Workgroup: OPSAWG Working Group
Internet-Draft:
draft-ietf-opsawg-yang-vpn-service-pm-04
Published: 21 March 2022
Intended Status: Standards Track
Expires: 22 September 2022
Authors: B. Wu, Ed. Q. Wu, Ed. M. Boucadair, Ed.
Huawei Huawei Orange
O. Gonzalez de Dios B. Wen
Telefonica Comcast

A YANG 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 networks and VPN services that can be used to monitor and manage network performance on the topology at higher layer or the service topology between VPN sites.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction
2. Terminology
   2.1. Acronyms
3. Network and VPN Service Performance Monitoring Model Usage
   3.1. Collecting Data via Pub/Sub Mechanism
   3.2. Collecting Data On-demand
4. Description of The Data Model
   4.1. Layering Relationship between Multiple Layers of Topology
   4.2. Network Level
   4.3. Node Level
   4.4. Link and Termination Point Level
5. Network and VPN Service Performance Monitoring YANG Module
6. Security Considerations
7. IANA Considerations
8. Acknowledgements
9. Contributors
10. References
   10.1. Normative References
   10.2. Informative References
Appendix A. Illustrating Examples
   A.1. VPN Performance Subscription Example
   A.2. Example of VPN Performance Snapshot
   A.3. Example of Percentile Monitoring
Authors' Addresses

1. Introduction

[RFC8969] describes a framework for automating service and network management with YANG models. It defines that the performance measurement telemetry model to be tied with the service, such as Layer 3 VPN and Layer 2 VPN, or network models to monitor the overall network performance or Service Level Agreement (SLA).

The performance of VPN services is associated with the performance changes of the underlay network that carries VPN services, such as the delay of the underlay tunnels and the packet loss status of the device interfaces. Additionally, the integration of Layer 2/Layer 3 VPN performance and network performance data enables the orchestrator to subscribe to VPN service performance in a unified manner. 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, in particular.

This document does not introduce new metrics for network performance or mechanisms for measuring network performance, but uses the existing mechanisms and statistics to display the performance monitoring statistics at the network and service layers. All these metrics are defined as unidirectional metrics.

The YANG module defined in this document is designed as an augmentation to the network topology YANG 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:

L2VPN Layer 2 Virtual Private Network
L3VPN Layer 3 Virtual Private Network
L2NM L2VPN Network Model
L3NM L3VPN Network Model
MPLS Multiprotocol Label Switching
OAM Operations, Administration, and Maintenance
OWAMP One-Way Active Measurement Protocol
PE Provider Edge
PM Performance Monitoring
TWAMP Two-Way Active Measurement Protocol
VPLS Virtual Private LAN Service
VPN Virtual Private Network

3. Network and VPN Service Performance Monitoring Model Usage

Models are key for automating network management operations. According to [RFC8969], 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      |
                        |  Orchestration  |
                        +------+-+--------+
                               | |
        Network Service Models | | Network and VPN Service PM Models
                               | |
                        +------+-+--------+
                        |     Network     |
                        |   Controller    |
                        +-------+---------+
                                |
        +-----------------------+-----------------------+
                              Network
```

Figure 1: Reference Architecture

As shown in Figure 1, in the context of the layering model architecture described in [RFC8309], the network and VPN service performance monitoring (PM) model can be used to expose a set of performance information to the above layer. Such information can be used by an orchestrator to subscribe to performance data. The network controller will then notify the orchestrator about corresponding parameter changes.

Before using the model, the controller needs to establish complete topology visibility of the network and VPN. For example, the controller can use information from [RFC8345], [I-D.ietf-opsawg-sap] or VPN instances. Then the controller derive network or VPN level performance data by aggregating (and filtering) lower-level data collected via monitoring counters of the involved devices.

The network or VPN performance data can be based on different sources. For example, the performance monitoring data per link in the underlying network can be collected using a network performance measurement method such as One-Way Active Measurement Protocol (OWAMP) [RFC4656], Two-Way Active Measurement Protocol (TWAMP) [RFC5357], and Multiprotocol Label Switching (MPLS) Loss and Delay Measurement [RFC6374]. The performance monitoring information reflecting the quality of the network or VPN service (e.g., end-to-end network performance data between source node and destination node in the network or between VPN sites) can be computed and aggregated, for example, using the information from the Traffic Engineering Database (TED), [RFC7471] [RFC8570] [RFC8571] or 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 a network-wide measurement interval to align measurement requirements for networks or VPN services.

In addition, the amount of performance data collected from the devices can be huge. To avoid receiving a large amount of operational data of VPN instances, VPN interfaces, or tunnels, the network controller can specifically subscribe to metric-specific data using the tagging methods defined in [I-D.ietf-netmod-node-tags].

3.1. Collecting Data via 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.

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

3.2. Collecting Data On-demand

To obtain a snapshot of a large amount of performance data from a network topology or VPN network, service-assurance applications may retrieve information using the network and VPN service PM model through a 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 Data Model

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

The performance monitoring data augments the service topology as shown in Figure 2.

```
+----------------------+          +----------------------+
|ietf-network          |          |Network and VPN Service|
|ietf-network-topology |<---------|Performance Monitoring |
+----------------------+ augments |        Model          |
                                  +----------------------+
```

Figure 2: Module Augmentation

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 virtual topology for a service layer above Layer 1 (L1), Layer 2 (L2), and Layer 3 (L3). This service topology has the generic topology elements of node, link, and terminating 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 common L3 topology.

Figure 3 illustrates an example of a topology that maps between the VPN service topology and an underlying network:

```
                     VPN 1                       VPN 2
           +-----------------------+    +----------------------+
          /                       /    /                     /
         /S1C_[VN3]:::           /    /S2A             S2B  /
        /         \   :::::     /    / _[VN1]______[VN3]_  /
       /           \       :   /    /   :            :    / Overlay
      /             \       :: : : : : :             :   /
     /S1B_[VN2]____[VN1]_S1A /    /   : :            :  /
    +--------:-------:------+    +---:----:----------:-+
             :        :       :: : :       :         :
             :         :    :              :         :
   Site-1A   :  +-------:--: ----- -------- : -------:-----+ Site-1C
     [CE1]___: /__ ___ [N1]__________________ [N2]__ :___ /__[CE3]
             :/      /  / \             _____/ /    :    /
   [CE5]___  : ___  /  /    \     _____/      /   ::    /
 Site-2A    /:        /       \  /           /   :     /
           / :                [N5]          /   :     / Underlay Network
          /   :     /       __/ \__        /   :     /
         /     :   /    ___/       \__    /   :     /
Site-1B /       : / ___/              \  /   :     /  Site-2B
[CE2]_ /________[N4]_________________ [N3]:::_____/____[CE4]
      +------------------------------------------+

    Legend:
    N:node   VN:VPN-Node  S:Site
    __  Link
    :   Mapping between networks
```

Figure 3: Example of Topology Mapping Between VPN Service Topology and Underlying Network

As shown in Figure 3, two VPN services topologies are both built on top of one common underlying physical network:

VPN 1: This service topology supports hub-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 physical network. 'Site-1A' plays the role of hub while 'Site-1B' and 'Site-1C' are configured as spoke.

VPN 2: This service 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 nodes 1 (N1) and node 3 (N3)5 in the underlying physical network. 'Site-2A' and 'Site-2B' have 'any-to-any' role.

Apart from the association between the VPN topology and the underlay topology, VPN Network PM can also provide the performance status of the underlay network and VPN services. For example, network PM can provide link PM statistics and port statistics. VPN PM can provides statistics on VPN access interfaces, the number of current VRF routes or L2VPN MAC entry of VPN nodes, and performance statistics on the logical point-to-point link between source and destination VPN nodes or between source and destination VPN access interfaces. Figure 4 illustrates an example of VPN PM and the difference between two VPN PM measurement methods. One is the VPN tunnel PM and the other is inter-VPN-access interface PM.

```
       +------------------------------------------------------+
       |                                                      |
       |                      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
        -:Link
```

Figure 4: An Example of VPN PM

4.2. Network Level

For network performance monitoring, the container of "networks" in [RFC8345] does not need to be extended.

For VPN service performance monitoring, the container "service-type" is defined to indicate the VPN type, 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 type, it represents a VPN instance that can perform performance monitoring.

The tree in Figure 5 is a part of 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 the VPN topology. This model supports "any-to-any", "Hub and Spoke" (where Hubs can exchange traffic), and "Hub and Spoke disjoint" (where Hubs cannot exchange traffic) that are taken from [RFC9181]. These VPN 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-type!
       +--rw service-type?   identityref
  augment /nw:networks/nw:network:
    +--rw vpn-pm-attributes
       +--rw vpn-id?                 vpn-common:vpn-id
       +--rw vpn-service-topology?   identityref
```

Figure 5: Network Level YANG Tree of the Hierarchies

4.3. Node Level

The tree in Figure 6 is the node part of ietf-network-vpn-pm tree.

For network performance monitoring, a container of "pm-attributes" is augmented to the list of "node" that are defined in [RFC8345]. The container includes the following attributes:

"node-type": Indicates the device type of Provider Edge (PE), Provider (P) device, or Autonomous System Border Router (ASBR), so that the performance metric between any two nodes each with 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 performance monitoring, the model 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, hub-role).

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

Figure 6: Node Level YANG Tree of the Hierarchies

4.4. Link and Termination Point Level

The tree in Figure 7 is the link and termination point (TP) part of 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.

The performance data of a link is a collection of counters that report the performance status.

```
augment /nw:networks/nw:network/nt:link:
  +--rw pm-attributes
     +--rw low-percentile?                      percentile
     +--rw intermediate-percentile?             percentile
     +--rw high-percentile?                     percentile
     +--rw measurement-interval?                uint32
     +--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:gauge32
     |     +--ro max-jitter-value?                 yang:gauge32
     |     +--ro low-jitter-percentile?            yang:gauge32
     |     +--ro intermediate-jitter-percentile?   yang:gauge32
     |     +--ro high-jitter-percentile?           yang:gauge32
     +--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:gauge32
     |     +--ro max-jitter-value?                 yang:gauge32
     |     +--ro low-jitter-percentile?            yang:gauge32
     |     +--ro intermediate-jitter-percentile?   yang:gauge32
     |     +--ro high-jitter-percentile?           yang:gauge32
     +--rw (vpn-pm-type)?
        +--:(inter-vpn-access-interface)
        |  +--rw inter-vpn-access-interface?    empty
```

```
        +--:(underlay-tunnel)
           +--ro vpn-underlay-transport-type?   identityref
augment /nw:networks/nw:network/nw:node/nt:termination-point:
  +--ro pm-statistics
     +--ro reference-time?             yang:date-and-time
     +--ro inbound-octets?             yang:counter64
     +--ro inbound-unicast?            yang:counter64
     +--ro inbound-nunicast?           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-nunicast?          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 reference-time?             yang:date-and-time
        +--ro inbound-octets?             yang:counter64
        +--ro inbound-unicast?            yang:counter64
        +--ro inbound-nunicast?           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-nunicast?          yang:counter64
        +--ro outbound-discards?          yang:counter64
        +--ro outbound-errors?            yang:counter64
```

Figure 7: Link and Termination point Level YANG Tree of the hierarchies

For the data nodes of 'link' depicted in Figure 7, the YANG module defines the following minimal set of link-level performance attributes:

Percentile parameters: The module supports reporting delay and jitter metric by percentile values. By default, low percentile (10th percentile), intermediate percentile (50th percentile), high percentile (90th percentile) are used. Setting a percentile to 0.00 indicates the client is not interested in receiving particular percentile. If all percentile nodes are set to 0.00, this represents that no percentile related nodes will be reported for a given performance metric (e.g., one-way delay, 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.

PM source ("pm-source"): Indicates the performance monitoring source. The data for the topology link can be based, e.g., on 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 [I-D.ietf-opsawg-l2nm]. Alternatively, the data can be based upon the underlay technology OAM mechanisms, for example, Generic Routing Encapsulation (GRE) tunnel OAM.

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.

Reference time ("reference-time"): Indicates the timestamp when the counters are obtained.

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 underlay 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 underlay-tunnel PM. These two methods are common VPN measurement methods. The inter-VPN-access-interface PM is to monitor the performance of logical point-to-point connections between a source and a destination VPN access interfaces. And the underlay-tunnel PM is to monitor the performance of underlay tunnels of VPNs. The inter-VPN-access-interface PM includes PE-PE monitoring. Therefore, only one of the two methods is needed , and the model defines "choice" to indicate these two methods, which also allows other methods to be extended.

VPN underlay transport type ("vpn-underlay-transport-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 to carry the traffic of the VPN service.

For the data nodes of 'termination-point' depicted in Figure 7, the module defines the following minimal set of statistics:

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 [RFC9182] or [I-D.ietf-opsawg-l2nm]. 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" module uses types defined in [RFC8345], [RFC6991], [RFC8532], and [RFC9181].

```
<CODE BEGINS> file "ietf-network-vpn-pm@2021-03-21.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 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 module defines a model for Network and VPN Service
   Performance monitoring.

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

   Redistribution and use in source and binary forms, with or
   without modification, is permitted pursuant to, and subject
   to the license terms contained in, the 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 XXXX
   (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
   for full legal notices.";

// RFC Ed.: update the date below with the date of RFC
// publication and remove this note.
// RFC Ed.: replace XXXX with actual RFC number and remove
// this note.

revision 2022-03-21 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG 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.";
  reference
    "RFC 4026: Provider Provisioned
     Virtual Private Network (VPN) Terminology";
}
```

```
identity p {
  base node-type;
  description
    "Provider router node type.";
  reference
    "RFC 4026: Provider Provisioned
     Virtual Private Network (VPN) Terminology";
}

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 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 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-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";
}

typedef percentage {
  type decimal64 {
    fraction-digits 5;
    range "0..100";
  }
  description
    "Percentage.";
}

typedef percentile {
  type decimal64 {
    fraction-digits 2;
    range "0..100";
  }
  description
    "The percentile is a value between 0 and 100,
     e.g. 10.00, 99.90 ,99.99 etc..
     For example, for a given one-way delay measurement,
     if the percentile is set to 95.00 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 {
      leaf maximum-routes {
        type uint32;
        description
          "Indicates the maximum number of IPv4 routes
```

```
             for the VPN.";
        }
        leaf total-active-routes {
          type uint32;
          description
            "Indicates total active IPv4 routes for the VPN.";
        }
        description
          "IPv4-specific parameters.";
      }
      container ipv6 {
        leaf maximum-routes {
          type uint32;
          description
            "Indicates the maximum number of IPv6 routes
             for the VPN.";
        }
        leaf total-active-routes {
          type uint32;
          description
            "Indicates total active IPv6 routes for the VPN.";
        }
        description
          "IPv6-specific parameters.";
      }
      container mac-num {
        leaf mac-num-limit {
          type uint32;
          description
            "Maximum number of MAC addresses.";
        }
        leaf total-active-mac-num {
          type uint32;
          description
            "Total active MAC entries for the VPN.";
        }
        description
          "MAC statistics.";
      }
    }
  }

  grouping link-loss-statistics {
    description
      "Grouping for per link error statistics.";
    container loss-statistics {
      description
        "Link loss summarized information. By default,
         one way measurement protocol (e.g., OWAMP) is used
```

```
          to measure one-way packet loss.";
      reference
        "RFC 4656: A One-way Active Measurement Protocol (OWAMP)";
      leaf packet-loss-count {
        type yang:counter64;
        description
          "Total received packet drops count.";
      }
      leaf loss-ratio {
        type percentage;
        description
          "Loss ratio of the packets. Express 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
        "Link delay summarized information. By default,
         one way measurement protocol (e.g., OWAMP) is used
         to measure delay.";
      reference
        "RFC 4656: A One-way Active Measurement Protocol (OWAMP)";
      leaf unit-value {
        type identityref {
          base lime:time-unit-type;
        }
        default "lime:milliseconds";
        description
          "Time units, where the options are s, ms, ns, etc.";
      }
      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
        "Link jitter summarized information. By default,
         jitter is measured using one-way IP Packet
         Delay Variation (IPDV).";
      reference
        "RFC 3393: IP Packet Delay Variation Metric
         for IP Performance Metrics (IPPM)";
      leaf unit-value {
        type identityref {
          base lime:time-unit-type;
        }
        default "lime:milliseconds";
        description
          "Time units, where the options are s, ms, ns, etc.";
      }
      leaf min-jitter-value {
        type yang:gauge32;
        description
          "Minimum observed one-way jitter.";
      }
      leaf max-jitter-value {
        type yang:gauge32;
        description
          "Maximum observed one-way jitter.";
      }
      leaf low-jitter-percentile {
        type yang:gauge32;
        description
```

```
          "Low percentile of observed one-way jitter.";
      }
      leaf intermediate-jitter-percentile {
        type yang:gauge32;
        description
          "Intermediate percentile of observed one-way jitter.";
      }
      leaf high-jitter-percentile {
        type yang:gauge32;
        description
          "High percentile of observed one-way jitter.";
      }
    }
  }

  grouping tp-svc-telemetry {
    leaf reference-time {
      type yang:date-and-time;
      config false;
      description
        "Indicates the time when the statistics are collected.";
    }
    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-nunicast {
      type yang:counter64;
      description
        "The total number of inbound non-unicast
         (i.e., broadcast or multicast) packets.";
    }
    leaf inbound-discards {
      type yang:counter64;
      description
        "The number of inbound packets that were chosen to be
         discarded even though no errors had been detected.
         One possible reason for discarding such a
         packet could be to free up buffer space";
    }
    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
         which 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-nunicast {
      type yang:counter64;
      description
        "The total number of outbound non unicast
         (i.e., broadcast or multicast) packets.";
    }
    leaf outbound-discards {
      type yang:counter64;
      description
        "The number of outbound packets which were chosen
         to be discarded even though no errors had been
         detected to prevent their being transmitted.
         One possible reason for discarding such a packet could
         be to free up buffer space.";
    }
    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-type {
    presence "Indicates network service topology.";
    leaf service-type {
      type identityref {
        base vpn-common:service-type;
      }
      description
        "The presence identifies the network service type,
         e.g., L3VPN, VPLS, etc.";
    }
    description
      "Container for VPN service type.";
  }
}

augment "/nw:networks/nw:network" {
  when 'nw:network-types/nvp:service-type' {
    description
      "Augments only for VPN Network topology.";
  }
  description
    "Augments the network with service topology attributes";
  container vpn-pm-attributes {
    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,
         hub-spoke-disjoint.";
    }
    description
      "Container for VPN topology attributes.";
  }
}

augment "/nw:networks/nw:network/nw:node" {
  description
    "Augments the network node with other general attributes.";
  container pm-attributes {
    leaf node-type {
      type identityref {
        base node-type;
      }
```

```
      description
        "Node type, e.g., PE, P, ASBR.";
    }
    description
      "Container for node attributes.";
    uses entry-summary;
  }
}

augment "/nw:networks/nw:network/nw:node/pm-attributes" {
  when '../../nw:network-types/nvp:service-type' {
    description
      "Augments only for VPN node attributes.";
  }
  description
    "Augments the network node with VPN specific attributes.";
  leaf role {
    type identityref {
      base vpn-common:role;
    }
    default "vpn-common:any-to-any-role";
    description
      "Role of the node in the VPN.";
  }
}

augment "/nw:networks/nw:network/nt:link" {
  description
    "Augments the network topology link with performance
     monitoring attributes.";
  container pm-attributes {
    description
      "Container for PM attributes.";
    leaf low-percentile {
      type percentile;
      default "10.00";
      description
        "Low percentile to report. Setting low-percentile
         into 0.00 indicates the client is not interested
         in receiving low percentile.";
    }
    leaf intermediate-percentile {
      type percentile;
      default "50.00";
      description
        "Intermediate percentile to report. Setting
         intermediate-percentile into 0.00 indicates the client
         is not interested in receiving intermediate percentile.";
    }
```

```
leaf high-percentile {
  type percentile;
  default "95.00";
  description
    "High percentile to report. Setting high-percentile
     into 0.00 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.";
}
leaf start-time {
  type yang:date-and-time;
  config false;
  description
    "The time that the current measurement started.";
}
leaf end-time {
  type yang:date-and-time;
  config false;
  description
    "The time that the current measurement 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/pm-attributes" {
  when '../../nw:network-types/nvp:service-type' {
    description
      "Augments only for VPN Network topology.";
  }
  description
    "Augments the network topology link with VPN service
     performance monitoring attributes.";
  choice vpn-pm-type {
    description
      "The VPN PM type of this logical point-to-point
       unidirectional VPN link.";
    case inter-vpn-access-interface {
      leaf inter-vpn-access-interface {
        type empty;
        description
          "This is a placeholder for inter-vpn-access-interface PM.
           There is no technology to be defined.";
      }
    }
    case underlay-tunnel {
      leaf vpn-underlay-transport-type {
        type identityref {
          base vpn-common:protocol-type;
        }
        config false;
        description
          "The leaf indicates the underlay transport type of
           a VPN service, e.g., GRE, LDP, etc.";
      }
    }
  }
}

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-type' {
      description
        "Augments only for VPN Network topology.";
    }
    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
          "References to an identifier for the VPN network
           access, e.g. L3VPN or VPLS.";
      }
      uses tp-svc-telemetry;
    }
  }
}

<CODE ENDS>
```

6. Security Considerations

The YANG modules defined in this document MAY be accessed via the RESTCONF protocol [RFC8040] or NETCONF protocol [RFC6241]. The lowest RESTCONF or NETCONF layer requires that the transport-layer protocol provides both data integrity and confidentiality, see Section 2 in [RFC8040] and [RFC6241]. 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 with the write operation that can be exploited to impact the network monitoring:

* "/nw:networks/nw:network/nw:network-types"

* "/nw:networks/nw:network/nvp:vpn-pm-attributes"

* "/nw:networks/nw:network/nw:node/nvp:pm-attributes"

* /nw:networks/nw:network/nt:link/nvp:pm-attributes"

* /nw:networks/nw:network/nw:node/nt:termination-point/nvp:pm-statistics"

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. The nodes reveals the quality of a service that is operated by an operator. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

* "/nw:networks/nw:network/nw:node/nvp:pm-attributes/nvp:vpn-summary-statistics": Unauthorized access to this subtree can disclose the operational state information of VPN instances.

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

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

7. IANA Considerations

This document requests IANA to register 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.

This document requests IANA to register 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 XXXX (RFC Ed.: replace XXXX with actual
     RFC number and remove this note.)
```

8. 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.

9. 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: xuhl.bri@chinatelecom.cn

10. References

10.1. 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>.

[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>.

[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>.

[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>.

[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>.

10.2. Informative References

[I-D.ietf-netmod-node-tags] Wu, Q., Claise, B., Liu, P., Du, Z., and M. Boucadair, "Self-Describing Data Object Tags in YANG Data Models", Work in Progress, Internet-Draft, draft-ietf-netmod-node-tags-06, 21 February 2022, <https://www.ietf.org/archive/id/draft-ietf-netmod-node-tags-06.txt>.

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

[I-D.ietf-opsawg-sap] Boucadair, M., Dios, O. G. D., Barguil, S., Wu, Q., and V. Lopez, "A Network YANG Model for Service Attachment Points (SAPs)", Work in Progress, Internet-Draft, draft-ietf-opsawg-sap-02, 22 February 2022, <https://www.ietf.org/archive/id/draft-ietf-opsawg-sap-02.txt>.

[ITU-T-Y-1731] ITU-T, "Operator Ethernet Service Definition", August 2015, <https://www.itu.int/rec/T-REC-Y.1731/en>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

Appendix A. Illustrating Examples

A.1. VPN Performance Subscription Example

The example shown in Figure 8 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.

```
   POST /restconf/operations
        /ietf-subscribed-notifications:establish-subscription
{
   "ietf-subscribed-notifications:input":{
      "stream-subtree-filter":{
         "ietf-network-topo:networks":{
            "network":{
               "network-id":"l3-network",
               "ietf-network-vpn-pm:service-type":{
                  "ietf-vpn-common:l3vpn":{}
               },
               "node":[
                  {
                     "node-id":"A",
                     "ietf-network-vpn-pm:pm-attributes":{
                        "node-type":"PE"
                     },
                     "termination-point":{
                        "tp-id":"1-0-1"
                     }
                  },
                  {
                     "node-id":"B",
                     "ietf-network-vpn-pm:pm-attributes":{
                        "node-type":"PE"
                     },
                     "termination-point":{
                        "tp-id":"2-0-1"
                     }
                  }
               ],
               "link":{
                  "link-id":"A-B",
                  "source":{
                     "source-node":"A"
                  },
                  "destination":{
                     "dest-node":"B"
                  },
                  "ietf-network-vpn-pm:pm-attributes":{
                     "one-way-pm-statistics":{
                        "loss-statistics":{
                           "packet-loss-count":{}
                        }
                     },
                     "vpn-underlay-transport-type":"ietf-vpn-common:gre"
                  }
               }
            }
```

```
            }
         },
         "ietf-yang-push:periodic":{
            "ietf-yang-push:period":"500"
         }
      }
   }
```

Figure 8: Pub/Sub Retrieval

A.2. Example of VPN Performance Snapshot

This example, depicted in Figure 9, illustrates an VPN PM instance example in which a client uses RESTCONF [RFC8040] to fetch the performance data of the link and TP belonged to "VPN1".

```
{
      "ietf-network-topo:networks": {
        "network": {
          "network-id": "foo:vpn1",
          "node": [
            {
              "node-id": "A",
              "ietf-network-vpn-pm:pm-attributes": {
                "node-type": "PE"
              },
              "termination-point": {
                "tp-id": "1-0-1",
                "ietf-network-vpn-pm:pm-statistics": {
                  "inbound-octets": "100",
                  "outbound-octets": "150"
                }
              }
            },
            {
              "node-id": "B",
              "ietf-network-vpn-pm:pm-attributes": {
                "node-type": "PE"
              },
              "termination-point": {
                "tp-id": "2-0-1",
                "ietf-network-vpn-pm:pm-statistics": {
                  "inbound-octets": "150",
                  "outbound-octets": "100"
                }
              }
            }
          ],
          "link": {
            "link-id": "A-B",
            "source": { "source-node": "A" },
            "destination": { "dest-node": "B" },
            "ietf-network-pm:pm-attributes": {
              "one-way-pm-statistics": {
                "loss-statistics": { "packet-loss-count": "120" }
              },
              "vpn-underlay-transport-type": "ietf-vpn-common:gre"
            }
          }
        }
      }
}
```

Figure 9

A.3. Example of Percentile Monitoring

The following shows an example of a percentile measurement for a VPN link.


```
{
  "ietf-network-topology:link": [
    {
      "link-id": "vpn1-link1",
      "source": {
        "source-node": "vpn-node1"
      },
      "destination": {
        "dest-node": "vpn-node3"
      },
      "ietf-network-vpn-pm:pm-attributes": {
        "low-percentile": "20.00",
        "middle-percentile": "50.00",
        "high-percentile": "90.00",
        "one-way-pm-statistics": {
          "delay-statistics": {
            "unit-values": "lime:milliseconds",
            "min-delay-value": "43",
            "max-delay-value": "99",
            "low-delay-percentile": "64",
            "intermediate-delay-percentile": "77",
            "high-delay-percentile": "98"
          }
        },
        "ietf-network-vpn-pm:inter-vpn-access-interface": [null]
      }
    }
  ]
}
```


Authors' Addresses

Bo Wu (editor)
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China

Email: lana.wubo@huawei.com

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

Oscar Gonzalez de Dios
Telefonica
Madrid
Spain

Email: oscar.gonzalezdedios@telefonica.com

Bin Wen
Comcast

Email: bin_wen@comcast.com