OPSAWG Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 3, 2018

Q. Wu
M. Wang
Huawei
M. Boucadair
Orange

March 2, 2018

A YANG Data Module for Network Virtualization Overlay Resource Management
draft-wu-opsawg-network-overlay-resource-model-00

Abstract

This document defines a YANG data module for Network Virtualization Overlay Resource Management. It is a resource facing model independent of control plane protocols and captures topological and resource related information pertaining to Network Virtualization Overlay.

This module enables clients, which interact with a network orchestrator or controller via a REST interface, for Network Virtualization Overlay topology related operations such as obtaining and allocating the relevant topology resource information.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 3, 2018.

Table of Contents

1. Introduction

[RFC8299] defines customer service model for L3VPN service that can be used to describe a service as offered or delivered to a customer by a network operator. As described in [RFC8309], a customer service model is not resource facing model and does not describes how a network operator realizes and delivers the service described by the module since it is not used to directly configure network devices, protocols, or functions or something sent to network devices (i.e., routers or switches) for processing.

This document defines a YANG module for Network Virtualization Overlay Management. It is a resource facing model independent of control plane protocols and captures topological and resource related information pertaining to Network Virtualization Overlay.

This module enables clients to interact with a network orchestrator or controller via a RESTful interface, for providing connectivity services over a Network Virtualization Overlay topology. In particular, this module supports operations such as exposing abstract service topology, retrieving, and allocating the relevant topology resource information.

As a reminder, and as defined in [RFC7297], the IP connectivity service is the IP transfer capability characterized by a (Source Nets, Destination Nets, Guarantees, Scope) tuple where "Source Nets" is a group of unicast IP addresses, "Destination Nets" is a group of IP unicast and/or multicast addresses, and "Guarantees" reflects the guarantees (expressed in terms of Quality Of Service (QoS), performance, and availability, for example) to properly forward traffic to the said "Destination". Finally, the "Scope" denotes the (network) perimeter (e.g., between Provider Edge (PE) routers or Customer Nodes) where the said guarantees need to be provided. These requirements include: reachability scope (e.g., limited scope, Internet-wide), direction (in/ou), bandwidth requirements, QoS parameters (e.g., one-way delay [RFC7679], loss [RFC7680], or one-way delay variation (jitter) [RFC3393]), protection, and high-availability guidelines (e.g., restoration in less than 50 ms, 100 ms, or 1 second).

The module includes flow identification and classification rules that are required for traffic conformance purposes.

How the data captured using this YANG module is tranlated into network-spefic clauses is out of scope.

2. Conventions used in this document

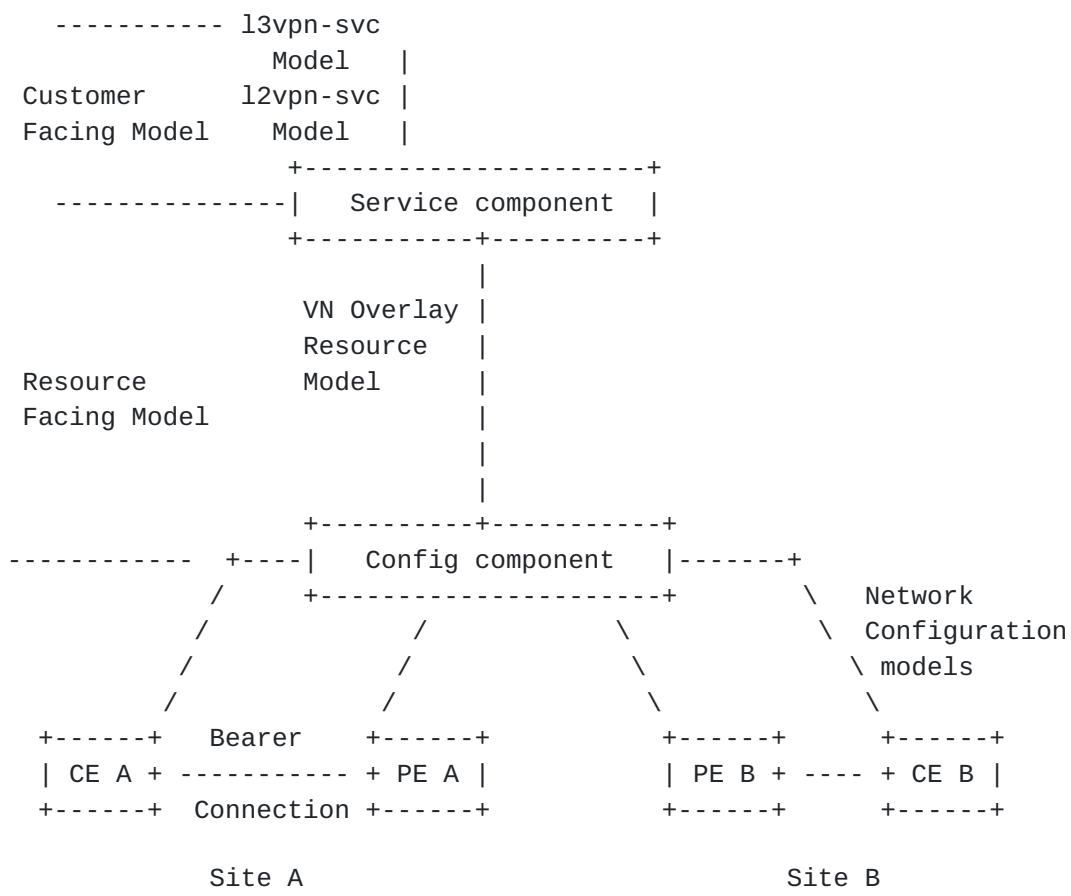
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses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

The following notations are used within the data tree and carry the meaning as below.

Each node is printed as:

```
    <status> <flags> <name> <opts> <type>

    <status> is one of:
         +  for current

    <flags> is one of:

        rw for configuration data
        ro for non-configuration data
        -x for rpcs
        -n for notifications
        -w for writable

    <name> is the name of the node

    If the node is augmented into the tree from another module, its name
    is printed as <prefix>:<name>.

    <opts> is one of:

         ?  for an optional leaf or choice
         !  for a presence container
         *  for a leaf-list or list
         [<keys>] for a list's keys
         (choice)/:(case) Parentheses enclose choice and case nodes,
         and case nodes are also marked with a colon (":")
         <type> is the name of the type for leafs and leaf-lists
```

3. Overview of Network Virtualization Overlay Resource Management Model



L3VPN and L2VPN service models provide an abstracted view of the Layer 3 and Layer 2 VPN service configuration components. Services are built from a combination of network elements and protocols configuration, but are specified for service users in more abstract terms, e.g., these models will specify where to create site and establish site-network-access of a particular site to the provider network (e.g., PE, aggregation switch) and what service requirements of each site-network-access are.

Site location can be determined based on proposed location parameters and constraints in these service models and service requirements of each site-network-access can be determined based on traffic performance metrics (e.g., one-way delay, one-way delay variation, bandwidth) of each PE-CE link connectivity and traffic performance metrics of each service flow or application. The management system will use service models as an input to select appropriate PEs and CEs, allocate interface on the node, generate PE and CE configuration associated with each PE-CE link.

Based on selected PE and CE configuration on each site-network-access of a particular site, the management system can use L3VPN service model and L2VPN service model as inputs and translate it into

resource facing model, i.e., the network virtualization overlay resource model.

This resource facing model can be seen as the projection model of L3VPN service and L2VPN service model and is used to compute path elements and the network access connectivity list when two sites belonging to one VPN spanning across several domains. It also can be combined with other performance measurement or warning models to expose abstract service topology and resource distribution in the network re-optimization cases.

3.1. VN Service Configuration

The YANG module is divided into two main containers: "vn-services" and "sites".

The "vn-service" list under the vn-services container defines global parameters for the VN service for a specific customer. The "vn-id" provided in the vn-service list refers to an internal reference for this VN service, while the customer name refers to a more-explicit reference to the customer. The "vn-type" in the vn-service list refers to a set of basic VPN type. In addition, each "vn-service" also include a list of "site-network-access".

The service requirements on each "site-network-access" or site to site service requirements is specified in details in the service container under "sites/site" or "sites/site/site-network-access".

3.1.1. VN and Network Access Association Configuration

Within a given VN service there can be one or more VN and Network Access Associations(VNAAs). VNAAs are represented as a list and indexed by the vn-id and vn-type.

```
 module: ietf-vn-rsc
+--rw vn-rsc
 +--rw vn-services
 | +--rw vn-service* [vn-id]
 |   +--rw vn-id         svc-id
 |   +--rw vn-type       identityref
 .
 .
 |   +--rw site-network-accesses
 |   +--rw site-network-access* [site-network-access-id]
 |     +--rw site-network-access-id   svc-id
```

Snippet of data hierarchy related to VN and Network Access Associations (VNAA)

3.1.2. Traffic Performance Requirements Configuration

3.1.2.1. Per-Site Network Access Requirements

Per-Site network access traffic performance requirements are represented as a list within the data hierarchy and indexed by the key site-network-access-id.

Traffic Performance requirements include latency, jitter, and bandwidth utilization. Upload bandwidth and download bandwidth are performance parameters associated each domain-network-access.

Latency, jitter, and bandwidth utilization are performance requirements associated with each service flow or application.

```
module: ietf-vn-rsc
   +--rw site-network-accesses
     +--rw site-network-access* [site-network-access-id]
      +--rw site-network-access-id   leafref
      +--rw device-id   leafref
      +--rw access-diversity {site-diversity}?
      | +--rw groups
      | | +--rw group* [group-id]
      | |   +--rw group-id  string
      | +--rw constraints
      |   +--rw constraint* [constraint-type]
      |    +--rw constraint-type  identityref
      |    +--rw target
      |      +--rw (target-flavor)?
      |       +--:(id)
      |       | +--rw group* [group-id]
      |       |    ...
      |       +--:(all-accesses)
      |       | +--rw all-other-accesses?  empty
      |       +--:(all-groups)
      |         +--rw all-other-groups?   empty
      +--rw service
      | +--rw svc-input-bandwidth?  uint32
      | +--rw svc-output-bandwidth?  uint32
      | +--rw svc-mtu?        uint16
      | +--rw qos {qos}?
      | | +--rw qos-classification-policy
      | | | +--rw rule* [id]
      | | |   +--rw id          uint16
      | | |   +--rw (match-type)?
      | | |   | +--:(match-flow)
      | | |   | | +--rw match-flow
      | | |   | |    ...
      | | |   | +--:(match-application)
      | | |   |   +--rw match-application?  identityref
      | | |   +--rw target-class-id?   string
      | | +--rw qos-profile
      | |   +--rw (qos-profile)?
      | |    +--:(standard)
      | |    | +--rw profile?  string
      | |    +--:(custom)
      | |      +--rw classes {qos-custom}?
      | |       +--rw class* [class-id]
```

Snippet of data hierarchy related to Per Site network access QoS requirements

3.1.2.2. Site-to-Site Traffic Performance Requirements

QoS guarantees denote a set of transfer performance metrics that characterize the quality of the transfer treatment to be experienced (when crossing a transport infrastructure) by a flow issued from or forwarded to a (set of) sites.

Suppose one VPN has multiple sites and any two sites span across multiple domains, site-to-site network access QoS requirements can be used to describe QoS requirements across sites.

Site-to-site network access traffic performance requirements are represented as a list within the data hierarchy and indexed by the key 'site-id'. The source site is specified as 'site-id' under site list, the 'target-site' is specified under match-flow case.

Traffic performance requirements include latency, jitter, and bandwidth utilization.

Shaping/policing filters may be applied so as to assess whether traffic is within the capacity profile or out of profile. Out-of-profile traffic may be discarded or assigned another class.

```
module: ietf-vn-rsc
+--rw sites
  +--rw site* [site-id]
   +--rw site-id         svc-id
   +--rw service
   | +--rw qos {qos}?
   | | +--rw qos-classification-policy
   | | | +--rw rule* [id]
   | | |   +--rw id          uint16
   | | |   +--rw (match-type)?
   | | |   | +--:(match-flow)
   | | |   | | +--rw match-flow
   | | |   | |   +--rw target-sites*    svc-id
   | | |   +--rw target-class-id?   string
   | | +--rw qos-profile
   | |   +--rw (qos-profile)?
   | |    +--:(standard)
   | |    | +--rw profile?  string
   | |    +--:(custom)
   | |      +--rw classes {qos-custom}?
   | |       +--rw class* [class-id]
   | |         +--rw class-id   string
   | |         +--rw rate-limit?  uint8
   | |         +--rw latency
   | |         | +--rw (flavor)?
   | |         |    ...
   | |         +--rw jitter
   | |         | +--rw (flavor)?
   | |         |    ...
   | |         +--rw bandwidth
   | |          +--rw guaranteed-bw-percent?  uint8
   | |          +--rw end-to-end?       empty
```

Snippet of data hierarchy related to Site to Site QoS requirements

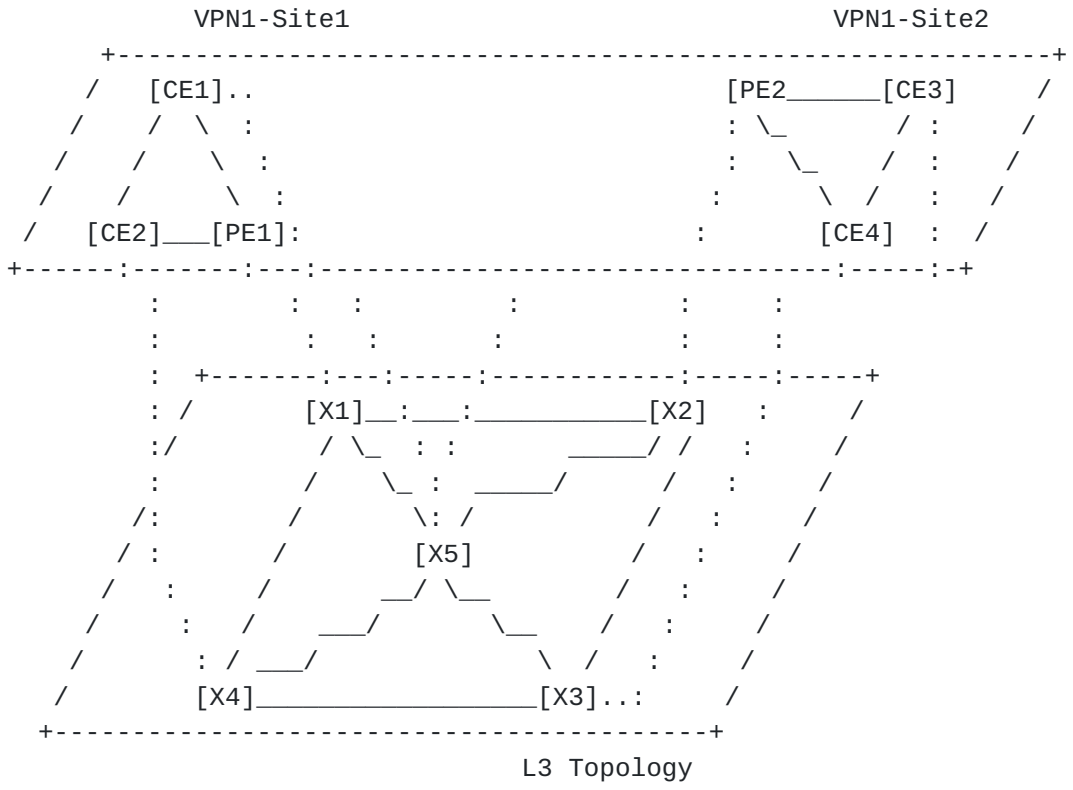
3.2. VN Service Topology Resource Distribution configuration

A 'site' is composed of at least one "site-network-access" and, in the case of multihoming, may have multiple site-network-access points.

For each "site-network-access", the ingress device/customer device and/or egress device has been selected to connect to the provider network, ingress device list is specified under site and egress device is specified under vn-attachment container.

With selected ingress device and egress device and VN membership, VN service topology can be constructed. Resource allocation for Site to

Site connectivity or connectivity within site can be further calculated based on this VN service topology.



L3 Topology

4. RPC Definitions for Computation of TE Path Element List and Network Access Connectivity List

The RPC model facilitates issuing commands to a NETCONF server (in this case to the device that need to execute the path computation API command or path computation algorithm) and obtain a response. RPC model defined here abstracts path computation specific commands in a technology independent manner.

There are two RPC commands defined for the purpose of computation of path element list and network access connectivity list respectively. In this section we present a snippet of the path element list computation command and network access connectivity list computation for illustration purposes. Please refer to Section 3.4 for the complete data hierarchy and Section 4 for the YANG model.

```
  rpcs:
    +---x vn-path-element-compute
    |  +---w input
    |  |  +---w vn-member-list* [vn-member-id]
    |  |     +---w vn-member-id          -> /vn-svc/vn-services/vn-service/vn-
id
    |  |     +---w constraint
    |  |     |  +---w path-element* [path-element-id]
    |  |     |     +---w path-element-id
    |  |     |     +---w address?
    |  |     +---w objective-function?   identityref
    |  |     +---w metric* [metric-type]
    |  |        +---w metric-type     identityref
    |  |        +---w metric-value?   uint32
    |  +--ro output
    |     +--ro vn-member-list* [vn-member-id]
    |        +--ro vn-member-id    -> /vn-svc/vn-services/vn-service/vn-id
    |        +--ro metric* [metric-type]
    |        |  +--ro metric-type     identityref
    |        |  +--ro metric-value?   uint32
    |        +--ro path
    |           +--ro path-element* [path-element-id]
    |              +--ro path-element-id
    +---x vn-network-connectivity-stitch
       +---w input
       |  +---w vn-member-list* [vn-id]
       |     +---w vn-id            -> /vn-svc/vn-services/vn-service/vn-id
       |     +---w source-access* [access-id]
       |     |  +---w access-id
       |     |  +---w destination-access* [access-id]
       |     +---w objective-function?   identityref
       |     +---w metric* [metric-type]
       |        +---w metric-type     identityref
       |        +---w metric-value?   uint32
       +--ro output
          +--ro vn-access-list* [index]
             +--ro index        uint32
             +--ro source-access -> /vn-svc/sites/site/site-network-accesses/
site-network-access/site-network-access-id
             +--ro destination-access-> /vn-svc/sites/site/site-network-
accesses/site-network-access/site-network-access-id
             +--ro multi-domain-network-access-list * [domain-id]
                +--ro domain-id                 svc-id
                +--ro network-access-id         svc-id
```

With these two RPC commands, we can calculate

Path element list that is applied to network access connectivity

within the site, or Site to Site connectivity or end to end connectivity.

Network access connectivity list that is applied to site to site connectivity and end to end connectivity spanning across multiple domains.

5. Data Hierarchy

The figure below describes the overall structure of the YANG module:

```
module: ietf-vn-rsc
    +--rw vn-rsc
       +--rw vn-services
       |  +--rw vn-service* [vn-id]
       |     +--rw vn-id                    svc-id
       |     +--rw customer-name?           string
       |     +--rw service-topology?        identityref
       |     +--rw site-network-accesses
       |        +--rw site-network-access* [site-network-access-id]
       |           +--rw site-network-access-id    svc-id
       +--rw sites
          +--rw site* [site-id]
             +--rw site-id                  svc-id
             +--rw cpe-devices
             |  +--rw cpe-device* [device-id]
             |     +--rw device-id         svc-id
             |     +--rw address-family?   address-family
             |     +--rw address?          inet:ip-address
             |     +--rw interfaces
             |        +--rw interface?        if:interface-ref
             |        +--rw sub-interfaces*   if:interface-ref
             +--rw service
             |  +--rw qos {qos}?
             |     +--rw qos-classification-policy
             |     |  +--rw rule* [id]
             |     |     +--rw id                  string
             |     |     +--rw (match-type)?
             |     |     |  +--:(match-flow)
             |     |     |  |  +--rw match-flow
             |     |     |  |     +--rw dscp?               inet:dscp
             |     |     |  |     +--rw dot1p?              uint8
             |     |     |  |     +--rw ipv4-src-prefix?    inet:ipv4-prefix
             |     |     |  |     +--rw ipv6-src-prefix?    inet:ipv6-prefix
             |     |     |  |     +--rw ipv4-dst-prefix?    inet:ipv4-prefix
             |     |     |  |     +--rw ipv6-dst-prefix?    inet:ipv6-prefix
             |     |     |  |     +--rw l4-src-port?        inet:port-number
             |     |     |  |     +--rw target-sites*       svc-id {target-
sites}?
             |     |     |  |     +--rw l4-src-port-range
             |     |     |  |     |  +--rw lower-port?   inet:port-number
```

```
|     |     |  |     |  +--rw upper-port?   inet:port-number
```

```
             |     |     |  |     +--rw l4-dst-port?         inet:port-number
             |     |     |  |     +--rw l4-dst-port-range
             |     |     |  |     |  +--rw lower-port?   inet:port-number
             |     |     |  |     |  +--rw upper-port?   inet:port-number
             |     |     |  |     +--rw protocol-field?      union
             |     |     |  +--:(match-application)
             |     |     |     +--rw match-application?   identityref
             |     |     +--rw target-class-id?     string
             |     +--rw qos-profile
             |        +--rw (qos-profile)?
             |           +--:(standard)
             |           |  +--rw profile?
                              -> /vn-svc/vpn-profiles/valid-provider-
identifiers/qos-profile-identifier/id
             |           +--:(custom)
             |              +--rw classes {qos-custom}?
             |                 +--rw class* [class-id]
             |                    +--rw class-id      string
             |                    +--rw direction?    identityref
             |                    +--rw rate-limit?   uint8
             |                    +--rw latency
             |                    |  +--rw (flavor)?
             |                    |     +--:(lowest)
             |                    |     |  +--rw use-lowest-latency?   empty
             |                    |     +--:(boundary)
             |                    |        +--rw latency-boundary?     uint16
             |                    +--rw jitter
             |                    |  +--rw (flavor)?
             |                    |     +--:(lowest)
             |                    |     |  +--rw use-lowest-jitter?   empty
             |                    |     +--:(boundary)
             |                    |        +--rw latency-boundary?    uint32
             |                    +--rw bandwidth
             |                       +--rw guaranteed-bw-percent    uint8
             |                       +--rw end-to-end?              empty
            +--rw site-network-accesses
               +--rw site-network-access* [site-network-access-id]
                  +--rw site-network-access-id
                          -> /vn-svc/vn-services/vn-service/site-network-
accesses/site-network-access/site-network-access-id
                  +--rw ingress-device-id?               -> /vn-svc/sites/
site/cpe-devices/cpe-device/device-id
                  +--rw access-diversity {site-diversity}?
                  |  +--rw groups
                  |  |  +--rw group* [group-id]
                  |  |     +--rw group-id    string
                  |  +--rw constraints
                  |     +--rw constraint* [constraint-type]
```

```
|        +--rw constraint-type    identityref
|        +--rw target
|           +--rw (target-flavor)?
```

```
                  |             +--:(id)
                  |             |  +--rw group* [group-id]
                  |             |     +--rw group-id    string
                  |             +--:(all-accesses)
                  |             |  +--rw all-other-accesses?   empty
                  |             +--:(all-groups)
                  |                +--rw all-other-groups?    empty
                  +--rw service
                  |  +--rw svc-input-bandwidth?    uint32
                  |  +--rw svc-output-bandwidth?   uint32
                  |  +--rw svc-mtu?                uint16
                  |  +--rw qos {qos}?
                  |     +--rw qos-classification-policy
                  |     |  +--rw rule* [id]
                  |     |     +--rw id                   string
                  |     |     +--rw (match-type)?
                  |     |     |  +--:(match-flow)
                  |     |     |  |  +--rw match-flow
                  |     |     |  |     +--rw dscp?               inet:dscp
                  |     |     |  |     +--rw dot1p?              uint8
                  |     |     |  |     +--rw ipv4-src-prefix?    inet:ipv4-
prefix
                  |     |     |  |     +--rw ipv6-src-prefix?    inet:ipv6-
prefix
                  |     |     |  |     +--rw ipv4-dst-prefix?    inet:ipv4-
prefix
                  |     |     |  |     +--rw ipv6-dst-prefix?    inet:ipv6-
prefix
                  |     |     |  |     +--rw l4-src-port?        inet:port-
number
                  |     |     |  |     +--rw target-sites*       svc-id
{target-sites}?
                  |     |     |  |     +--rw l4-src-port-range
                  |     |     |  |     |  +--rw lower-port?   inet:port-number
                  |     |     |  |     |  +--rw upper-port?   inet:port-number
                  |     |     |  |     +--rw l4-dst-port?        inet:port-
number
                  |     |     |  |     +--rw l4-dst-port-range
                  |     |     |  |     |  +--rw lower-port?   inet:port-number
                  |     |     |  |     |  +--rw upper-port?   inet:port-number
                  |     |     |  |     +--rw protocol-field?     union
                  |     |     |  +--:(match-application)
                  |     |     |     +--rw match-application?   identityref
                  |     |     +--rw target-class-id?    string
                  |     +--rw qos-profile
                  |        +--rw (qos-profile)?
                  |           +--:(standard)
                  |           |  +--rw profile?
```

```
                                        -> /vn-svc/vpn-profiles/valid-
provider-identifiers/qos-profile-identifier/id
                    |           +--:(custom)
                    |              +--rw classes {qos-custom}?
                    |                 +--rw class* [class-id]
                    |                    +--rw class-id      string
                    |                    +--rw direction?    identityref
                    |                    +--rw rate-limit?   uint8
```

```
                  |                   +--rw latency
                  |                   |  +--rw (flavor)?
                  |                   |     +--:(lowest)
                  |                   |     |  +--rw use-lowest-latency?
empty
                  |                   |     +--:(boundary)
                  |                   |        +--rw latency-boundary?
uint16
                  |                   +--rw jitter
                  |                   |  +--rw (flavor)?
                  |                   |     +--:(lowest)
                  |                   |     |  +--rw use-lowest-jitter?
empty
                  |                   |     +--:(boundary)
                  |                   |        +--rw latency-boundary?
uint32
                  |                   +--rw bandwidth
                  |                      +--rw guaranteed-bw-percent    uint8
                  |                      +--rw end-to-end?              empty
                  +--rw vn-attachments
                     +--rw vn-attachment* [vn-id]
                        +--rw vn-id               svc-id
                        +--rw vn-type?            identityref
                        +--rw attachment-point
                           +--rw egress-device-id?     svc-id
                           +--rw address-family?   address-family
                           +--rw address?          inet:ip-address
                           +--rw interfaces
                              +--rw interface?        if:interface-ref
                              +--rw sub-interfaces*   if:interface-ref

  rpcs:
    +---x vn-path-element-compute
    |  +---w input
    |  |  +---w vn-member-list* [vn-member-id]
    |  |     +---w vn-member-id          -> /vn-svc/vn-services/vn-service/vn-
id
    |  |     +---w src
    |  |     |  +---w src-address?              -> /vn-svc/sites/site/site-id
    |  |     |  +---w site-network-access-id?
                        -> /vn-svc/sites/site/site-network-accesses/site-
network-access/site-network-access-id
    |  |     +---w dst
    |  |     |  +---w dst-address?              -> /vn-svc/sites/site/site-id
    |  |     |  +---w site-network-access-id?
                        -> /vn-svc/sites/site/site-network-accesses/site-
network-access/site-network-access-id
    |  |     +---w constraint
```

```
    |  |     |  +---w path-element* [path-element-id]
    |  |     |     +---w path-element-id    -> /vn-svc/sites/site/site-network-
accesses/site-network-access/vn-attachments/vn-attachment/attachment-point/pe-
device-id
    |  |     |     +---w address?          -> /vn-svc/sites/site/site-network-
accesses/site-network-access/vn-attachments/vn-attachment/attachment-point/
address
    |  |     +---w objective-function?   identityref
    |  |     +---w metric* [metric-type]
    |  |        +---w metric-type     identityref
    |  |        +---w metric-value?   uint32
```

```
    |  +--ro output
    |     +--ro vn-member-list* [vn-member-id]
    |        +--ro vn-member-id    uint32
    |        +--ro src
    |        |  +--ro src-address?              -> /vn-svc/sites/site/site-id
    |        |  +--ro site-network-access-id?   -> /vn-svc/sites/site/site-
network-accesses/site-network-access/site-network-access-id
    |        +--ro dst
    |        |  +--ro dst-address?              -> /vn-svc/sites/site/site-id
    |        |  +--ro site-network-access-id?   -> /vn-svc/sites/site/site-
network-accesses/site-network-access/site-network-access-id
    |        +--ro metric* [metric-type]
    |        |  +--ro metric-type     identityref
    |        |  +--ro metric-value?   uint32
    |        +--ro path
    |           +--ro path-element* [path-element-id]
    |              +--ro path-element-id    -> /vn-svc/sites/site/site-network-
accesses/site-network-access/vn-attachments/vn-attachment/attachment-point/pe-
device-id
    |              +--ro index?             uint32
    |              +--ro address?           -> /vn-svc/sites/site/site-network-
accesses/site-network-access/vn-attachments/vn-attachment/attachment-point/
address
    |              +--ro hop-type?          identityref
    +---x vn-network-connectivity-stitch
       +---w input
       |  +---w vn-list* [vn-id]
       |     +---w vn-id                -> /vn-svc/vn-services/vn-service/vn-
id
       |     +---w source-access* [access-id]
       |     |  +---w access-id             -> /vn-svc/sites/site/site-network-
accesses/site-network-access/site-network-access-id
       |     |  +---w destination-access* [access-id]
       |     |     +---w access-id    -> /vn-svc/sites/site/site-network-
accesses/site-network-access/site-network-access-id
       |     +---w objective-function?   identityref
       |     +---w metric* [metric-type]
       |        +---w metric-type     identityref
       |        +---w metric-value?   uint32
       +--ro output
          +--ro vn-access-list* [index]
             +--ro index        uint32
             +--ro source-access -> /vn-svc/sites/site/site-network-accesses/
site-network-access/site-network-access-id
             +--ro destination-access-> /vn-svc/sites/site/site-network-
accesses/site-network-access/site-network-access-id
             +--ro multi-domain-network-access-list *
                +--ro domain-id                svc-id
```

```
               +--ro network-access-id         svc-id
```

6. Network Virtualization Overlay Management YANG Module

```
<CODE BEGINS> file "ietf-vn-rsc@2018-02-03.yang"
module ietf-vn-rsc {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vn-rsc";
  prefix vnrsc;

  import ietf-inet-types {
```

```
      prefix inet;
    }
    import ietf-l3vpn-svc {
      prefix l3vpn-svc;
    }
    import ietf-interfaces{
      prefix if;
    }

    organization
      "IETF OPSAWG Working Group.";
    contact
      "WG List: foo@ietf.org
      Editor:  Qin Wu <mailto:bill.wu@huawei.com>
      Editor:  Zitao Wang <mailto:wangzitao@huawei.com>";

    description
      "The YANG module defines a generic service configuration
      model for Layer VN services common across all of the
      vendor implementations.";

  revision 2018-02-03{
  description
  "Initial revision";
  reference
  "A YANG Data Model for VN Service Delivery.";
  }
  /* Features */


  /* Typedefs */
  typedef svc-id {
   type string;
   description
   "Type definition for servicer identifier";
  }
   typedef address-family {
    type enumeration {
     enum ipv4 {
      description
       "IPv4 address family.";
     }
     enum ipv6 {
      description
       "IPv6 address family.";
     }
    }
    description
```

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

   /* Identities */
    identity vn-type {
     description
     "Base identity for VN type";
    }
    identity l2vpn {
     base vn-type;
     description
     "Identity for Layer 2 vpn";
    }
    identity l3vpn {
     base vn-type;
     description
     "Identity for Layer 3 vpn";
    }
    identity evpn {
     base l2vpn;
     description
     "Identity for evpn";
    }
    identity vpls {
     base l2vpn;
     description
     "Identity for vpls";
    }
    identity vpw {
     base l2vpn;
     description
     "Identity for vpw";
    }
    identity vpn-topology {
     description
      "Base identity for VPN topology.";
    }
    identity any-to-any {
     base vpn-topology;
     description
      "Identity for any-to-any VPN topology.";
    }
    identity hub-spoke {
     base vpn-topology;
     description

      "Identity for Hub-and-Spoke VPN topology.";
```

```
    }
    identity hub-spoke-disjoint {
     base vpn-topology;
     description
      "Identity for Hub-and-Spoke VPN topology
       where Hubs cannot communicate with each other.";
    }

    identity objective-function{
     description
     "Identity for objective function";
    }

    identity metric-type{
     description
     "Identity for metric type";
    }

    identity hop-type{
     description
     "Identity for hop-type";
    }
    identity loose{
     base hop-type;
     description
     "loose hop in an explicit path";
    }
    identity strict{
     base hop-type;
     description
     "strict hop in an explicit path";
    }
   /* Grouping */
   grouping vn-service-list {
    list vn-service {
     key "vn-id";
     leaf vn-id {
      type svc-id;
      description
      "VN id";
     }
     leaf customer-name {
      type string;
      description
      "Customer name";
     }
     leaf service-topology {
      type identityref {
```

```
       base vpn-topology;
      }
      default any-to-any;
      description
       "VPN service topology.";
     }
     container site-network-accesses{
      list site-network-access{
       key "site-network-access-id";
       leaf site-network-access-id{
        type svc-id;
        description
        "Site network access identifier";
       }
       description
       "List for site-network access";
      }
      description
      "Container for site network accesses";
     }

    description
    "List for vn service";
   }
   description
   "Grouping for vn service list";
  }
  grouping vn-services-grouping{
   container vn-services{
    uses vn-service-list;
    description
    "Container for virtual network service";
   }
   description
   "Grouping for vn services";
  }

  grouping interfaces-grouping{
   container interfaces{
    leaf interface{
     type if:interface-ref;
     description
     "Base interface";
    }
    leaf-list sub-interfaces{
     type if:interface-ref;
     description
     "Sub interfaces";
```

```
     }
     description
     "Container for interfaces";
    }
    description
    "Grouping for interfaces";
   }

   grouping cpe-device-list{
    list cpe-device{
     key "device-id";
     leaf device-id {
      type svc-id;
      description
      "Device identifier";
     }
     leaf address-family{
      type address-family;
      description
       "Address family used for management. If address-family
       is specified, the address may or may not be specified
      (by the customer).";
     }
     leaf address{
      type inet:ip-address;
      description
      "IP address";
     }
     uses interfaces-grouping;
     description
     "List for devices";
    }
    description
    "Grouping for cpe device list";

   }
   grouping cpe-devices-grouping{
    container cpe-devices{
     uses cpe-device-list;
     description
     "Container for cpe devices";
    }
    description
    "grouping for cpe-devices-grouping";
   }

   grouping bandwidth-grouping {
    leaf svc-input-bandwidth{
```

```
     type uint32;
     description
     "Service input bandwidth";
    }
    leaf svc-output-bandwidth{
     type uint32;
     description
     "Service output bandwidth";
    }
    description
    "Grouping for bandwidth";
   }

   grouping attachment-point-grouping{
    container attachment-point{
      leaf pe-device-id {
      type svc-id;
      description
      "PE Device identifier";
     }
     leaf address-family{
      type address-family;
      description
       "Address family used for management. If address-family
       is specified, the address may or may not be specified
      (by the customer).";
     }
     leaf address{
      type inet:ip-address;
      description
      "IP address";
     }
     uses interfaces-grouping;
     description
     "Container for attachment point";
    }
    description
    "Grouping for attachment points";
   }

   grouping vn-attachment-list{
    list vn-attachment{
     key "vn-id";
     leaf vn-id{
      type svc-id;
      description
      "Virtual network identifier";
     }
```

```
     leaf vn-type{
      type identityref{
       base vn-type;
      }
      description
      "VN type";
     }
     uses attachment-point-grouping;
     description
     "List for VN attachments";
    }
    description
    "Grouping for VN attachment list";
   }

   grouping vn-attachments-grouping{
    container vn-attachments{
     uses vn-attachment-list;
     description
     "Container for VN attachments";
    }
    description
    "Grouping for VN attachments";
   }

   grouping site-network-access-list{
    list site-network-access{
     key "site-network-access-id";
     leaf site-network-access-id{
      type leafref{
       path "/vn-svc/vn-services/vn-service"
    +"/site-network-accesses/site-network-access"
    +"/site-network-access-id";
      }
      description
      "Site network access identifier";
     }
     leaf device-id {
      type leafref{
       path "/vn-svc/sites/site/cpe-devices"
    +"/cpe-device/device-id";
      }
      description
      "Device id";
     }
     uses l3vpn-svc:access-diversity;
     container service {
      uses bandwidth-grouping;
```

```
      leaf svc-mtu {
       type uint16;
    description
    "Service-mtu";
      }
      uses l3vpn-svc:site-service-qos-profile;
      description
      "Container for service";
     }
     uses vn-attachments-grouping;
     description
     "List for site-network access";


    }
    description
    "Grouping for site-network access list";
   }

   grouping site-network-accesses-grouping{
    container site-network-accesses{
     uses site-network-access-list;
     description
     "Container for site network accesses";
    }
    description
    "Grouping for site network accesses";
   }

   grouping site-list-grouping{
    list site {
     key "site-id";
     leaf site-id {
      type svc-id;
      description
      "Site identifier";
     }
     uses cpe-devices-grouping;
     container service {
      uses l3vpn-svc:site-service-qos-profile;
      description
      "Site service";
     }
     uses site-network-accesses-grouping;
     description
     "List for sites";
    }
    description
```

```
    "Grouping for site list";
   }

   grouping sites-grouping {
    container sites{
     uses site-list-grouping;
     description
     "Container for sites";
    }
    description
    "Grouping for sites";
   }

   grouping src-grouping{
      container src{
       leaf src-address{
     type leafref {
      path "/vn-svc/sites/site/site-id";
     }
     description
     "Leaf list for source address";
    }
    leaf site-network-access-id{
     type leafref {
      path "/vn-svc/sites/site/site-network-accesses"+
      "/site-network-access/site-network-access-id";
     }
     description
     "Leaf list for site-network-access id";
    }
       description
    "Container for source id";
      }
    description
    "Grouping for source site";
   }

   grouping dst-grouping{
     container dst{
       leaf dst-address{
     type leafref {
      path "/vn-svc/sites/site/site-id";
     }
     description
     "Leaf list for source address";
    }
    leaf site-network-access-id{
     type leafref {
```

```
      path "/vn-svc/sites/site/site-network-accesses"+
      "/site-network-access/site-network-access-id";
     }
     description
     "Leaf list for site-network-access id";
    }
       description
    "Container for destination id";
     }
    description
    "Grouping for source site";
   }

   grouping objective-function-group{
    leaf objective-function {
     type identityref{
      base objective-function;
     }
     description
     "operational state of the objective function";
     }
     description
     "Grouping for objective functions";
   }


   grouping path-element-list{
    list path-element{
     key "path-element-id";
     leaf path-element-id{
      type leafref{
      path "/vn-svc/sites/site/site-network-accesses"+
      "/site-network-access/vn-attachments/vn-attachment"+
      "/attachment-point/pe-device-id";
      }
      description
      "Path element identifier";
     }
     leaf address{
      type leafref{
      path "/vn-svc/sites/site/site-network-accesses"+
      "/site-network-access/vn-attachments/vn-attachment"+
      "/attachment-point/address";
      }
      description
      "Path element address";
     }
     description
```

```
     "List for path elements";
    }
    description
    "Grouping for path elements";
   }

   grouping constraint-grouping{
    container constraint{
     config false;
     uses path-element-list;
     description
     "Container for constraint";
    }
    description
    "Grouping for constraint";
   }

   grouping metric-grouping{
    list metric {
     key metric-type;
     leaf metric-type {
      type identityref{
       base metric-type;
      }
      description
      "Metric type";
     }
     leaf metric-value {
      type uint32;
      description
      "Metric value";
     }
     description
     "List for metric";
    }
    description
    "Grouping for metric";
   }

   grouping path-list{
    list path-element{
     key "path-element-id";
     leaf path-element-id{
      type leafref{
      path "/vn-svc/sites/site/site-network-accesses"+
      "/site-network-access/vn-attachments/vn-attachment"+
      "/attachment-point/pe-device-id";
      }
```

```
      description
      "Path element identifier";
     }
     leaf index{
      type uint32;
      description
      "Index";
     }
     leaf address{
      type leafref{
      path "/vn-svc/sites/site/site-network-accesses"+
      "/site-network-access/vn-attachments/vn-attachment"+
      "/attachment-point/address";
      }
      description
      "Path element address";
     }
     leaf hop-type{
      type identityref {
       base hop-type;
      }
      description
      "Hop type";
     }
     description
     "List for path elements";
    }
    description
    "Grouping for path list";
   }

   grouping path-grouping{
    container path{
    uses path-list;
    description
    "Container for path";
    }
    description
    "Grouping for path";
   }
   grouping access-grouping{
    list source-access{
     key "access-id";
     leaf access-id {
       type leafref{
     path "/vn-svc/sites/site/site-network-accesses"
     +"/site-network-access/site-network-access-id";
    }
```

```
      description
      "Access id";
     }
     list destination-access{
     key "access-id";
     leaf access-id {
       type leafref{
     path "/vn-svc/sites/site/site-network-accesses"
     +"/site-network-access/site-network-access-id";
    }
      description
      "Access id";
     }
     description
     "List for destination access id";
    }
    description
    "List for source access id";
    }
    description
    "Grouping for access";
   }
   /* ....................................*/

   container vn-svc{
    uses vn-services-grouping;
    uses sites-grouping;
    description
    "Container for vn service";
   }

   rpc vn-compute{
    description
    "RPC for VN compute";
    input {
     list vn-member-list {
      key "vn-member-id";
      leaf vn-member-id{
       type leafref{
     path "/vn-svc/vn-services/vn-service/vn-id";
    }
    description
    "VN member identifier";
      }
      uses src-grouping;
      uses dst-grouping;
      uses constraint-grouping;
      uses objective-function-group;
```

```
      uses metric-grouping;
      description
      "List for vn member";
     }

    }
    output{
     list vn-member-list {
      key "vn-member-id";
      leaf vn-member-id{
       type uint32;
    description
    "VN member identifier";
      }
      uses src-grouping;
      uses dst-grouping;
      uses metric-grouping;
      uses path-grouping;
      description
      "List for vn member";
     }
    }
   }

   rpc vn-stitch{
    description
    "RPC for VN compute";
    input {
     list vn-list {
      key "vn-id";
      leaf vn-id{
       type leafref{
     path "/vn-svc/vn-services/vn-service/vn-id";
    }
    description
    "VN identifier";
      }
      uses access-grouping;
      uses objective-function-group;
      uses metric-grouping;
      description
      "List for vn";
     }

    }
    output{
     list vn-access-list {
      key "index";
```

```
      leaf index{
       type uint32;
    description
    "Index for VN access";
      }
      leaf source-access {
       type leafref{
     path "/vn-svc/sites/site/site-network-accesses"
     +"/site-network-access/site-network-access-id";
    }
    description
    "Source Access ID";
      }
      leaf destination-access {
       type leafref{
     path "/vn-svc/sites/site/site-network-accesses"
     +"/site-network-access/site-network-access-id";
    }
    description
    "Destination Access ID";
      }
      list multi-domain-network-access-list {
       key "domain-id network-access-id";
       leaf domain-id {
       type string;
       description
       "Domain ID";
    }
    leaf network-access-id {
     type leafref{
     path "/vn-svc/sites/site/site-network-accesses"
     +"/site-network-access/site-network-access-id";
     }
      description
     "Network access ID";
    }
      description
      "List for multiple domain network access";
      }
      description
      "List for vn access";
     }
    }
   }
   }
   <CODE ENDS>
```

7. 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:

- /vn-svc/vn-services/vn-service

  The entries in this list include the whole vn service configurations to which the customer subscribed, and indirectly create or modify the egress and ingress device configurations. Unexpected changes to these entries could lead to the service disruption and/or network misbehavior.

- /vn-svc/sites/site

  The entries in this list include the customer site configurations. Unexpected changes to these entries could lead to the service disruption and/or network misbehavior.

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

- /vn-svc/vn-services/vn-service

- /vn-svc/sites/site

The entries in these lists include customer-proprietary or confidential information, e.g., customer-name, site location, what service the customer subscribes.

8. 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-vn-rsc

   Registrant Contact: The IESG.

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

This document registers a YANG module in the YANG Module Names registry [RFC7950].

```
---------------------------------------------------------------------
   Name:         ietf-vn-rsc
   Namespace:    urn:ietf:params:xml:ns:yang:ietf-vn-rsc
   Prefix:       vnrsc
   Reference:    RFC xxxx
---------------------------------------------------------------------
```

9. References

9.1. 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>.

[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>.

[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>.

[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>.

9.2. Informative 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>.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <https://www.rfc-editor.org/info/rfc7297>.

[RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <https://www.rfc-editor.org/info/rfc7679>.

[RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <https://www.rfc-editor.org/info/rfc7680>.

Authors' Addresses

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

Email: bill.wu@huawei.com

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

Email: wangzitao@huawei.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com