahm, T., Ota, A., dcmgash@cisco.com, d., Carrel, D., and L. Grant, "The TACACS+ Protocol", draft-ietf-opsawg-tacacs-13 (work in progress), March 2019.

Authors' Addresses

Guangying Zheng
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: zhengguangying@huawei.com

Michael Wang
Huawei Technologies, Co., Ltd
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: wangzitao@huawei.com

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

Email: lana.wubo@huawei.com