Stream: Internet Engineering Task Force (IETF)
RFC: 9375
Category: Standards Track
Published: April 2023
ISSN: 2070-1721
Authors: B. Wu, Ed. (Huawei), Q. Wu, Ed. (Huawei), M. Boucadair, Ed. (Orange), O. Gonzalez de Dios (Telefonica), B. Wen (Comcast)

RFC 9375
A YANG Data Model for Network and VPN Service Performance Monitoring

Abstract

The data model for network topologies defined in RFC 8345 introduces vertical layering relationships between networks that can be augmented to cover network and service topologies. This document defines a YANG module for performance monitoring (PM) of both underlay networks and overlay VPN services that can be used to monitor and manage network performance on the topology of both layers.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9375.

Copyright Notice

Table of Contents

1. Introduction

[RFC8969] describes a framework for automating service and network management with YANG [RFC7950] data models. It states that the performance measurement telemetry model should be tied to the services (such as a Layer 3 VPN or Layer 2 VPN) or to the network models to monitor the overall network performance and the Service Level Agreements (SLAs).

The performance of VPN services is associated with the performance changes of the underlay networks that carry VPN services. For example, link delay between Provider Edge (PE) and Provider (P) devices and packet loss status on Layer 2 and Layer 3 interfaces connecting PEs and Customer Edge (CE) devices directly impact VPN service performance. Additionally, the integration of Layer 2 / Layer 3 VPN performance and network performance data enables the orchestrator to monitor consistently. Therefore, this document defines a YANG module for both network and VPN service performance monitoring (PM). The module can be used to monitor and manage network performance on the topology level or the service topology between VPN sites.

The base model specified in Section 5 can be extended to include technology-specific details, e.g., adding Explicit Congestion Notification (ECN) statistics for Layer 3 networks or VPN services to support performance-sensitive applications.

This document does not introduce new metrics for network performance or mechanisms for measuring network performance, but it uses the existing mechanisms and statistics to monitor the performance of the network and the services.

The YANG module defined in this document is designed as an augmentation to the network topology YANG data model defined in [RFC8345] and draws on relevant YANG types defined in [RFC6991], [RFC8345], [RFC8532], and [RFC9181].

Appendix A provides a set of examples to illustrate the use of the module.

2. Terminology

The following terms are defined in [RFC7950] and are used in this specification:

- augment
- data model
- data node

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

The tree diagrams used in this document follow the notation defined in [RFC8340].

2.1. Acronyms

The following acronyms are used in the document:

CE	Customer Edge, as defined in [RFC4026]
L2VPN	Layer 2 Virtual Private Network, as defined in [RFC4026]
L3VPN	Layer 3 Virtual Private Network, as defined in [RFC4026]
L2NM	L2VPN Network Model
L3NM	L3VPN Network Model
MPLS	Multiprotocol Label Switching
OAM	Operations, Administration, and Maintenance
OSPF	Open Shortest Path First
OWAMP	One-Way Active Measurement Protocol, as defined in [RFC4656]
P	Provider router, as defined in [RFC4026]
PE	Provider Edge, as defined in [RFC4026]
PM	Performance Monitoring
SLA	Service Level Agreement
TP	Termination Point, as defined in [RFC8345], Section 4.2
TWAMP	Two-Way Active Measurement Protocol, as defined in [RFC5357]
VPLS	Virtual Private LAN Service, as defined in [RFC4026]
VPN	Virtual Private Network

3. Network and VPN Service Performance Monitoring Model Usage

Models are key for automating network management operations (Section 3 of [RFC8969]). Particularly, together with service and network models, performance measurement telemetry models are needed to monitor network performance to meet specific service requirements (typically captured in an SLA).

```
                      +---------------+
                      |   Customer    |
                      +-------+-------+
                              |
        Customer Service Models |
                              |
                      +-------+---------+
                      |     Service     |
                      |   Orchestrator  |
                      +------+-+--------+
                             | |
        Network Service Models | | Network and VPN Service PM Models
                             | |
                      +------+-+--------+
                      |     Network     |
                      |   Controller    |
                      +-------+---------+
                              |
        +---------------------+----------------------+
                            Network
```

Figure 1: An Example Architecture with a Service Orchestrator

The network and VPN service PM model can be used to expose operational performance information to the layer above, e.g., to an orchestrator or other Business Support System (BSS) / Operational Support System (OSS) client application, via standard network management APIs. Figure 1 shows an example usage in a layered model architecture as described in [RFC8309].

Before using the model, the controller needs to establish topology visibility of the network and VPN. For example, the controller can use network information from [RFC8345] and [YANG-SAP] or VPN information from the L3VPN Network Model (L3NM) [RFC9182] and the L2VPN Network Model (L2NM) [RFC9291]. Then the controller derives network or VPN performance data by aggregating (and filtering) lower-level data collected via monitoring counters of the devices involved.

The network or VPN performance data can be based on different sources. For example, the performance monitoring data per link in the underlying networks can be collected using a network performance measurement method such as the One-Way Active Measurement Protocol (OWAMP) [RFC4656], Two-Way Active Measurement Protocol (TWAMP) [RFC5357], Simple Two-way Active Measurement Protocol (STAMP) [RFC8762], Multiprotocol Label Switching (MPLS) Loss and Delay Measurement [RFC6374], or In situ OAM (IOAM) [RFC9197]. The performance monitoring information reflecting the quality of the network or VPN service (e.g., network performance data between source node and destination node in the networks or between VPN sites) can be computed and aggregated, for example, using the information from the Traffic Engineering Database (TED) [RFC7471] [RFC8570] [RFC8571] or Large-Scale Measurement Platform (LMAP) [RFC8194].

The measurement and report intervals that are associated with these performance data usually depend on the configuration of the specific measurement method or collection method or various combinations. This document defines network-wide measurement intervals to align measurement requirements for networks or VPN services.

3.1. Collecting Data via the Pub/Sub Mechanism

Some applications, such as service-assurance applications, which must maintain a continuous view of operational data and state, can use the subscription model specified in [RFC8641] to subscribe to the specific network performance data or VPN service performance data they are interested in, at the data source. For example, network or VPN topology updates may be obtained through on-change notifications [RFC8641]. For dynamic PM data (e.g., VPN Routing and Forwarding (VRF) routes or Media Access Control (MAC) entries, link metrics, and interface metrics), various notifications can be specified to obtain more complete data. A periodic notification [RFC8641] can be specified to obtain real-time performance data. For devices/controllers that maintain historical performance data for a period of time, a replay notification (see [RFC5277] or [RFC8639]) can be used to obtain the historical data. And alarm notifications [RFC8632] can be specified to get alarms for the metrics that exceed or fall below the performance threshold.

The data source can then use the network and VPN service performance monitoring model defined in this document and the YANG-Push data model [RFC8641] to distribute specific telemetry data to target recipients.

3.2. Collecting Data On Demand

To obtain a snapshot of performance data from a network topology or a VPN service topology, service-assurance applications may retrieve information using the network and VPN service PM model through a Network Configuration Protocol (NETCONF) [RFC6241] or a RESTCONF [RFC8040] interface. For example, a specified "link-id" of a VPN can be used as a filter in a RESTCONF GET request to retrieve per-link VPN PM data.

4. Description of the YANG Data Model

This document defines the "ietf-network-vpn-pm" YANG module, which is an augmentation to the "ietf-network" and "ietf-network-topology" YANG modules.

4.1. Layering Relationship between Multiple Layers of Topology

[RFC8345] defines a YANG data model for network/service topologies and inventories. The service topology described in [RFC8345] includes the abstract topology for a service layer above Layer 1 (L1), Layer 2 (L2), and Layer 3 (L3) underlay topologies. This service topology has the generic topology elements of node, link, and termination point. One typical example of a service topology is described in Figure 3 of [RFC8345]: two VPN service topologies instantiated over a common L3 topology. Each VPN service topology is mapped onto a subset of nodes from the L3 topology.

Figure 2 illustrates an example of a topology hierarchy that maps between the VPN service topology and an underlying Layer 3 network topology.

```
                     VPN 1                       VPN 2
          +------------------------+   +------------------------+
         /                        /   /                        /
        / S1C_[VN3]..........    /   /                        /
       /         \          :   /   / S2A_[VN1]____[VN3]_S2B /
      /           \         :  /   /        *        *      /
     /             \        :............ * ....     *     /
    / S1B_[VN2]____[VN1]_S1A /   /       *     :     *    /
   +---------:-------:------+   +-------*------:-----*---+
             :        :      * * *  * *        :     *
             :         :   *                   :     *
   Site-1A   :  +-------:-*--------------------:-----*-----+ Site-1C
     [CE1]___:_/_______[N1]___________________[N2]___*____/__[CE3]
             :/       / / \             _____//      *   /
   [CE5]_____:_______/ /    \     _____/     /     *    /
 Site-2A    /:        /       \  /          /    *     /
           / :                [N5]         /   *      /
          /   :     /       __/ \__       /  *       /
         /     :   /    ___/       \__   / *        /
Site-1B /       : / ___/              \ /*         /  Site-2B
[CE2]__/________[N4]__________________[N3]________/____[CE4]
      /                                          /
     +------------------------------------------+
                                   L3 Topology

   Legend:
      N:   Node
      VN:  VPN Node
      S:   Site
      CE:  Customer Edge
      __   Link within a network layer
      :    Mapping between VPN 1 service topology and L3 topology
      *    Mapping between VPN 2 service topology and L3 topology
```

Figure 2: Example of Topology Mapping between VPN Service Topology and an Underlying Network

As shown in Figure 2, two VPN services topologies are built on top of one underlying Layer 3 network:

VPN 1: This service topology supports Hub-and-Spoke communications for "customer 1", connecting the customer's access at three sites: Site-1A, Site-1B, and Site-1C. These sites are connected to nodes that are mapped to node 1 (N1), node 2 (N2), and node 4 (N4) in the underlying Layer 3 network. Site-1A plays the role of Hub while Site-1B and Site-1C are configured as Spokes.

VPN 2: This service topology supports any-to-any communications for "customer 2", connecting the customer's access at two sites: Site-2A and Site-2B. These sites are connected to nodes that are mapped to node 1 (N1) and node 3 (N3) in the underlying Layer 3 network. Site-2A and Site-2B have an "any-to-any" role.

Based on the association between VPN service topologies and underlying network topologies, the Network and VPN Service PM YANG module extends the performance status of the underlay networks and VPN services. For example, the module can provide link PM statistics and port statistics of an underlay network, e.g., Layer 1, Layer 2, Layer 3, and OSPF networks. It can also provide VPN PM statistics, which can be further split into PM for the VPN tunnel and PM at the VPN PE access node, as illustrated in the following diagram.

```
       +-----------------------------------------------------+
       |                                                     |
       |                      VPN2 Link                      |
       |              |<-------------------->|               |
       |              |                      |               |
       |      VPN2+---+---+              +---+---+VPN2       |
       |       TP1| VN1   |  Tunnel PM   |  VN3  |TP2        |
       |       ---+ PE A  |==============|  PE B +----       |
       |vpn-access+-------+              +-------+ vpn-access|
       |-interface|                              | -interface|
       |          |##############################|           |
       |          |inter-vpn-access-interface PM |           |
       |                                                     |
       +-----------------------------------------------------+
       |                                                     |
       |                                                     |
+----+ |        TP+-----+ Link  +---+ Link  +-----+TP        | +----+
| CE4+-+----------+ N1  +-------+-N2+-------+  N3 +----------+-+CE5 |
+----+ |       1-1+-----+1-2 2-1+---+2-2 3-1+-----+3-2       | +----+
       |                                                     |
       |                                                     |
       +-----------------------------------------------------+

       Legend:
         N:  node
         VN: VPN Node
         TP: Termination Point
         -:  Link
```

Figure 3: An Example of VPN PM

Figure 3 illustrates an example of VPN PM and two VPN PM measurement methods including the VPN tunnel PM and the inter-VPN-access interface PM. VPN PM can also provide statistics on VPN access interfaces, the number of current VRF routes, or L2VPN MAC entry of a VPN node.

4.2. Network-Level Performance Monitoring Augmentation

The module described below can be used for performance monitoring for both the underlay networks and the VPN services, which would be separate entries in the network list [RFC8345]. The differences are as follows:

- When the "service" presence container is absent, then it indicates performance monitoring of the network itself.
- When the "service" presence container is present, then it indicates performance monitoring of the VPN service specified by the "service-type" leaf, e.g., L3VPN or Virtual Private LAN Service (VPLS). The values are taken from [RFC9181]. When a network topology instance contains the L3VPN or other L2VPN network types, it represents a VPN instance that can perform performance monitoring.

The YANG tree in Figure 4 is a part of the "ietf-network-vpn-pm" tree. It defines the following set of network-level attributes:

"vpn-id": Refers to an identifier of VPN service defined in [RFC9181]. This identifier is used to correlate the performance status with the network service configuration.

"vpn-service-topology": Indicates the type of VPN service topology. This model supports "any-to-any", "hub-spoke" (where Hubs can exchange traffic), and "hub-spoke-disjoint" (where Hubs cannot exchange traffic), which are taken from [RFC9181]. These VPN service topology types can be used to describe how VPN sites communicate with each other.

```
module: ietf-network-vpn-pm
  augment /nw:networks/nw:network/nw:network-types:
    +--rw service!
       +--rw service-type            identityref
       +--rw vpn-id?                 vpn-common:vpn-id
       +--rw vpn-service-topology?   identityref
```

Figure 4: Network-Level YANG Tree

4.3. Node-Level Performance Monitoring Augmentation

The YANG tree in Figure 5 is the node part of the "ietf-network-vpn-pm" tree.

For network performance monitoring, the module defines the following attributes:

"node-type": Indicates the device type of the PE, P device, or Autonomous System Border Router (ASBR) as defined in [RFC4026] and [RFC4364] so that the performance metric between any two nodes that each have a specific node type can be reported.

"entry-summary": Lists a set of IPv4 statistics, IPv6 statistics, and MAC statistics. The detailed statistics are specified separately.

For VPN service topology, the module defines one attribute:

"role": Defines the role in a particular VPN service topology. The roles are taken from [RFC9181] (e.g., "any-to-any-role", "spoke-role", and "hub-role").

```
augment /nw:networks/nw:network/nw:node:
  +--rw node-type?       identityref
  +--ro entry-summary
     +--ro ipv4-num
     |  +--ro maximum-routes?        uint32
     |  +--ro total-active-routes?   uint32
     +--ro ipv6-num
     |  +--ro maximum-routes?        uint32
     |  +--ro total-active-routes?   uint32
     +--ro mac-num
        +--ro maximum-mac-entries?        uint32
        +--ro total-active-mac-entries?   uint32
augment /nw:networks/nw:network/nw:node:
  +--rw role?   identityref
```

Figure 5: Node-Level YANG Tree

4.4. Performance Monitoring Augmentation at Link and Termination Point Level

The YANG tree in Figure 6 is the link and termination point (TP) part of the "ietf-network-vpn-pm" tree.

The "links" are classified into two types: topology link (defined in [RFC8345]) and abstract link of a VPN between PEs (defined in this module).

The performance data of a link is a collection of counters and gauges that report the performance status. All these metrics are defined as unidirectional metrics.

```
  augment /nw:networks/nw:network/nt:link:
    +--rw perf-mon
       +--rw low-percentile?            percentile
       +--rw intermediate-percentile?   percentile
       +--rw high-percentile?           percentile
       +--rw measurement-interval?      uint32
       +--ro pm* [pm-type]
       |  +--ro pm-type          identityref
       |  +--ro pm-attributes
       |     +--ro start-time?                     yang:date-and-time
       |     +--ro end-time?                       yang:date-and-time
       |     +--ro pm-source?                      identityref
       |     +--ro one-way-pm-statistics
       |     |  +--ro loss-statistics
       |     |  |  +--ro packet-loss-count?   yang:counter64
       |     |  |  +--ro loss-ratio?          percentage
       |     |  +--ro delay-statistics
       |     |  |  +--ro unit-value?                     identityref
       |     |  |  +--ro min-delay-value?                yang:gauge64
       |     |  |  +--ro max-delay-value?                yang:gauge64
       |     |  |  +--ro low-delay-percentile?           yang:gauge64
       |     |  |  +--ro intermediate-delay-percentile?  yang:gauge64
       |     |  |  +--ro high-delay-percentile?          yang:gauge64
       |     |  +--ro jitter-statistics
       |     |     +--ro unit-value?                     identityref
       |     |     +--ro min-jitter-value?               yang:gauge64
       |     |     +--ro max-jitter-value?               yang:gauge64
       |     |     +--ro low-jitter-percentile?          yang:gauge64
       |     |     +--ro intermediate-jitter-percentile? yang:gauge64
       |     |     +--ro high-jitter-percentile?         yang:gauge64
       |     +--ro one-way-pm-statistics-per-class* [class-id]
       |        +--ro class-id             string
       |        +--ro loss-statistics
       |        |  +--ro packet-loss-count?   yang:counter64
       |        |  +--ro loss-ratio?          percentage
       |        +--ro delay-statistics
       |        |  +--ro unit-value?                     identityref
       |        |  +--ro min-delay-value?                yang:gauge64
       |        |  +--ro max-delay-value?                yang:gauge64
       |        |  +--ro low-delay-percentile?           yang:gauge64
       |        |  +--ro intermediate-delay-percentile?  yang:gauge64
       |        |  +--ro high-delay-percentile?          yang:gauge64
       |        +--ro jitter-statistics
       |           +--ro unit-value?                     identityref
       |           +--ro min-jitter-value?               yang:gauge64
       |           +--ro max-jitter-value?               yang:gauge64
       |           +--ro low-jitter-percentile?          yang:gauge64
       |           +--ro intermediate-jitter-percentile? yang:gauge64
       |           +--ro high-jitter-percentile?         yang:gauge64
       +--rw vpn-pm-type
          +--rw inter-vpn-access-interface
          |  +--rw inter-vpn-access-interface?   empty
          +--rw vpn-tunnel!
             +--ro vpn-tunnel-type?   identityref
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +--ro pm-statistics
       +--ro last-updated?               yang:date-and-time
```

```
        +--ro inbound-octets?             yang:counter64
        +--ro inbound-unicast?            yang:counter64
        +--ro inbound-broadcast?          yang:counter64
        +--ro inbound-multicast?          yang:counter64
        +--ro inbound-discards?           yang:counter64
        +--ro inbound-errors?             yang:counter64
        +--ro inbound-unknown-protocol?   yang:counter64
        +--ro outbound-octets?            yang:counter64
        +--ro outbound-unicast?           yang:counter64
        +--ro outbound-broadcast?         yang:counter64
        +--ro outbound-multicast?         yang:counter64
        +--ro outbound-discards?          yang:counter64
        +--ro outbound-errors?            yang:counter64
        +--ro vpn-network-access* [network-access-id]
           +--ro network-access-id           vpn-common:vpn-id
           +--ro last-updated?               yang:date-and-time
           +--ro inbound-octets?             yang:counter64
           +--ro inbound-unicast?            yang:counter64
           +--ro inbound-broadcast?          yang:counter64
           +--ro inbound-multicast?          yang:counter64
           +--ro inbound-discards?           yang:counter64
           +--ro inbound-errors?             yang:counter64
           +--ro inbound-unknown-protocol?   yang:counter64
           +--ro outbound-octets?            yang:counter64
           +--ro outbound-unicast?           yang:counter64
           +--ro outbound-broadcast?         yang:counter64
           +--ro outbound-multicast?         yang:counter64
           +--ro outbound-discards?          yang:counter64
           +--ro outbound-errors?            yang:counter64
```

Figure 6: Link and Termination Point YANG Subtree

For the data nodes of "link" depicted in Figure 6, the YANG module defines the following minimal set of link-level performance attributes:

Percentile parameters: The module supports reporting delay and jitter metrics with percentile values. There are three percentile values for configuring various percentile reporting levels. By default, low percentile (10th percentile), intermediate percentile (50th percentile), and high percentile (90th percentile) are used. Configuring a percentile to 0.000 indicates the client is not interested in receiving a particular percentile. If all percentile nodes are configured to 0.000, it represents that no percentile-related nodes will be reported for a given performance metric (e.g., one-way delay and one-way delay variation) and only peak/min values will be reported. For example, a client can inform the server that it is interested in receiving only high percentiles. Then for a given link at a given "start-time", "end-time", and "measurement-interval", the "high-delay-percentile" and "high-jitter-percentile" will be reported. An example to illustrate the use of percentiles is provided in Appendix A.3.

Measurement interval ("measurement-interval"): Specifies the performance measurement interval, in seconds.

Start time ("start-time"): Indicates the start time of the performance measurement for link statistics.

End time ("end-time"): Indicates the end time of the performance measurement for link statistics.

PM source ("pm-source"): Indicates the performance monitoring source. The data for the topology link can be based, e.g., on BGP - Link State (BGP-LS) [RFC8571]. The statistics of the VPN abstract links can be collected based upon VPN OAM mechanisms, e.g., OAM mechanisms referenced in [RFC9182] or Ethernet service OAM [ITU-T-Y-1731] referenced in [RFC9291]. Alternatively, the data can be based upon the underlay technology OAM mechanisms, e.g., Generic Routing Encapsulation (GRE) tunnel OAM.

Loss statistics: A set of one-way loss statistics attributes that are used to measure end-to-end loss between VPN sites or between any two network nodes. The exact loss value or the loss percentage can be reported.

Delay statistics: A set of one-way delay statistics attributes that are used to measure end-to-end latency between VPN sites or between any two network nodes. The peak/min values or percentile values can be reported.

Jitter statistics: A set of one-way IP Packet Delay Variation [RFC3393] statistics attributes that are used to measure end-to-end jitter between VPN sites or between any two network nodes. The peak/min values or percentile values can be reported.

PM statistics per class: "one-way-pm-statistics-per-class" lists performance measurement statistics for the topology link or the abstract link between VPN PEs with given "class-id" names. The list is defined separately from "one-way-pm-statistics", which is used to collect generic metrics for unspecified "class-id" names.

VPN PM type ("vpn-pm-type"): Indicates the VPN performance type, which can be "inter-vpn-access-interface" PM or "vpn-tunnel" PM. These two methods are common VPN measurement methods. The "inter-VPN-access-interface" PM is used to monitor the performance of logical point-to-point VPN connections between source and destination VPN access interfaces. And the "vpn-tunnel" PM is used to monitor the performance of VPN tunnels. The "inter-VPN-access-interface" PM includes PE-PE monitoring. Therefore, usually only one of the two methods is used. The "inter-VPN-access-interface" PM is defined as an empty leaf, which is not bound to a specific VPN access interface. The source or destination VPN access interface of the measurement can be augmented as needed.

VPN tunnel type ("vpn-tunnel-type"): Indicates the abstract link protocol-type of a VPN, such as GRE or IP-in-IP. The leaf refers to an identifier of the "underlay-transport" defined in [RFC9181], which describes the transport technology that carries the traffic of the VPN service. In the case of multiple types of tunnels between a single pair of VPN nodes, a separate link for each type of tunnel can be created.

For the data nodes of "termination-point" depicted in Figure 6, the module defines the following minimal set of statistics:

Last updated time ("last-updated"): Indicates the date and time when the counters were last updated.

Inbound statistics: A set of inbound statistics attributes that are used to measure the inbound statistics of the termination point, such as received packets, received packets with errors, etc.

Outbound statistics: A set of outbound statistics attributes that are used to measure the outbound statistics of the termination point, such as sent packets, packets that could not be sent due to errors, etc.

VPN network access ("vpn-network-access"): Lists counters of the VPN network access defined in the L3NM [RFC9182] or the L2NM [RFC9291]. When multiple VPN network accesses are created using the same physical port, finer-grained metrics can be monitored. If a TP is associated with only a single VPN, this list is not required.

5. Network and VPN Service Performance Monitoring YANG Module

The "ietf-network-vpn-pm" YANG module uses types defined in [RFC6991], [RFC8345], [RFC8532], and [RFC9181].

```
<CODE BEGINS> file "ietf-network-vpn-pm@2023-03-20.yang"

module ietf-network-vpn-pm {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm";
  prefix nvp;

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-vpn-common {
    prefix vpn-common;
    reference
      "RFC 9181: A Common YANG Data Model for Layer 2 and
           Layer 3 VPNs";
  }
  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network
           Topologies, Section 6.1";
  }
  import ietf-network-topology {
    prefix nt;
    reference
      "RFC 8345: A YANG Data Model for Network
           Topologies, Section 6.2";
  }
  import ietf-lime-time-types {
    prefix lime;
    reference
      "RFC 8532: Generic YANG Data Model for the Management of
           Operations, Administration, and Maintenance (OAM)
```

```
           Protocols That Use Connectionless Communications";
  }

  organization
    "IETF OPSAWG (Operations and Management Area Working Group)";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/opsawg/>
     WG List:  <mailto:opsawg@ietf.org>

     Editor: Bo Wu
          <lana.wubo@huawei.com>

     Editor: Mohamed Boucadair
          <mohamed.boucadair@orange.com>

     Editor: Qin Wu
          <bill.wu@huawei.com>

     Author: Oscar Gonzalez de Dios
          <oscar.gonzalezdedios@telefonica.com>

     Author: Bin Wen
          <bin_wen@comcast.com>";
  description
    "This YANG module defines a model for network and VPN service
     performance monitoring (PM).

     Copyright (c) 2023 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 Revised 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 9375
     (https://www.rfc-editor.org/info/rfc9375); see the RFC itself
     for full legal notices.";

  revision 2023-03-20 {
    description
      "Initial revision.";
    reference
      "RFC 9375: A YANG Data Model for Network and VPN Service
           Performance Monitoring";
  }

  identity node-type {
    description
      "Base identity for node type";
  }

  identity pe {
    base node-type;
    description
      "Provider Edge (PE) node type.  A PE is the device or set
```

```
         of devices at the edge of the provider network with the
         functionality that is needed to interface with the
         customer.";
    }

    identity p {
      base node-type;
      description
        "Provider router node type.  That is, a router
         in the core network that does not have interfaces
         directly toward a customer.";
    }

    identity asbr {
      base node-type;
      description
        "Autonomous System Border Router (ASBR) node type.";
      reference
        "RFC 4364: BGP/MPLS IP Virtual Private Networks (VPNs)";
    }

    identity pm-source-type {
      description
        "Base identity from which specific performance monitoring
         mechanism types are derived.";
    }

    identity pm-source-bgpls {
      base pm-source-type;
      description
        "Indicates BGP-LS as the performance monitoring metric
         source.";
      reference
        "RFC 8571: BGP - Link State (BGP-LS) Advertisement of
             IGP Traffic Engineering Performance Metric
             Extensions";
    }

    identity pm-source-owamp {
      base pm-source-type;
      description
        "Indicates the One-Way Active Measurement Protocol (OWAMP)
         as the performance monitoring metric source.";
      reference
        "RFC 4656: A One-way Active Measurement Protocol (OWAMP)";
    }

    identity pm-source-twamp {
      base pm-source-type;
      description
        "Indicates the Two-Way Active Measurement Protocol (TWAMP)
         as the performance monitoring metric source.";
      reference
        "RFC 5357: A Two-Way Active Measurement Protocol (TWAMP)";
    }

    identity pm-source-stamp {
      base pm-source-type;
```

```
    description
      "Indicates the Simple Two-way Active Measurement Protocol
       (STAMP) as the performance monitoring metric source.";
    reference
      "RFC 8762: Simple Two-Way Active Measurement Protocol";
  }

  identity pm-source-y-1731 {
    base pm-source-type;
    description
      "Indicates Ethernet OAM Y.1731 as the performance monitoring
       metric source.";
    reference
      "ITU-T Y.1731: Operations, administration and
             maintenance (OAM) functions and mechanisms
             for Ethernet-based networks";
  }

  identity pm-source-ioam {
    base pm-source-type;
    description
      "Indicates In Situ Operations, Administration, and Maintenance
       (IOAM) as the performance monitoring metric source.";
    reference
      "RFC 9197: Data Fields for In Situ Operations, Administration,
           and Maintenance (IOAM)";
  }

  identity pm-type {
    description
      "Base identity for the PM type.";
  }

  identity pm-type-network-link {
    base pm-type;
    description
      "Indicates that the PM type is for the link in
       the network topology.";
  }

  identity pm-type-vpn-inter-access {
    base pm-type;
    description
      "Indicates that the PM type is for logical point-to-point VPN
       connections between source and destination VPN access
       interfaces.";
  }

  identity pm-type-vpn-tunnel {
    base pm-type;
    description
      "Indicates that the PM type is for VPN tunnels.";
  }

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

```
      }
      description
        "Percentage to 5 decimal places.";
    }

    typedef percentile {
      type decimal64 {
        fraction-digits 3;
        range "0..100";
      }
      description
        "The percentile is a value between 0 and 100 to 3
         decimal places, e.g., 10.000, 99.900, and 99.990.
         For example, for a given one-way delay measurement,
         if the percentile is set to 95.000 and the 95th percentile
         one-way delay is 2 milliseconds, then the 95 percent of
         the sample value is less than or equal to 2 milliseconds.";
    }

    grouping entry-summary {
      description
        "Entry summary grouping used for network topology
         augmentation.";
      container entry-summary {
        config false;
        description
          "Container for VPN or network entry summary.";
        container ipv4-num {
          leaf maximum-routes {
            type uint32;
            description
              "Indicates the maximum number of IPv4 routes
               for the VPN or network.";
          }
          leaf total-active-routes {
            type uint32;
            description
              "Indicates total active IPv4 routes
               for the VPN or network.";
          }
          description
            "IPv4-specific parameters.";
        }
        container ipv6-num {
          leaf maximum-routes {
            type uint32;
            description
              "Indicates the maximum number of IPv6 routes
               for the VPN or network.";
          }
          leaf total-active-routes {
            type uint32;
            description
              "Indicates total active IPv6 routes
               for the VPN or network.";
          }
          description
            "IPv6-specific parameters.";
```

```
      }
      container mac-num {
        leaf maximum-mac-entries {
          type uint32;
          description
            "Indicates the maximum number of MAC entries
             for the VPN or network.";
        }
        leaf total-active-mac-entries {
          type uint32;
          description
            "Indicates the total active MAC entries
             for the VPN or network.";
        }
        description
          "MAC statistics.";
      }
    }
  }

  grouping link-loss-statistics {
    description
      "Grouping for per-link error statistics.";
    container loss-statistics {
      description
        "One-way link loss summarized information.";
      reference
        "RFC 4656: A One-way Active Measurement Protocol (OWAMP)
         ITU-T Y.1731: Operations, administration and
               maintenance (OAM) functions and mechanisms
               for Ethernet-based networks";
      leaf packet-loss-count {
        type yang:counter64;
        description
          "Total number of lost packets.";
      }
      leaf loss-ratio {
        type percentage;
        description
          "Loss ratio of the packets.  Expressed as percentage
           of packets lost with respect to packets sent.";
      }
    }
  }

  grouping link-delay-statistics {
    description
      "Grouping for per-link delay statistics.";
    container delay-statistics {
      description
        "One-way link delay summarized information.";
      reference
        "RFC 4656: A One-way Active Measurement Protocol (OWAMP)
         ITU-T Y.1731: Operations, administration and
               maintenance (OAM) functions and mechanisms
               for Ethernet-based networks";
      leaf unit-value {
        type identityref {
```

```
            base lime:time-unit-type;
          }
          default "lime:milliseconds";
          description
            "Time units, where the options are hours, minutes, seconds,
             milliseconds, microseconds, and nanoseconds.";
        }
        leaf min-delay-value {
          type yang:gauge64;
          description
            "Minimum observed one-way delay.";
        }
        leaf max-delay-value {
          type yang:gauge64;
          description
            "Maximum observed one-way delay.";
        }
        leaf low-delay-percentile {
          type yang:gauge64;
          description
            "Low percentile of observed one-way delay with
             specific measurement method.";
        }
        leaf intermediate-delay-percentile {
          type yang:gauge64;
          description
            "Intermediate percentile of observed one-way delay with
             specific measurement method.";
        }
        leaf high-delay-percentile {
          type yang:gauge64;
          description
            "High percentile of observed one-way delay with
             specific measurement method.";
        }
      }
    }

    grouping link-jitter-statistics {
      description
        "Grouping for per-link jitter statistics.";
      container jitter-statistics {
        description
          "One-way link jitter summarized information.";
        reference
          "RFC 3393: IP Packet Delay Variation Metric
               for IP Performance Metrics (IPPM)
           RFC 4656: A One-way Active Measurement Protocol (OWAMP)
           ITU-T Y.1731: Operations, administration and
                 maintenance (OAM) functions and mechanisms
                 for Ethernet-based networks";
        leaf unit-value {
          type identityref {
            base lime:time-unit-type;
          }
          default "lime:milliseconds";
          description
            "Time units, where the options are hours, minutes, seconds,
```

```
             milliseconds, microseconds, and nanoseconds.";
        }
        leaf min-jitter-value {
          type yang:gauge64;
          description
            "Minimum observed one-way jitter.";
        }
        leaf max-jitter-value {
          type yang:gauge64;
          description
            "Maximum observed one-way jitter.";
        }
        leaf low-jitter-percentile {
          type yang:gauge64;
          description
            "Low percentile of observed one-way jitter.";
        }
        leaf intermediate-jitter-percentile {
          type yang:gauge64;
          description
            "Intermediate percentile of observed one-way jitter.";
        }
        leaf high-jitter-percentile {
          type yang:gauge64;
          description
            "High percentile of observed one-way jitter.";
        }
      }
    }

    grouping tp-svc-telemetry {
      leaf last-updated {
        type yang:date-and-time;
        config false;
        description
          "Indicates the date and time when the counters were
           last updated.";
      }
      leaf inbound-octets {
        type yang:counter64;
        description
          "The total number of octets received on the
           interface, including framing characters.";
      }
      leaf inbound-unicast {
        type yang:counter64;
        description
          "The total number of inbound unicast packets.";
      }
      leaf inbound-broadcast {
        type yang:counter64;
        description
          "The total number of inbound broadcast packets.";
      }
      leaf inbound-multicast {
        type yang:counter64;
        description
          "The total number of inbound multicast packets.";
```

```
      }
      leaf inbound-discards {
        type yang:counter64;
        description
          "The number of inbound packets that were discarded
           even though no errors had been detected.  Possible
           reasons for discarding such a packet could be to
           free up buffer space, not enough buffer for too
           much data, etc.";
      }
      leaf inbound-errors {
        type yang:counter64;
        description
          "The number of inbound packets that contained errors.";
      }
      leaf inbound-unknown-protocol {
        type yang:counter64;
        description
          "The number of packets received via the interface
           that were discarded because of an unknown or
           unsupported protocol.";
      }
      leaf outbound-octets {
        type yang:counter64;
        description
          "The total number of octets transmitted out of the
           interface, including framing characters.";
      }
      leaf outbound-unicast {
        type yang:counter64;
        description
          "The total number of outbound unicast packets.";
      }
      leaf outbound-broadcast {
        type yang:counter64;
        description
          "The total number of outbound broadcast packets.";
      }
      leaf outbound-multicast {
        type yang:counter64;
        description
          "The total number of outbound multicast packets.";
      }
      leaf outbound-discards {
        type yang:counter64;
        description
          "The number of outbound packets that were discarded
           even though no errors had been detected to
           prevent their transmission.  Possible reasons
           for discarding such a packet could be to free
           up buffer space, not enough buffer for too
           much data, etc.";
      }
      leaf outbound-errors {
        type yang:counter64;
        description
          "The number of outbound packets that contained errors.";
      }
```

```
    description
      "Grouping for interface service telemetry.";
  }

  augment "/nw:networks/nw:network/nw:network-types" {
    description
      "Defines the service topologies types.";
    container service {
      presence "Presence of the container indicates performance
                monitoring of the VPN service, and absence of
                the container indicates performance monitoring
                of the network itself.";
      description
        "Container for VPN service.";
      leaf service-type {
        type identityref {
          base vpn-common:service-type;
        }
        mandatory true;
        description
          "This indicates the network service type,
           e.g., L3VPN and VPLS.";
      }
      leaf vpn-id {
        type vpn-common:vpn-id;
        description
          "VPN identifier.";
      }
      leaf vpn-service-topology {
        type identityref {
          base vpn-common:vpn-topology;
        }
        description
          "VPN service topology, e.g., hub-spoke, any-to-any,
           and hub-spoke-disjoint.";
      }
    }
  }

  augment "/nw:networks/nw:network/nw:node" {
    description
      "Augments the network node with other general attributes.";
    leaf node-type {
      type identityref {
        base node-type;
      }
      description
        "Node type, e.g., PE, P, and ASBR.";
    }
    uses entry-summary;
  }

  augment "/nw:networks/nw:network/nw:node" {
    when '../nw:network-types/nvp:service' {
      description
        "Augments for VPN service PM.";
    }
    description
```

```
        "Augments the network node with VPN service attributes.";
      leaf role {
        type identityref {
          base vpn-common:role;
        }
        default "vpn-common:any-to-any-role";
        description
          "Role of the node in the VPN service topology.";
      }
    }

    augment "/nw:networks/nw:network/nt:link" {
      description
        "Augments the network topology link with performance
         monitoring attributes.";
      container perf-mon {
        description
          "Container for PM attributes.";
        leaf low-percentile {
          type percentile;
          default "10.000";
          description
            "Low percentile to report.  Setting low-percentile
             to 0.000 indicates the client is not interested
             in receiving low percentile.";
        }
        leaf intermediate-percentile {
          type percentile;
          default "50.000";
          description
            "Intermediate percentile to report.  Setting
             intermediate-percentile to 0.000 indicates the client
             is not interested in receiving intermediate percentile.";
        }
        leaf high-percentile {
          type percentile;
          default "95.000";
          description
            "High percentile to report.  Setting high-percentile
             to 0.000 indicates the client is not interested in
             receiving high percentile.";
        }
        leaf measurement-interval {
          type uint32 {
            range "1..max";
          }
          units "seconds";
          default "60";
          description
            "Indicates the time interval to perform PM
             measurement over.";
        }
        list pm {
          key "pm-type";
          config false;
          description
            "The list of PM based on PM type.";
          leaf pm-type {
```

```
            type identityref {
              base pm-type;
            }
            config false;
            description
              "The PM type of the measured PM attributes.";
          }
          container pm-attributes {
            description
              "Container for PM attributes.";
            leaf start-time {
              type yang:date-and-time;
              config false;
              description
                "The date and time the measurement last started.";
            }
            leaf end-time {
              type yang:date-and-time;
              config false;
              description
                "The date and time the measurement last ended.";
            }
            leaf pm-source {
              type identityref {
                base pm-source-type;
              }
              config false;
              description
                "The OAM tool used to collect the PM data.";
            }
            container one-way-pm-statistics {
              config false;
              description
                "Container for link telemetry attributes.";
              uses link-loss-statistics;
              uses link-delay-statistics;
              uses link-jitter-statistics;
            }
            list one-way-pm-statistics-per-class {
              key "class-id";
              config false;
              description
                "The list of PM data based on class of service.";
              leaf class-id {
                type string;
                description
                  "The class-id is used to identify the class
                   of service.  This identifier is internal
                   to the administration.";
              }
              uses link-loss-statistics;
              uses link-delay-statistics;
              uses link-jitter-statistics;
            }
          }
        }
      }
    }
```

```
  augment "/nw:networks/nw:network/nt:link/perf-mon" {
    when '../../nw:network-types/nvp:service' {
      description
        "Augments for VPN service PM.";
    }
    description
      "Augments the network topology link with VPN service
       performance monitoring attributes.";
    container vpn-pm-type {
      description
        "The VPN PM type of this logical point-to-point
         unidirectional VPN link.";
      container inter-vpn-access-interface {
        description
          "Indicates inter-vpn-access-interface PM, which is used
           to monitor the performance of logical point-to-point
           VPN connections between source and destination VPN
           access interfaces.";
        leaf inter-vpn-access-interface {
          type empty;
          description
            "This is a placeholder for inter-vpn-access-interface PM,
             which is not bound to a specific VPN access interface.
             The source or destination VPN access interface
             of the measurement can be augmented as needed.";
        }
      }
      container vpn-tunnel {
        presence "Enables VPN tunnel PM";
        description
          "Indicates VPN tunnel PM, which is used to monitor
           the performance of VPN tunnels.";
        leaf vpn-tunnel-type {
          type identityref {
            base vpn-common:protocol-type;
          }
          config false;
          description
            "The leaf indicates the VPN tunnel type, e.g.,
             Generic Routing Encapsulation (GRE) and Generic
             Network Virtualization Encapsulation (Geneve).";
        }
      }
    }
  }

  augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
    description
      "Augments the network topology termination point with
       performance monitoring attributes.";
    container pm-statistics {
      config false;
      description
        "Container for termination point PM attributes.";
      uses tp-svc-telemetry;
    }
  }
```

```
  augment "/nw:networks/nw:network/nw:node"
        + "/nt:termination-point/pm-statistics" {
    when '../../../nw:network-types/nvp:service' {
      description
        "Augments for VPN service PM.";
    }
    description
      "Augments the network topology termination-point with
       VPN service performance monitoring attributes.";
    list vpn-network-access {
      key "network-access-id";
      description
        "The list of PM based on VPN network accesses.";
      leaf network-access-id {
        type vpn-common:vpn-id;
        description
          "The reference to an identifier for the VPN network
           access.";
      }
      uses tp-svc-telemetry;
    }
  }
}

<CODE ENDS>
```

6. Security Considerations

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

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

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 write operations can lead to inaccurate or incomplete network measurements that can impact the visibility and decisions this data would be used to inform. Unauthorized write access to the following subtrees could have the following impacts:

Access	Node	Potential Impact
/nw:networks/nw:network/nw:network-types		

Access	Node	Potential Impact
write	service type	disable VPN PM
write	VPN identifier	disable VPN PM
write	VPN service topology	render data unusable
/nw:networks/nw:network/nw:node		
write	node type	render data unusable
write	VPN topology role	render data unusable
/nw:networks/nw:network/nw:link/nvp:perf-mon		
write	percentile	impact reporting cadence
write	measurement interval	impact monitoring fidelity
write	vpn-pm-type	impact monitoring fidelity

Table 1: Write Operation Sensitivity Impact

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. When using, the trade-off between confidentiality and proper monitoring of performance needs to be considered. Unauthorized access to the following subtrees could have the following impacts:

"/nw:networks/nw:network/nw:node": Unauthorized read access to this subtree can disclose the operational state information of underlay network instances or VPN instances.

"/nw:networks/nw:network/nt:link/nvp:perf-mon/nvp:one-way-pm-statistics": Unauthorized read access to this subtree can disclose the operational state information of underlay network links or VPN abstract links.

"/nw:networks/nw:network/nw:node/nt:termination-point/nvp:pm-statistics": Unauthorized read access to this subtree can disclose the operational state information of underlay network termination points or VPN network accesses.

This YANG module does not define any Remote Procedure Call (RPC) operations and actions.

7. IANA Considerations

IANA has registered the following URI in the "ns" subregistry within the "IETF XML Registry" [RFC3688]:

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

IANA has registered the following YANG module in the "YANG Module Names" subregistry [RFC6020] within the "YANG Parameters" registry.

Name: ietf-network-vpn-pm
Namespace: urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm
Maintained by IANA: N
Prefix: nvp
Reference: RFC 9375

8. References

8.1. Normative References

[RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <https://www.rfc-editor.org/info/rfc3393>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[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>.

[RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP)", RFC 4656, DOI 10.17487/RFC4656, September 2006, <https://www.rfc-editor.org/info/rfc4656>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <https://www.rfc-editor.org/info/rfc5357>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <https://www.rfc-editor.org/info/rfc6374>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <https://www.rfc-editor.org/info/rfc8345>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[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>.

[RFC8571] Ginsberg, L., Ed., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, DOI 10.17487/RFC8571, March 2019, <https://www.rfc-editor.org/info/rfc8571>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

[RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <https://www.rfc-editor.org/info/rfc8762>.

[RFC9181] Barguil, S., Gonzalez de Dios, O., Ed., Boucadair, M., Ed., and Q. Wu, "A Common YANG Data Model for Layer 2 and Layer 3 VPNs", RFC 9181, DOI 10.17487/RFC9181, February 2022, <https://www.rfc-editor.org/info/rfc9181>.

8.2. Informative References

[ITU-T-Y-1731] ITU-T, "Operations, administration and maintenance (OAM) functions and mechanisms for Ethernet-based networks", ITU-T Recommendation G.8013/Y.1731, August 2015, <https://www.itu.int/rec/T-REC-Y.1731/en>.

[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>.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <https://www.rfc-editor.org/info/rfc5277>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[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>.

[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>.

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

[RFC8632] Vallin, S. and M. Bjorklund, "A YANG Data Model for Alarm Management", RFC 8632, DOI 10.17487/RFC8632, September 2019, <https://www.rfc-editor.org/info/rfc8632>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

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

[RFC9182] Barguil, S., Gonzalez de Dios, O., Ed., Boucadair, M., Ed., Munoz, L., and A. Aguado, "A YANG Network Data Model for Layer 3 VPNs", RFC 9182, DOI 10.17487/RFC9182, February 2022, <https://www.rfc-editor.org/info/rfc9182>.

[RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197, May 2022, <https://www.rfc-editor.org/info/rfc9197>.

[RFC9291] Boucadair, M., Ed., Gonzalez de Dios, O., Ed., Barguil, S., and L. Munoz, "A YANG Network Data Model for Layer 2 VPNs", RFC 9291, DOI 10.17487/RFC9291, September 2022, <https://www.rfc-editor.org/info/rfc9291>.

[YANG-SAP] Boucadair, M., Ed., Gonzalez de Dios, O., Barguil, S., Wu, Q., and V. Lopez, "A YANG Network Model for Service Attachment Points (SAPs)", Work in Progress, Internet-Draft, draft-ietf-opsawg-sap-15, 18 January 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-sap-15>.

Appendix A. Illustrative Examples

A.1. Example of VPN Performance Subscription

The example shown in Figure 7 illustrates how a client subscribes to the performance monitoring information between nodes ("node-id") A and B in the L3 network topology. The performance monitoring parameter that the client is interested in is end-to-end loss.

```
============== NOTE: '\' line wrapping per RFC 8792 ===============

POST /restconf/operations/ietf-subscribed-notifications:establish-\
                                   subscription
Host: example.com
Content-Type: application/yang-data+json

{
  "ietf-subscribed-notifications:input": {
    "stream-subtree-filter": {
      "ietf-network:networks": {
        "network": {
          "network-id": "example:VPN1",
          "ietf-network-vpn-pm:service": {
            "service-type": "ietf-vpn-common:l3vpn"
          },
          "node": [
            {
              "node-id": "example:A",
              "ietf-network-vpn-pm:node-type": "pe",
              "termination-point": [
                {
                  "tp-id": "example:1-0-1"
                }
              ]
            },
            {
              "node-id": "example:B",
              "ietf-network-vpn-pm:node-type": "pe",
              "termination-point": [
                {
                  "tp-id": "example:2-0-1"
                }
              ]
            }
          ],
          "ietf-network-topology:link": [
            {
              "link-id": "example:A-B",
              "source": {
                "source-node": "example:A"
              },
              "destination": {
                "dest-node": "example:B"
              },
              "ietf-network-vpn-pm:perf-mon": {
                "pm": [
                  {
                    "pm-type": "pm-type-vpn-tunnel",
                    "pm-attributes": {
                      "one-way-pm-statistics": {
                        "loss-statistics": {
                          "packet-loss-count": {}
                        }
                      }
                    }
                  }
```

```
                  ],
                  "vpn-pm-type": {
                    "vpn-tunnel": {
                      "vpn-tunnel-type": "ietf-vpn-common:gre"
                    }
                  }
                }
              }
            ]
          }
        },
        "ietf-yang-push:periodic": {
          "period": "500"
        }
      }
    }
}
```


Figure 7: Example of Pub/Sub Retrieval

A.2. Example of VPN Performance Snapshot

The example depicted in Figure 8 illustrates a VPN PM instance message body of a RESTCONF request to fetch the performance data of the link and TP that belongs to "VPN1".

```
{
  "ietf-network:networks": {
    "network": {
      "network-id": "example:VPN1",
      "node": [
        {
          "node-id": "example:A",
          "ietf-network-vpn-pm:node-type": "pe",
          "termination-point": [
            {
              "tp-id": "example:1-0-1",
              "ietf-network-vpn-pm:pm-statistics": {
                "inbound-octets": "100",
                "outbound-octets": "150"
              }
            }
          ]
        },
        {
          "node-id": "example:B",
          "ietf-network-vpn-pm:node-type": "pe",
          "termination-point": [
            {
              "tp-id": "example:2-0-1",
              "ietf-network-vpn-pm:pm-statistics": {
                "inbound-octets": "150",
                "outbound-octets": "100"
              }
            }
          ]
        }
      ],
      "ietf-network-topology:link": [
        {
          "link-id": "example:A-B",
          "source": {
            "source-node": "example:A"
          },
          "destination": {
            "dest-node": "example:B"
          },
          "ietf-network-pm:perf-mon": {
            "pm": [
              {
                "pm-type": "pm-type-vpn-tunnel",
                "pm-attributes": {
                  "one-way-pm-statistics": {
                    "loss-statistics": {
                      "packet-loss-count": "120"
                    }
                  }
                }
              }
            ],
            "vpn-pm-type": {
              "vpn-tunnel": {
                "vpn-tunnel-type": "ietf-vpn-common:gre"
```

```
              }
            }
          }
        }
      ]
    }
  }
}
```


Figure 8: Example of VPN PM

A.3. Example of Percentile Monitoring

This is an example of percentile measurement data that could be returned for link "example:A-B" between "example:A" and "example:B".

```
{
  "ietf-network-topology:link": [
    {
      "link-id": "example:A-B",
      "source": {
        "source-node": "example:A"
      },
      "destination": {
        "dest-node": "example:B"
      },
      "ietf-network-vpn-pm:perf-mon": {
        "low-percentile": "20.000",
        "intermediate-percentile": "50.000",
        "high-percentile": "90.000",
        "pm": [
          {
            "pm-type": "pm-type-vpn-inter-access",
            "pm-attributes": {
              "one-way-pm-statistics": {
                "delay-statistics": {
                  "unit-value": "ietf-lime-time-types:milliseconds",
                  "min-delay-value": "43",
                  "max-delay-value": "99",
                  "low-delay-percentile": "64",
                  "intermediate-delay-percentile": "77",
                  "high-delay-percentile": "98"
                }
              }
            }
          }
        ],
        "vpn-pm-type": {
          "inter-vpn-access-interface": {
            "inter-vpn-access-interface": [null]
          }
        }
      }
    }
  ]
}
```


Figure 9: Example of VPN PM with Percentile Value

Acknowledgements

Thanks to Joe Clarke, Adrian Farrel, Tom Petch, Greg Mirsky, Roque Gagliano, Erez Segev, and Dhruv Dhody for reviewing and providing important input to this document.

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

Contributors

The following authors contributed significantly to this document:

Michale Wang
Huawei
Email: wangzitao@huawei.com

Roni Even
Huawei
Email: ron.even.tlv@gmail.com

Change Liu
China Unicom
Email: liuc131@chinaunicom.cn

Honglei Xu
China Telecom
Email: xuhl6@chinatelecom.cn

Authors' Addresses

Bo Wu (EDITOR)
Huawei
Yuhua District
101 Software Avenue
Nanjing
Jiangsu, 210012
China
Email: lana.wubo@huawei.com

Qin Wu (EDITOR)
Huawei
Yuhua District
101 Software Avenue
Nanjing
Jiangsu, 210012
China
Email: bill.wu@huawei.com

Mohamed Boucadair (EDITOR)
Orange
Rennes 35000
France
Email: mohamed.boucadair@orange.com

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

Bin Wen
Comcast
Email: bin_wen@comcast.com