BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2019

M. Wang
Q. Wu
R. Even
Huawei
B. Wen
Comcast
C. Liu
China Unicom

March 8, 2019

A YANG Model for Network and VPN Service Performance Monitoring
draft-www-bess-yang-vpn-service-pm-02

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 Network and VPN Service Performance Monitoring that can be used to monitor and manage network performance on the topology at different layer or the overlay topology between VPN sites. This model is an augmentation to the network topology YANG data model defined in [RFC8345].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[RFC8345] defines an abstract YANG data model for network/service topologies and inventories. Service topology in [RFC8345] includes the a virtual topology for a service layer above the L1, L2, and L3 layers. This virtual 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.

In [RFC8299], 3 types of VPN service topologies are defined for the L3VPN service data model: 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.

This document defines a YANG Model for Network performance monitoring and VPN Service Performance Monitoring that can be used to monitor and manage network Performance on the topology at different layer or the overlay topology between VPN sites and it is an augmentation to the network topology YANG data model defined in [RFC8345].

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC2119] significance.

2.1. Tree Diagrams

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

3. Network and VPN service assurance model

This module describes Network and VPN Service assurance that can be used to monitor and manage the network Performance in the underlying network or between VPN sites and it is a augmentation to the "ietf-network" and "ietf-network-topology" YANG data model [RFC8345]. The performance monitoring data is augmented to service topology.

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

4. Layering relationship between underlay topology and overlay topology

The data model defined in [RFC8345] can describe vertical layering relationships between networks. That model can be augmented to cover network/service topologies. Figure 1 describes an example on layering relationship between L3 topology and the Optical topology:

```
                       +----:--------------:----:--------------+
                      /      :              :   :        "L3" /
                     /        :              :  :            /
                    /         :               : :           /
                   /         [Y1]_____________[Y2]         /
                  /           *               * *         /
                 /            *              *  *        /
                +--------------*-------------*--*-------+
                                *           *   *
                       +--------*----------*----*--------------+
                      /     [Z1]_______________[Z2] "Optical" /
                     /        \_         *   _/             /
                    /           \_      *  _/              /
                   /              \_   * _/               /
                  /                 \ * /                /
                 /                    [Z]               /
                +---------------------------------------+
```

Example of Layering relationship between the L3 Topology and the Optical topology

The "L3" topology shows network elements at Layer 3 (IP), and the "Optical" topology shows network elements at Layer 1. Network elements in the "L3" topology are mapped onto network elements in the "Optical" topology.

Figure 2 describes another example on relationships between the L3VPN service topology and the underlying network:

```
                VPN-SVC 1           VPN-SVC 2
                   /                     \
    L3VPN-Service-topology 1  L3VPN-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
```

Example of Layering relationships between L3VPN Service Topo and Underlying network

As shown in Figure 2, Site-1A, Site-1B, and Site-1C are mapped to nodes 1, 2, and 3, while Site-2A, Site-2B, and Site-2C are mapped to nodes 4, 5, and 6 in the underlying physical network. In this figure, a L3VPN Service Model (L3SM) [RFC8299] has two VPN services topologies with both built on top of one common underlying physical network.

VPN-SVC 1: supporting hub-spoke communication for Customer 1 connecting the customers access at 3 sites

VPN-SVC 2: supporting hub-spoke disjoint communication for Customer 2 connecting the customers access at 3 sites

L3VPN service topology 1 is hub and spoke topology while L3VPN service topology 2 is hub and spoke disjoint topology. In L3VPN service topology 1, Site-1 A plays the role of hub while Site-2 B and C plays the role of spoke. In L3VPN service topoogy 2, Site-2 A and B play the role of hub while Site-2 C plays the role of spoke.

5. Model Usage Guideline

An SP must be able to manage the capabilities and characteristics of their Network/VPN services when Network connection is established or VPN sites are setup to communicate with each other. Network and VPN service topology such as hub and spoke describes how these VPN sites are communicating with each other.

5.1. Performance Monitoring Data Source

As described in section 2, once the mapping between overlay network topology/VPN Service topology and underlying physical network has been setup, 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 PCEP solution [RFC5440] or LMAP solution [RFC8194]. The information can be fed into data source such as the management system or network devices. The measurement interval and report interval associated with these performance data usually depends on configuration parameters.

5.2. Retrieval via I2RS Pub/Sub [RFC7923]

Some applications such as service-assurance applications, which must maintain a continuous view of operational data and state, can use subscription model [I-D.ietf-netconf-yang-push] to subscribe to the

Network and 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 push model [I-D.ietf-netconf-yang-push] to publish specific telemetry data to target recipients.

5.3. On demand Retrieval via RPC model

To obtain a snapshot of a large amount of performance data from the network element, service-assurance applications can also use polling based solution such as RPC model to fetch performance data on demand.

6. Design of the Data Model

This document defines the YANG module "ietf-network-vpn-svc-pm", which has the following structure

6.1. Network Level

```
module: ietf-network-vpn-svc-pm
  augment /nw:networks/nw:network/nw:network-types:
    +--rw svc-topo-type?   identityref
  augment /nw:networks/nw:network:
    +--rw svc-topo-attributes
       +--rw vpn-topo?   identityref
```

Network Level View of the hierarchies

The Network and VPN service performance monitoring model defines only the following minimal set of Network level service topology attributes:

- svc-topo-type: Indicate the network type is service topology type such as L3VPN service topology, L2VPN service topology.

- 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).

6.2. Node Level

```
   augment /nw:networks/nw:network/nw:node:
       +--rw node-attributes
          +--rw node-type?   identityref
          +--rw site-id?     string
          +--rw site-role?   Identityref
```

Node Level View of the hierarchies

The Network and VPN service performance monitoring model defines only the following minimal set of Node level service topology attributes and constraints:

- Node-type (Attribute): Indicate the type of the node, such as PE or ASBR.

- 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.

6.3. Link and Termination Point Level

```
     augment /nw:networks/nw:network/nt:link:
       +--ro svc-telemetry-attributes
          +--ro loss-statistics
          |  +--ro direction                   identityref
          |  +--ro packet-loss-count?          uint32
          |  +--ro loss-ratio?                 percentage
          |  +--ro packet-reorder-count?       uint32
          |  +--ro packets-out-of-seq-count?   uint32
          |  +--ro packets-dup-count?          uint32
          +--ro delay-statistics
          |  +--ro direction?                identityref
          |  +--ro min-delay-value?          uint32
          |  +--ro max-delay-value?          uint32
          |  +--ro average-delay-value?      uint32
          +--ro jitter-statistics
             +--ro direction?                identityref
             +--ro min-jitter-value?         uint32
             +--ro max-jitter-value?         uint32
             +--ro average-jitter-value?     uint32
     augment /nw:networks/nw:network/nw:node/nt:termination-point:
       +--ro tp-telemetry-attributes
          +--ro in-octets?           uint32
          +--ro inbound-unicast?     uint32
          +--ro inbound-nunicast?    uint32
          +--ro inbound-discards?    uint32
          +--ro inbound-errors?      uint32
          +--ro inunknow-protos?     uint32
          +--ro out-octets?          uint32
          +--ro outbound-unicast?    uint32
          +--ro outbound-nunicast?   uint32
          +--ro outbound-discards?   uint32
          +--ro outbound-errors?     uint32
          +--ro outbound-qlen?       uint32
```

Link and Termination point Level View of the hierarchies

The Network and VPN service performance monitoring model defines only the following minimal set of Link level service topology attributes:

Loss Statistics: A set of loss statistics attributes that are used to measure end to end loss between VPN sites.

Delay Statistics: A set of delay statistics attributes that are used to measure end to end latency between VPN sites.

Jitter Statistics: A set of jitter statistics attributes that are used to measure end to end jitter between VPN sites.

The Network and VPN service performance monitoring defines the following minimal set of Termination point level service topology attributes:

Inbound statistics: A set of inbound statistics attributes that are used to measure the inbound statistics of the termination point, such as "the total number of octets received on the termination point", "The number of inbound packets which were chosen to be discarded", "The number of inbound packets that contained errors", etc.

Outbound statistics: A set of outbound statistics attributes that are used to measure the outbound statistics of the termination point, such as "the total number of octets transmitted out of the termination point", "The number of outbound packets which were chosen to be discarded", "The number of outbound packets that contained errors", etc.

7. Example of I2RS Pub/Sub Retrieval [RFC7923]

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 optical 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>
                  <node>
                   <node-id>A</node-id>
                   <node-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-
network-vpn-svc-pm">
                    <node-type>pe</node-type>
                   </node-attribtues>
                  </node>
                  <node>
                   <node-id>B</node-id>
                   <node-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-
network-vpn-svc-pm">
                    <node-type>pe</node-type>
                   </node-attribtues>
                  </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>
                    <svc-telemetry-attributes
                     xmlns="urn:ietf:params:xml:ns:yang:ietf-svc-topo">
                     <loss-statistics>
                      <packet-loss-count>100</packet-loss-count>
                     </loss-statistics>
                    </svc-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>
```

8. Example of RPC model 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 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-
svc-pm">
          <node-type>pe</node-type>
          </node-attribtues>
         </node>
         <node>
          <node-id>B</node-id>
          <node-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-
svc-pm">
          <node-type>pe</node-type>
          </node-attribtues>
         </node>
         <link-id>A-B</link-id>
          <source>
           <source-node>A</source-node>
          </source>
          <destination>
           <dest-node>B</dest-node>
          </destination>
          <svc-telemetry-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-
network-vpn-svc-pm">
           <loss-statistics>
            <packet-loss-count>120</packet-loss-count>
           </loss-statistics>
          </svc-telemetry-attributes>
         </link>
       </network>
     </report>
   </rpc>
```

9. Network and VPN Service Assurance YANG Module

```
<CODE BEGINS> file "ietf-network-vpn-svc-pm.yang"
module ietf-network-vpn-svc-pm {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-vpn-svc-pm";
  prefix svc-topo;
```

```
  import ietf-network {
    prefix nw;
  }
  import ietf-network-topology {
    prefix nt;
  }
  import ietf-l3vpn-svc {
    prefix l3vpn-svc;
  }

  organization
    "IETF xxx Working Group";
  contact
    "Zitao Wang: wangzitao@huawei.com
     Qin Wu: bill.wu@huawei.com";
  description
    "This module defines a model for the VPN Service Performance monitoring.";

  revision 2019-03-01 {
    description
      "Initial revision.";
    reference
      "foo";
  }

  identity service-type {
    description
      "Base type for service topology";
  }

  identity l3vpn-svc {
    base service-type;
    description
      "Indentity for layer3 vpn service";
  }

  identity l2vpn-svc {
    base service-type;
    description
      "Identity for layer2 vpn service";
  }

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

  identity pe {
```

```
    base node-type;
    description
      "Identity for PE type";
  }

  identity ce {
    base node-type;
    description
      "Identity for CE type";
  }

  identity asbr {
    base node-type;
    description
      "Identity for ASBR type";
  }

  identity p {
    base node-type;
    description
      "Identity for P type";
  }

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

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

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

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

```
  }

  grouping link-error-statistics {
    description
      "Grouping for per link error statistics";
    container loss-statistics {
      description
        "Per link loss statistics.";
      leaf direction {
        type identityref {
          base direction;
        }
        default "oneway";
        description
          "Define measurement direction including one way
           measurement and two way measurement.";
      }
      leaf packet-loss-count {
        type uint32 {
          range "0..4294967295";
        }
        default "0";
        description
          "Total received packet drops count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
      }
      leaf loss-ratio {
        type percentage;
        description
          "Loss ratio of the packets. Express as percentage
           of packets lost with respect to packets sent.";
      }
      leaf packet-reorder-count {
        type uint32 {
          range "0..4294967295";
        }
        default "0";
        description
          "Total received packet reordered count.
           The value of count will be set to zero (0)
           on creation and will thereafter increase
           monotonically until it reaches a maximum value
           of 2^32-1 (4294967295 decimal), when it wraps
           around and starts increasing again from zero.";
```

```
        }
        leaf packets-out-of-seq-count {
          type uint32 {
            range "0..4294967295";
          }
          description
            "Total received out of sequence count.
             The value of count will be set to zero (0)
             on creation and will thereafter increase
             monotonically until it reaches a maximum value
             of 2^32-1 (4294967295 decimal), when it wraps
             around and starts increasing again from zero..";
        }
        leaf packets-dup-count {
          type uint32 {
            range "0..4294967295";
          }
          description
            "Total received packet duplicates count.
             The value of count will be set to zero (0)
             on creation and will thereafter increase
             monotonically until it reaches a maximum value
             of 2^32-1 (4294967295 decimal), when it wraps
             around and starts increasing again from zero.";
        }
      }
    }

    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 "oneway";
          description
            "Define measurement direction including one way
             measurement and two way measurement.";
        }
        leaf min-delay-value {
          type uint32;
          description
```

```
          "Minimum delay value observed.";
      }
      leaf max-delay-value {
        type uint32;
        description
          "Maximum delay value observed.";
      }
      leaf average-delay-value {
        type uint32;
        description
          "Average delay is calculated on all the packets of a sample
           and is a simple computation to be performed for single marking
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) as defined in RFC3393.";
      leaf direction {
        type identityref {
          base direction;
        }
        default "oneway";
        description
          "Define measurement direction including one way
           measurement and two way measurement.";
      }
      leaf min-jitter-value {
        type uint32;
        description
          "Minimum jitter value observed.";
      }
      leaf max-jitter-value {
        type uint32;
        description
          "Maximum jitter value observed.";
      }
      leaf average-jitter-value {
        type uint32;
        description
          "Average jitter is calculated on all the packets of a sample
```

```
           and is a simple computation to be performed for single marking
method.";
```

```
      }
    }
  }

  grouping tp-svc-telemetry {

    leaf in-octets {
      type uint32;
      description
        "The total number of octets received on the
         interface, including framing characters.";
    }
    leaf inbound-unicast {
      type uint32;
      description
        "Inbound unicast packets were received, and delivered
         to a higher layer during the last period.";
    }
    leaf inbound-nunicast {
      type uint32;
      description
        "The number of non-unicast (i.e., subnetwork-
         broadcast or subnetwork-multicast) packets
         delivered to a higher-layer protocol.";
    }
    leaf inbound-discards {
      type uint32;
      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 uint32;
      description
        "The number of inbound packets that contained
         errors preventing them from being deliverable to a
         higher-layer protocol.";
    }
    leaf inunknow-protos {
      type uint32;
      description
        "The number of packets received via the interface
         which were discarded because of an unknown or
         unsupported protocol";
    }
    leaf out-octets {
```

```
      type uint32;
      description
        "The total number of octets transmitted out of the
         interface, including framing characters";
    }
    leaf outbound-unicast {
      type uint32;
      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 uint32;
      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 uint32;
      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 uint32;
      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
      "Augment the network-types with service topologyies types";
    leaf svc-topo-type {
      type identityref {
        base service-type;
      }
      description
        "Identify the topology type to be composited service topology";
    }
  }
  augment "/nw:networks/nw:network" {
    description
      "Augment the network with service topology attributes";
    container svc-topo-attributes {
      leaf vpn-topology {
        type identityref {
          base l3vpn-svc:vpn-topology;
        }
        description
          "VPN service topology, e.g. hub-spoke, any-to-any, hub-spoke-
disjoint, etc";
      }
      description
        "Container for vpn services";
    }
  }
  augment "/nw:networks/nw:network/nw:node" {
    description
      "Augment the network node with serice attributes";
    container node-attributes {
      leaf node-type {
        type identityref {
          base node-type;
        }
        description
          "Node type, e.g. PE, P, ASBR, etc";
      }
      leaf site-id {
        type string;
        description
          "Asscoiated vpn site";
      }
      leaf site-role {
        type identityref {
          base l3vpn-svc:site-role;
```

```
        }
        default "l3vpn-svc:any-to-any-role";
        description
          "Role of the site in the IP VPN.";
      }
      description
        "Container for service topology attributes";
    }
  }
  augment "/nw:networks/nw:network/nt:link" {
    description
      "Augment the network topology link with vpn service attributes";
    container svc-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>
```

10. 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 [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

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:

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

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

11. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

```
---------------------------------------------------------------------
   URI: urn:ietf:params:xml:ns:yang:ietf-network-vpn-svc-pm

   Registrant Contact: The IESG.

   XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------
```

This document registers a YANG module in the YANG Module Names registry [RFC6020].

```
---------------------------------------------------------------------
   Name:         ietf-vpn-svc-pm
   Namespace:    urn:ietf:params:xml:ns:yang:ietf-network-vpn-svc-pm
   Prefix:       vnrsc
   Reference:    RFC xxxx
---------------------------------------------------------------------
```

12. Normative References

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

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

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[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>.

[RFC6370] Bocci, M., Swallow, G., and E. Gray, "MPLS Transport Profile (MPLS-TP) Identifiers", RFC 6370, DOI 10.17487/RFC6370, September 2011, <https://www.rfc-editor.org/info/rfc6370>.

[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>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <https://www.rfc-editor.org/info/rfc6536>.

[RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <https://www.rfc-editor.org/info/rfc7923>.

[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>.

[RFC7952] Lhotka, L., "Defining and Using Metadata with YANG", RFC 7952, DOI 10.17487/RFC7952, August 2016, <https://www.rfc-editor.org/info/rfc7952>.

[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>.

Authors' Addresses

Michael Wang
Huawei Technologies,Co.,Ltd
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: wangzitao@huawei.com

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

Email: bill.wu@huawei.com

Roni Even
Huawei Technologies,Co.,Ltd
Tel Aviv
Israel

Email: roni.even@huawei.com

Bin Wen
Comcast

Email: bin_wen@comcast.com

Change Liu
China Unicom

Email: liuc131@chinaunicom.cn