OPSAWG Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 25, 2021

B. Wu
Q. Wu
Huawei
M. Boucadair
Orange
O. Gonzalez de Dios
Telefonica
B. Wen
Comcast
C. Liu
China Unicom
H. Xu
China Telecom

January 21, 2021

A YANG Model for Network and VPN Service Performance Monitoring
draft-www-opsawg-yang-vpn-service-pm-03

Abstract

The data model defined in RFC8345 introduces vertical layering relationships between networks that can be augmented to cover network/service topologies. This document defines a YANG model for both Network Performance Monitoring and VPN Service Performance Monitoring that can be used to monitor and manage network performance on the topology at higher layer or the service topology between VPN sites.

This document does not define metrics for network performance or mechanisms for measuring network performance. The YANG model defined in this document is designed as an augmentation to the network topology YANG model defined in RFC 8345 and draws on relevant YANG types defined in RFC 6991, RFC 8299, RFC 8345, and RFC 8532.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[RFC4176] provides a framework for L3VPN operations and management and specifies that performance management is required after service configuration. This document defines a YANG Model for both network performance monitoring and VPN service performance monitoring that can be used to monitor and manage network performance on the topology level or the service topology between VPN sites.

This document does not introduce new metrics for network performance or mechanisms for measuring network performance, but uses the existing mechanisms and statistics to show the performance monitoring statistics at the network and service layers. The YANG model 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], [RFC8299], [RFC8345], and [RFC8532].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

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

3. Network and VPN Service Performance Monitoring Model Usage

Models are key for automatic management operations. According to [I-D.ietf-opsawg-model-automation-framework] , together with service and network models, performance measurement telemetry model can monitor network performance to meet specific service SLA requirements. The model defined in this document is to derive VPN or network level performance data based on lower-level data collected via monitoring counters in the devices.

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

Figure 1: Reference Architecture

As shown in Figure 1 , the network and VPN service performance monitoring model can be used to expose some performance information to the above layer. The information can be used by the orchestrator to subscribe to performance data. The controller will then notify the orchestrator of corresponding parameter changes.

Before using the Network and VPN Service PM Model, the mapping between the VPN Service topology and the underlying physical network has been setup, and the performance monitoring data per link in the underlying network can be collected using network performance measurement method such as MPLS Loss and Delay Measurement [RFC6374].

The performance monitoring information reflecting the quality of the Network or VPN service such as end to end network performance data between source node and destination node in the network or between VPN sites can be aggregated or calculated using, for example, PCEP solution [RFC8233] [RFC7471] [RFC8570] [RFC8571] or LMAP [RFC8194].

The measurement interval and report interval associated with these performance data usually depends on configuration parameters.

3.1. Retrieval via Pub/Sub Mechanism

Some applications such as service-assurance applications, which must maintain a continuous view of operational data and state, can use subscription model [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. On demand Retrieval via RPC Model

To obtain a snapshot of a large amount of performance data from a network element (including network controllers), service-assurance applications may use polling-based methods such as RPC model to fetch performance data on demand.

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 is augmented to 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 [RFC7950] 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 mapping between the VPN service topology and an underlying network:

```
                VPN-SVC 1           VPN-SVC 2
                   /                     \
      VPN-Service-topology 1     VPN-Service-topology-2
          /     |      \            /     |      \
     Site-1A Site-1B Site1-C    Site-2A Site-2B Site-2C    Top-Down
       |          |       |       |         |     |     Service Topology
       CE         CE     CE      CE        CE     CE
       |          |       |       |         |     |
       PE         PE     PE      PE        PE     PE
   ====|==========|=======|=======|=========|=====|====================
       +-------+  |        \    /           /     |
    Bottom-up  |  |         \ /           /       |
    Network    |  |         /\           /        |
    topology   |  |       /    \        |         |
               |  |      |       |      |         |
           node1 node2 node3   node4   node5    node6
```

Figure 3: Example of topology mapping between VPN Service Topo and Underlying network

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

- VPN-SVC 1: supporting "hub-spoke" communications for Customer 1 connecting the customer's access at 3 sites. Site-1A, Site-1B, and Site-1C are connected to PEs that are mapped to nodes 1, 2, and 3 in the underlying physical network.
  Site-1 A plays the role of hub while Site-2 B and C plays the role of spoke.

- VPN-SVC 2: supporting "hub-spoke disjoint" communications for Customer 2 connecting the customer's access at 3 sites. Site-2A, Site-2B, and Site-2C are connected to PEs that are mapped to nodes 4, 5, and 6 in the underlying physical network.

  Site-2 A and B play the role of hub while Site-2 C plays the role of spoke.

4.2. Network Level

For network performance monitoring, the attributes of "Network Level" that defined in [RFC8345] do not need to be extended.

For VPN service performance monitoring, this document defines some new network service type: "L3VPN, L2VPN". When a network topology data instance contains the L3VPN or L2VPN network type, it represents an VPN instance that can perform performance monitoring.

This model defines only the following minimal set of Network level network topology attributes:

- "vpn-id": Refers to an identifier of VPN service (e.g.,L3NM[I-D.ietf-opsawg-l3sm-l3nm]). This identifier allows to correlate the performance status with the network service configuration.

- "vpn-topo": The type of VPN service 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). [RFC8299] defines a YANG model for L3VPN Service Delivery. Three types of VPN service topologies are supported in : "any to any", "hub and spoke", and "hub and spoke disjoint". 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 network-service-type!
       +--rw network-service-type?   identityref
  augment /nw:networks/nw:network:
    +--rw vpn-topo-attributes
       +--rw vpn-id?    vpn-common:vpn-id
       +--rw vpn-topology?   identityref
```

Figure 4: Network Level View of the hierarchies

4.3. Node Level

For network performance monitoring, the attributes of "Node Level" that defined in [RFC8345] do not need to be extended.

For VPN service performance monitoring, this model defines only the following minimal set of Node level network topology attributes:

- "node-type" (Attribute): Indicates the type of the node, such as PE or ASBR. This "node-type" can be used to report performance metric between any two nodes each with specific node-type.

- "site-id" (Constraint): Uniquely identifies the site within the overall network infrastructure.

- "site-role" (Constraint): Defines the role of the site in a particular VPN topology.

- "vpn-summary-statistics": IPv4 statistics, and IPv6 statistics have been specified separately. And MAC statistics could be extended for L2VPN.

```
  augment /nw:networks/nw:network/nw:node:
    +--rw node-attributes
    |  +--rw node-type?   identityref
    |  +--rw site-id?     string
    |  +--rw site-role?   identityref
    +--rw vpn-summary-statistics
       +--rw ipv4
       |  +--rw total-routes?          uint32
       |  +--rw total-active-routes?   uint32
       +--rw ipv6
          +--rw total-routes?          uint32
          +--rw total-active-routes?   uint32
```

Figure 5: Node Level View of the hierarchies

4.4. Link and Termination Point Level

The link nodes are classified into two types: one is topology link defined in [RFC8345], and the other is abstract link of a VPN between PEs.

The performance data of the link is a collection of counters that report the performance status. The data for the topology link can be based on BGP-LS [RFC8571]. The statistics of the VPN abstract links can be collected based on VPN OAM mechanisms, e.g. TWAMP etc. Alternatively, the data can base on the underlay technology OAM mechanism, for example, GRE tunnel OAM.

```
     augment /nw:networks/nw:network/nt:link:
       +--rw link-type?   identityref
     augment /nw:networks/nw:network/nt:link:
       +--rw low-percentile?              percentile
       +--rw middle-percentile?           percentile
       +--rw high-percentile?             percentile
       +--rw reference-time?              yang:date-and-time
       +--rw measurement-interval?        uint32
       +--ro link-telemetry-attributes
          +--ro loss-statistics
          |  +--ro packet-loss-count?          yang:counter32
          |  +--ro packet-reorder-count?       yang:counter32
          |  +--ro packets-out-of-seq-count?   yang:counter32
          |  +--ro packets-dup-count?          yang:counter32
          |  +--ro loss-ratio?                 percentage
          +--ro delay-statistics
          |  +--ro direction?                 identityref
          |  +--ro unit-value?                identityref
          |  +--ro min-delay-value?           yang:gauge64
          |  +--ro max-delay-value?           yang:gauge64
          |  +--ro low-delay-percentile?      yang:gauge64
          |  +--ro middle-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 middle-jitter-percentile?   yang:gauge32
             +--ro high-jitter-percentile?     yang:gauge32
     augment /nw:networks/nw:network/nw:node/nt:termination-point:
       +--ro tp-telemetry-attributes
          +--ro inbound-octets?             yang:counter64
          +--ro inbound-unicast?            yang:counter64
          +--ro inbound-nunicast?           yang:counter64
          +--ro inbound-discards?           yang:counter32
          +--ro inbound-errors?             yang:counter32
          +--ro inbound-unknown-protocol?   yang:counter32
          +--ro outbound-octets?            yang:counter64
          +--ro outbound-unicast?           yang:counter64
          +--ro outbound-nunicast?          yang:counter64
          +--ro outbound-discards?          yang:counter32
          +--ro outbound-errors?            yang:counter32
          +--ro outbound-qlen?              uint32
```

Figure 6: Link and Termination point Level View of the hierarchies

For the nodes of the link in the figure, this module defines the following minimal set of link level performance attributes:

o "link-type": Indicates the abstract link of a VPN, such as GRE or IP-in-IP. The leaf refers to an identifier of VPN Common "underlay-transport" [I-D.ietf-opsawg-vpn-common], which describes the transport technology to carry the traffic of the VPN service.

o Percentile parameters: The module supports reporting delay and jitter metric by percentile values. By default, low percentile (10th percentile), mid percentile (50th percentile), high percentile (90th percentile) are used. Setting a percentile into 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 "reference-time" "measurement-interval", the high-delay-percentile and high-jitter-percentile will be reported.

o Loss Statistics: A set of 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.

o Delay Statistics: A set of 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.

o Jitter Statistics: A set of 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.

For the nodes of "termination points" in the figure, the module defines the following minimal set of statistics:

o 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.

o 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.

5. Example of I2RS Pub/Sub Retrieval

This example shows the way for a client to subscribe for the Performance monitoring information between node A and node B in the L3 network topology built on top of the underlying network . The performance monitoring parameter that the client is interested in is end to end loss attribute.

```
    <rpc netconf:message-id="101"
       xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
       <establish-subscription
      xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications">
          <stream-subtree-filter>
             <networks
        xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topo">
                <network>
                 <network-id>l3-network</network-id>
                 <network-service-type
      xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
                    L3VPN
                 </network-service-type>
                   <node>
                     <node-id>A</node-id>
                     <node-attributes
      xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
                       <node-type>pe</node-type>
                     </node-attribtues>
                     <termination-point
      xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topology">
                      <tp-id>1-0-1</tp-id>
                      <tp-telemetry-attributes
      xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
                       <inbound-octets>150</inbound-octets>
                       <outbound-octets>100</outbound-octets>
                      </tp-telemetry-attributes>
                     </termination-point>
                   </node>
                   <node>
                     <node-id>B</node-id>
                     <node-attributes
           xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
                      <node-type>pe</node-type>
                     </node-attribtues>
                       <termination-point
           xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topology">
```

```
                         <tp-id>2-0-1</tp-id>
                         <tp-telemetry-attributes
           xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
                           <inbound-octets>150</inbound-octets>
                           <outbound-octets>100</outbound-octets>
                         </tp-telemetry-attributes>
                      </termination-point>
                    </node>
                    <link
             xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topology">
                     <link-id>A-B</link-id>
                      <source>
                        <source-node>A</source-node>
                      </source>
                      <destination>
                       <dest-node>B</dest-node>
                      </destination>
                      <link-type>mpls-te</link-type>
                      <link-telemetry-attributes
               xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
                        <loss-statistics>
                         <packet-loss-count>100</packet-loss-count>
                        </loss-statistics>
                      </link-telemetry-attributes>
                    </link>
                </network>
              </networks>
           </stream-subtree-filter>
         <period
           xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
          500
       </period>
       </establish-subscription>
    </rpc>
```

6. Example of RPC-based Retrieval

This example shows the way for the client to use RPC model to fetch performance data on demand, e.g., the client requests "packet-loss-count" between PE1 in site 1 and PE2 in site 2 belonging to the same VPN1.

```
  <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
         message-id="1">
    <report
         xmlns="urn:ietf:params:xml:ns:yang:example-service-pm-report">
       <networks xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topo">
         <network>
```

```
          <network-id>vpn1</network-id>
          <node>
           <node-id>A</node-id>
           <node-attributes
                xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
             <node-type>pe</node-type>
           </node-attribtues>
           <termination-point
              xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topology">
            <tp-id>1-0-1</tp-id>
            <tp-telemetry-attributes
               xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
             <inbound-octets>100</inbound-octets>
             <outbound-octets>150</outbound-octets>
            </tp-telemetry-attributes>
           </termination-point>
          </node>
          <node>
           <node-id>B</node-id>
           <node-attributes
               xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
             <node-type>pe</node-type>
           </node-attribtues>
           <termination-point
              xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topology">
            <tp-id>2-0-1</tp-id>
            <tp-telemetry-attributes
                xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
             <inbound-octets>150</inbound-octets>
             <outbound-octets>100</outbound-octets>
            </tp-telemetry-attributes>
           </termination-point>
          </node>
          <link>
          <link-id>A-B</link-id>
           <source>
            <source-node>A</source-node>
           </source>
           <destination>
            <dest-node>B</dest-node>
           </destination>
          <link-type>mpls-te</link-type>
           <telemetry-attributes
              xmlns="urn:ietf:params:xml:ns:yang:ietf-network-pm">
            <loss-statistics>
             <packet-loss-count>120</packet-loss-count>
            </loss-statistics>
           </telemetry-attributes>
```

```
          </link>
        </network>
      </report>
    </rpc>
```

7. Network and VPN Service Assurance YANG Module

This module uses types defined in [RFC8345], [RFC8299] and [RFC8532].

```
<CODE BEGINS> file "ietf-network-vpn-pm@2021-01-15.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;
  }
  import ietf-network {
    prefix nw;
    reference
      "Section 6.1 of RFC 8345: A YANG Data Model for Network
       Topologies";
  }
  import ietf-network-topology {
    prefix nt;
    reference
      "Section 6.2 of RFC 8345: A YANG Data Model for Network
       Topologies";
  }
  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 Working Group";
  contact
    "Editor: Qin Wu
             <bill.wu@huawei.com>
```

```
     Editor: Bo Wu
             <lana.wubo@huawei.com>
     Editor: Mohamed Boucadair
             <mohamed.boucadair@orange.com>";
  description
    "This module defines a model for Network and VPN Service Performance
     monitoring.

     Copyright (c) 2021 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

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

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

  revision 2021-01-15 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Model for Network and VPN Service Performance
                 Monitoring";
  }

  identity pe {
    base vpn-common:role;
    description
      "Identity for PE type";
  }

  identity ce {
    base vpn-common:role;
    description
      "Identity for CE type";
  }

  identity asbr {
    base vpn-common:role;
    description
      "Identity for ASBR type";
  }

  identity p {
```

```
    base vpn-common:role;
    description
      "Identity for P type";
  }

  identity link-type {
    base vpn-common:protocol-type;
    description
      "Base identity for link type, e.g.,GRE, MPLS TE, VXLAN.";
  }

  identity VXLAN {
    base link-type;
    description
      "Base identity for VXLAN Tunnel.";
  }

  identity ip-in-ip {
    base link-type;
    description
      "Base identity for IP in IP Tunnel.";
  }

  identity direction {
    description
      "Base Identity for measurement direction including
       one way measurement and two way measurement.";
  }

  identity one-way {
    base direction;
    description
      "Identity for one way measurement.";
  }

  identity two-way {
    base direction;
    description
      "Identity for two way measurement.";
  }

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

```
  }

  typedef percentile {
    type decimal64 {
      fraction-digits 5;
    }
    description
      "The percentile is a statistical value that indicates that a
       certain percentage of a set of data falls below it.";
  }

  grouping vpn-summary-statistics {
    description
      "VPN Statistics grouping used for network topology
       augmentation.";
    container vpn-summary-statistics {
      description
        "Container for VPN summary statistics.";
      container ipv4 {
        leaf total-routes {
          type uint32;
          description
            "Total routes for the VPN.";
        }
        leaf total-active-routes {
          type uint32;
          description
            "Total active routes for the VPN.";
        }
        description
          "IPv4-specific parameters.";
      }
      container ipv6 {
        leaf total-routes {
          type uint32;
          description
            "Total routes for the VPN.";
        }
        leaf total-active-routes {
          type uint32;
          description
            "Total active routes for the VPN.";
        }
        description
          "IPv6-specific parameters.";
      }
    }
  }
```

```
  grouping link-error-statistics {
    description
      "Grouping for per link error statistics.";
    container loss-statistics {
      description
        "Per link loss statistics.";
      leaf packet-loss-count {
        type yang:counter32;
        description
          "Total received packet drops count.";
      }
      leaf packet-reorder-count {
        type yang:counter32;
        description
          "Total received packet reordered count.";
      }
      leaf packets-out-of-seq-count {
        type yang:counter32;
        description
          "Total received out of sequence count.";
      }
      leaf packets-dup-count {
        type yang:counter32;
        description
          "Total received packet duplicates 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 summarised information. By default,
         one way measurement protocol (e.g., OWAMP) is used
         to measure delay.";
      leaf direction {
        type identityref {
          base direction;
        }
        default "one-way";
```

```
        description
          "Define measurement direction including one way
           measurement and two way measurement.";
      }
      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 delay value observed.";
      }
      leaf max-delay-value {
        type yang:gauge64;
        description
          "Maximum delay value observed.";
      }
      leaf low-delay-percentile {
        type yang:gauge64;
        description
          "Low percentile of the delay observed with
           specific measurement method.";
      }
      leaf middle-delay-percentile {
        type yang:gauge64;
        description
          "Middle percentile of the delay observed with
           specific measurement method.";
      }
      leaf high-delay-percentile {
        type yang:gauge64;
        description
          "High percentile of the delay observed with
           specific measurement method.";
      }
    }
  }

  grouping link-jitter-statistics {
    description
      "Grouping for per link jitter statistics";
    container jitter-statistics {
      description
```

```
        "Link jitter summarised information. By default,
         jitter is measured using IP Packet Delay Variation
         (IPDV).";
      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 jitter value observed.";
      }
      leaf max-jitter-value {
        type yang:gauge32;
        description
          "Maximum jitter value observed.";
      }
      leaf low-jitter-percentile {
        type yang:gauge32;
        description
          "Low percentile of the jitter observed.";
      }
      leaf middle-jitter-percentile {
        type yang:gauge32;
        description
          "Middle percentile of the jitter observed.";
      }
      leaf high-jitter-percentile {
        type yang:gauge32;
        description
          "High percentile of the jitter observed.";
      }
    }
  }

  grouping tp-svc-telemetry {
    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
            "Inbound unicast packets were received, and delivered
             to a higher layer during the last period.";
        }
        leaf inbound-nunicast {
          type yang:counter64;
          description
            "The number of non-unicast (i.e., subnetwork-
             broadcast or subnetwork-multicast) packets
             delivered to a higher-layer protocol.";
        }
        leaf inbound-discards {
          type yang:counter32;
          description
            "The number of inbound packets which were chosen
             to be discarded even though no errors had been
             detected to prevent their being deliverable to a
             higher-layer protocol.";
        }
        leaf inbound-errors {
          type yang:counter32;
          description
            "The number of inbound packets that contained
             errors preventing them from being deliverable to a
             higher-layer protocol.";
        }
        leaf inbound-unknown-protocol {
          type yang:counter32;
          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 packets that higher-level
             protocols requested be transmitted to a
             subnetwork-unicast address, including those that
             were discarded or not sent.";
        }
        leaf outbound-nunicast {
```

```
      type yang:counter64;
      description
        "The total number of packets that higher-level
         protocols requested be transmitted to a non-
         unicast (i.e., a subnetwork-broadcast or
         subnetwork-multicast) address, including those
         that were discarded or not sent.";
    }
    leaf outbound-discards {
      type yang:counter32;
      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:counter32;
      description
        "The number of outbound packets that contained
         errors preventing them from being deliverable to a
         higher-layer protocol.";
    }
    leaf outbound-qlen {
      type uint32;
      description
        " Length of the queue of the interface from where
          the packet is forwarded out.  The queue depth could
           be the current number of memory buffers used by the
          queue and a packet can consume one or more memory buffers
          thus constituting device-level information.";
    }
    description
      "Grouping for interface service telemetry.";
  }

  augment "/nw:networks/nw:network/nw:network-types" {
    description
      "Defines the service topologyies types";
    container network-service-type {
      presence "Indicates Network service topology";
      leaf network-service-type {
        type identityref {
          base vpn-common:service-type;
        }
        description
          "The presence identifies the network service type,
```

```
             e.g., L3VPN, L2VPN, etc.";
        }
        description
          "Container for vpn service type.";
      }
    }

    augment "/nw:networks/nw:network" {
      when 'nw:network-types/nvp:network-service-type' {
        description
          "Augment only for VPN Network topology.";
      }
      description
        "Augment the network with service topology attributes";
      container vpn-topo-attributes {
        leaf vpn-id {
          type vpn-common:vpn-id;
          description
            "Pointer to the parent VPN service(e.g., L3NM),
             if any.";
        }
        leaf vpn-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" {
      when '../nw:network-types/nvp:network-service-type' {
        description
          "Augment only for VPN Network topology.";
      }
      description
        "Augment the network node with service topology attributes";
      container node-attributes {
        leaf node-type {
          type identityref {
            base vpn-common:role;
          }
          description
            "Node type, e.g., PE, P, ASBR.";
```

```
      }
      leaf site-id {
        type string;
        description
          "Associated vpn site";
      }
      leaf site-role {
        type identityref {
          base vpn-common:role;
        }
        default "vpn-common:any-to-any-role";
        description
          "Role of the site in the VPN.";
      }
      description
        "Container for service topology attributes.";
    }
    uses vpn-summary-statistics;
  }

  augment "/nw:networks/nw:network/nt:link" {
    when '../nw:network-types/nvp:network-service-type' {
      description
        "Augment only for VPN Network topology.";
    }
    description
      "Augment the network topology link with service topology
       attributes";
    leaf link-type {
      type identityref {
        base vpn-common:protocol-type;
      }
      description
        "Underlay-transport type, e.g., GRE, LDP, etc.";
    }
  }

  augment "/nw:networks/nw:network/nt:link" {
    description
      "Augment the network topology link with service topology
       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 middle-percentile {
      type percentile;
      default "50.00";
      description
        "Middle percentile to report. Setting middle-percentile
         into 0.00 indicates the client is not interested in receiving
         middle percentile.";
    }
    leaf high-percentile {
      type percentile;
      default "90.00";
      description
        "High percentile to report. Setting high-percentile
         into 0.00 indicates the client is not interested in receiving
         high percentile";
    }
    leaf reference-time {
      type yang:date-and-time;
      description
        "The time that the current Measurement Interval started.";
    }
    leaf measurement-interval {
      type uint32;
      units "seconds";
      default "60";
      description
        "Interval to calculate performance metric.";
    }
    container link-telemetry-attributes {
      config false;
      uses link-error-statistics;
      uses link-delay-statistics;
      uses link-jitter-statistics;
      description
        "Container for service telemetry attributes.";
    }
  }

  augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
    description
      "Augment the network topology termination point with vpn
       service attributes";
    container tp-telemetry-attributes {
      config false;
      uses tp-svc-telemetry;
      description
        "Container for termination point service telemetry attributes.";
```

```
    }
  }
}
<CODE ENDS>
```

8. 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 and data nodes and their sensitivity/vulnerability:

o /nw:networks/nw:network/svc-topo:svc-telemetry-attributes

o /nw:networks/nw:network/nw:node/svc-topo:node-attributes

9. 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
```

10. Acknowledgements

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

11. Contributors

Michale Wang
Huawei
Email:wangzitao@huawei.com

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

12. References

12.1. Normative References

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

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <https://www.rfc-editor.org/info/rfc8299>.

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

[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>.

12.2. Informative References

[I-D.ietf-opsawg-l3sm-l3nm] barguil, s., Dios, O., Boucadair, M., Munoz, L., and A. Aguado, "A Layer 3 VPN Network YANG Model", draft-ietf-opsawg-l3sm-l3nm-05 (work in progress), October 2020.

[I-D.ietf-opsawg-model-automation-framework] WU, Q., Boucadair, M., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", draft-ietf-opsawg-model-automation-framework-10 (work in progress), October 2020.

[I-D.ietf-opsawg-vpn-common] barguil, s., Dios, O., Boucadair, M., and Q. WU, "A Layer 2/3 VPN Common YANG Model", draft-ietf-opsawg-vpn-common-03 (work in progress), January 2021.

[RFC4176] El Mghazli, Y., Ed., Nadeau, T., Boucadair, M., Chan, K., and A. Gonguet, "Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management", RFC 4176, DOI 10.17487/RFC4176, October 2005, <https://www.rfc-editor.org/info/rfc4176>.

[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>.

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

[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>.

[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>.

Authors' Addresses

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

Email: lana.wubo@huawei.com

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

Email: bill.wu@huawei.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Oscar Gonzalez de Dios
Telefonica
Madrid
ES

Email: oscar.gonzalezdedios@telefonica.com

Bin Wen
Comcast

Email: bin_wen@comcast.com

Change Liu
China Unicom

Email: liuc131@chinaunicom.cn

Honglei Xu
China Telecom

Email: xuhl.bri@chinatelecom.cn