NVO3 Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

R. Chen
ZTE Corporation
K. Zhao
Huawei Technologies
F. Qin
China Mobile
R. Rahman
Juniper networks
B. Liu
Huawei Technologies

20 October 2025

Base YANG Data Model for NVO3 Protocols
draft-ietf-nvo3-yang-cfg-06

Abstract

This document describes the base YANG data model that can be used by operators to configure and manage Network Virtualization Overlay protocols. The model is focused on the common configuration requirement of various encapsulation options, such as VXLAN, NVGRE, GENEVE and VXLAN-GPE. Using this model as a starting point, incremental work can be done to satisfy the requirement of a specific encapsulation. The model is based on YANG 1.1, which is defined in RFC 7950 and conforms to the Network Management Datastore Architecture (NMDA).

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 23 April 2026.

Table of Contents

1. Introduction

Network Virtualization Overlays (NVO3), such as VXLAN [RFC7348], NVGRE [RFC7637], GENEVE [RFC8926] and VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe], enable network virtualization for data center networks environment that assumes an IP-based underlay.

YANG [RFC7950] is a data modeling language that was introduced to model the configuration and operational state of a device managed using network management protocols such as the Network Configuration Protocol (NETCONF) [RFC6241] or RESTCONF [RFC8040]. YANG is now also being used as a component of other management interfaces, such as command-line interfaces (CLIs).

This document specifies a YANG data model that can be used to configure and manage NVO3 protocols. The model covers the configuration of NVO3 instances as well as their operation states, which are the basic common requirements of the different tunnel encapsulations. Thus it is called "the base model for NVO3" in this document. The model is based on YANG 1.1, which is defined in RFC 7950 and conforms to the Network Management Datastore Architecture (NMDA) as described in [RFC8342].

1.1. Requirements Language

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.

1.2. Terminology

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

The following abbreviations are used in this document and the defined model:

>BUM: Broadcast, Unknown Unicast, Multicast traffic [RFC9469]

Geneve: Generic Network Virtualization Encapsulation. An NVO3 encapsulation defined in [RFC8926]

NVO3: Network Virtualization Overlays Layer 3 [RFC8014]

NVE: Network Virtualization Edge [RFC7365]

VAP: Virtual Access Points [RFC7365]

VNI: Virtual Network Instance[RFC7365]

1.3. Tree Diagrams

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

1.4. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as the context clearly indicates the YANG module in which each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG Module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt	ietf-routing	[RFC8349]
if	ietf-interfaces	[RFC8343]
ni	ietf-network-instance	[RFC8529]
l2vpn	ietf-l2vpn	[I-D.ietf-bess-l2vpn-yang]
l3vpn	ietf-l3vpn	[I-D.ietf-bess-l3vpn-yang]

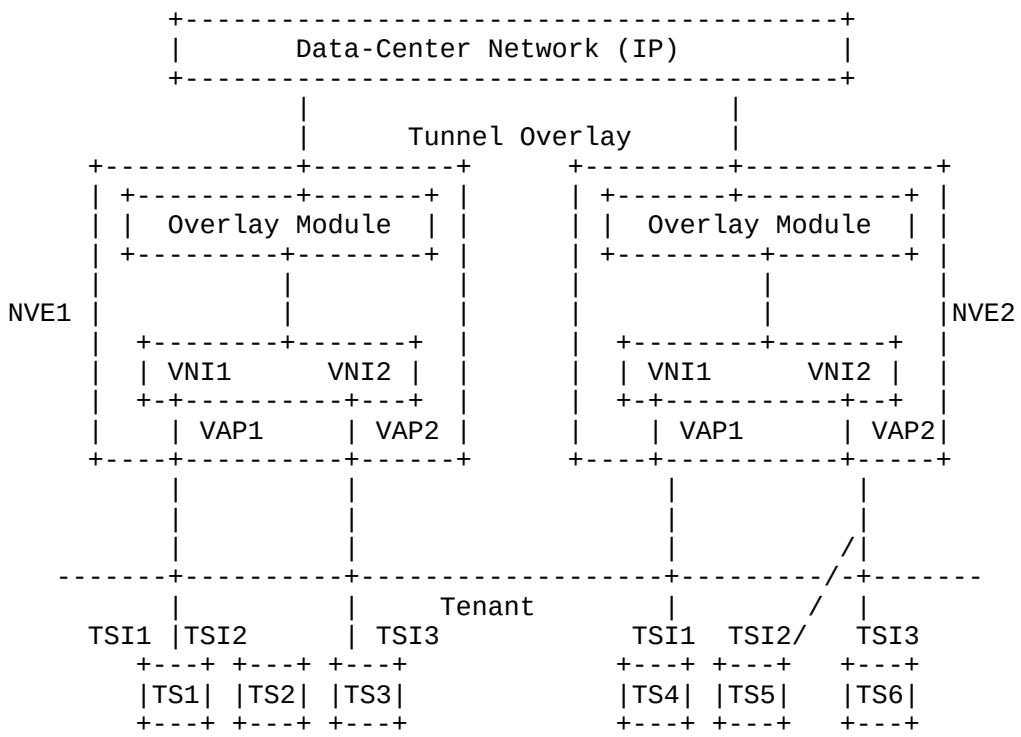
Table 1: Prefixes and Corresponding YANG Modules

2. The YANG Data Model for NVO3

The NVO3 base YANG model defined in this document is used to configure the NVEs. It is divided into three containers. The first container contains the configuration of the virtual network instances, e.g. the VNI, the NVE that the instance is mounted, the peer NVEs which can be determined dynamically via a control plane or given statically, and the statistical states of the instance. The other two containers are separately the statistical states of the peer NVEs and the tunnels.

2.1. Mapping to the NVO3 architecture

The NVO3 base YANG model is defined according to the NVO3 architecture [RFC8014]. As shown in Figure 1, the reference model of the NVE defined in [RFC8014], multiple instances can be mounted under a NVE. The key of the instance is VNI. The source NVE of the instance is the NVE configured by the base YANG. An instance can have several peer NVEs. A NVO3 tunnel can be determined by the VNI, the source NVE and the peer NVE. The tunnel can be built statically by manually indicate the addresses of the peer NVEs, or dynamically via a control plane, e.g. EVPN [RFC8365]. An enabler is defined in this document to choose from these two modes.



2.2. Configuration and Operational State

2.2.1. Configuration

2.2.1.1. NVE as an interface

A NVE in the NVO3 base YANG is defined via augmenting the IETF interface YANG. If anycast gateway is enabled, the source VTEP address is the address of the anycast gateway, and a bypass address is used to uniquely identify the NVE. Otherwise, the source VTEP address is the NVE interface's own IP address.

2.2.1.2. Virtual Network Instance

A Virtual Network Instance ('VNI') is a specific VN instance on an NVE [RFC7365]. At each NVE, a Tenant System is connect to VNIs through Virtual Access Points (VAP). VAPs can be physical ports or virtual ports identified by the bridge domain Identifier ('bdId'). The mapping between VNI and bdId is managed by the operator.

As defined in [RFC9135], a tenant can have multiple bridge domains, and each domain has its own VNI.

Thus these VNIs are used as L2VPN. Besides, a dedicated VNI can be used for routing between the bridge domains, i.e. used as L3VPN. The mapping relationship between VNI and L2VPN (respectively, L3VPN) is given by augmenting the IETF YANG of L2VPN (respectively L3VPN).

2.2.1.3. BUM Mode

An NVE SHOULD support either ingress replication, or multicast proxy, or point to multipoint tunnels on a per-VNI basis. It is possible that both modes be used simultaneously in one NVO3 network by different NVEs.

If ingress replication is used, the receiver addresses are listed in 'peers'. If multicast proxy [RFC8293] is used, the proxy's address is given in "flood-proxy". If the choice is point to multipoint tunnels, the multicast address is given as 'multiAddr'.

2.2.2. Statistics

Operators can determine whether a NVE should gather statistic values on a per-VNI basis. An enabler is contained in the 'static' list as 'statistic-enable' leaf. If the gathering for a VNI is enabled, the statistical information about the local NVEs, the remote NVEs, the flows and the MAC addresses will be collected by the NVEs in this VNI.

2.3. Module Structure

The ietf-nvo3-base YANG module augments the routing container in the ietf-routing model [RFC8349] with a NVO3 container and defines base NVO3 YANG data model and operational state. This module is augmented by modules supporting different data planes.

```
module: ietf-nvo3-base
  +--rw nvo3
  |  +--rw vni-instances
  |     +--rw vni-instance*[vni-id]
  |        +--rw vni-id                uint32
  |        +--rw vni-mode?             vni-mode
  |        +--rw source-nve            if:interface-ref
  |        +--rw protocol-bgp?         boolean
  |        +--ro status?               vni-status-type
  |        +--rw static-ipv4-peers
  |        |  +--rw static-peer*[peer-ip]
  |        |     +--rw peer-ip       inet:ipv4-address-no-zone
  |        |     +--rw out-vni-id?   uint32
  |        +--rw static-ipv6-peers
  |        |  +--rw static-ipv6-peer*[peer-ip]
  |        |     +--rw peer-ip       inet:ipv6-address-no-zone
  |        |     +--rw out-vni-id?   uint32
  |        +--rw flood-proxys
  |        |  +--rw flood-proxy*[peer-ip]
  |        |     +--rw peer-ip    inet:ip-address-no-zone
  |        +--rw mcast-groups
  |        |  +--rw mcast-group*[mcast-ip]
  |        |     +--rw mcast-ip    inet:ip-address-no-zone
  |        +--rw statistic
  |           +--rw enable?   boolean
  |           +--ro info
  |              +--ro send-bits-rate?           uint64
  |              +--ro send-pkts-rate?           uint64
  |              +--ro send-unicast-pkts?        uint64
  |              +--ro send-multicast-pkts?      uint64
  |              +--ro send-broadcast-pkts?      uint64
  |              +--ro send-total-bytes?         uint64
  |              +--ro send-total-pkts?          uint64
  |              +--ro receive-bits-rate?        uint64
  |              +--ro receive-pkts-rate?        uint64
  |              +--ro receive-unicast-pkts?     uint64
  |              +--ro receive-multicast-pkts?   uint64
  |              +--ro receive-broadcast-pkts?   uint64
  |              +--ro receive-total-bytes?      uint64
  |              +--ro receive-total-pkts?       uint64
  |              +--ro drop-unicast-pkts?        uint64
```

```
     |              +--ro drop-multicast-pkts?      uint64
     |              +--ro drop-broadcast-pkts?      uint64
     +--ro vni-peer-infos
     |  +--ro peers
     |     +--ro peer*[vni-id source-ip peer-ip]
     |        +--ro vni-id        uint32
     |        +--ro source-ip     inet:ip-address-no-zone
     |        +--ro peer-ip       inet:ip-address-no-zone
     |        +--ro type?         tunnel-type
     |        +--ro out-vni-id?   uint32
     +--ro tunnel-infos
        +--ro tunnel-info*[tunnel-id]
           +--ro tunnel-id    uint32
           +--ro source-ip?   inet:ip-address-no-zone
           +--ro peer-ip?     inet:ip-address-no-zone
           +--ro status?      tunnel-status
           +--ro type?        tunnel-type
           +--ro up-time?     string
           +--ro vrf-name? ->/ni:network-instances/network-instance/name

   augment /if:interfaces/if:interface:
      +--rw nvo3-nve
         +--rw nve-ip?            inet:ipv4-address-no-zone
         +--rw nve-ipv6?          inet:ipv6-address-no-zone
         +--rw bypass-nve-ip?     inet:ipv4-address-no-zone
         +--rw bypass-nve-ipv6?   inet:ipv6-address-no-zone
         +--rw statistics
         +--rw statistic* [vni-id peer-ip direction]
            +--rw vni-id       uint32
            +--rw peer-ip      inet:ip-address-no-zone
            +--rw direction    direction-type
            +--ro info
               +--ro send-bits-rate?           uint64
               +--ro send-pkts-rate?           uint64
               +--ro send-unicast-pkts?        uint64
               +--ro send-multicast-pkts?      uint64
               +--ro send-broadcast-pkts?      uint64
               +--ro send-total-bytes?         uint64
               +--ro send-total-pkts?          uint64
               +--ro receive-bits-rate?        uint64
               +--ro receive-pkts-rate?        uint64
               +--ro receive-unicast-pkts?     uint64
               +--ro receive-multicast-pkts?   uint64
               +--ro receive-broadcast-pkts?   uint64
               +--ro receive-total-bytes?      uint64
               +--ro receive-total-pkts?       uint64
               +--ro drop-unicast-pkts?        uint64
               +--ro drop-multicast-pkts?      uint64
```

```
            +--ro drop-broadcast-pkts?      uint64

     augment /ni:network-instances/ni:network-instance
     /ni:ni-type/l3vpn:l3vpn/l3vpn:l3vpn:
       +--rw vnis
          +--rw vni*[vni-id]
             +--rw vni-id    uint32

     augment /ni:network-instances/ni:network-instance
     /ni:ni-type/l2vpn:l2vpn:
       +--rw vnis
          +--rw vni*[vni-id]
             +--rw vni-id                uint32
             +--rw split-horizon-mode?   vni-bind-type
                +--rw split-group?       string

     rpcs:
       +---x reset-vni-instance-statistic
       |  +---w input
       |     +---w vni-id    uint32
       +---x reset-vni-peer-statistic
          +---w input
             +---w vni-id       uint32
             +---w peer-ip      inet:ip-address-no-zone
             +---w direction    direction-type
```

2.4. YANG Module

```
   <CODE BEGINS> file "ietf-nvo3-base@2023-10-16.yang"
   module ietf-nvo3-base {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-nvo3-base";
     prefix "nvo3";

     import ietf-network-instance {
       prefix "ni";
     }

     import ietf-interfaces {
       prefix "if";
     }

     import ietf-inet-types {
       prefix "inet";
     }

     import ietf-l2vpn {
       prefix "l2vpn";
```

```
    }

    import ietf-bgp-l3vpn {
      prefix "l3vpn";
    }

    import iana-if-type {
      prefix ianaift;
    }

    organization
      "IETF NVO3(Network Virtualization Overlays) Working Group";
    contact
      "WG Web:   &lt;https://datatracker.ietf.org/wg/nvo3&gt;
       WG List:  &lt;mailto:nvo3@ietf.org&gt;

       WG Chair: Matthew Bocci
                 &lt;mailto:matthew.bocci@nokia.com&gt;

       WG Chair: Sam Aldrin
                 &lt;mailto:aldrin.ietf@gmail.com&gt;


       Editor:  Ran Chen
                &lt;mailto:chen.ran@zte.com.cn&gt;
       Editor:  Kunding Zhao
                &lt;mailto:zhaokun.ding@huawei.com&gt;
       Editor:  Fengwei Qin
                &lt;mailto:qinfengwei@chinamobile.com&gt;
       Editor:  Reshad Rahman
                &lt;mailto:reshad@yahoo.com&gt;
       Editor:  Bing Liu
                &lt;mailto:remy.liubing@huawei.com&gt;
      ";
    description
          "This model is focused on the common configuration
          requirement of various encapsulation options,
          such as VXLAN, NVGRE, GENEVE and VXLAN-GPE.

          Copyright (c) 2025 IETF Trust and the persons identified as
          authors of the code.  All rights reserved.

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

```
          This version of this YANG module is part of RFC 9719
          (https://www.rfc-editor.org/info/rfc9719); see the RFC
                  itself for full legal notices.

          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 (RFC 2119)
                  (RFC 8174) when, and only when, they appear in all capitals,
                  as shown here.";

      revision 2025-10-16 {
        description
          "initial version.";
        reference
          "RFC XXXX: Base YANG Data Model for NVO3 Protocols";
      }
    typedef vni-status-type {
      type enumeration {
        enum "up" {
          description
            "The state is up.";
        }
        enum "down" {
          description
            "The state is down.";
        }
      }
      description
        "The state for VNI.";
    }

    typedef tunnel-status {
      type enumeration {
        enum "up" {
          description
            "The tunnel is up.";
  }
        enum "down" {
          description
            "The tunnel is down.";
        }
      }
      description
        "The status of NVO3 Tunnel.";
    }
    typedef tunnel-type {
      type enumeration {
```

```
          enum "dynamic" {
            description
              "The tunnel is dynamic.";
          }
          enum "static" {
            description
              "The tunnel is static.";
          }
          enum "invalid" {
            description
              "The tunnel is invalid.";
          }
        }
        description
          "The type of NVO3 Tunnel.";
      }

      typedef direction-type {
        type enumeration {
          enum "inbound" {
            description
              "Inbound.";
          }
          enum "outbound" {
            description
              "Outbound.";
          }
          enum "bidirection" {
            description
              "Bidirection.";
          }
        }
        description
          "Bound direction.";
      }
      typedef vni-bind-type {
        type enumeration {
          enum "hub-mode" {
            description
              "Hub mode. The vni instance can't communicate
                      with other hub mode vni instances.";
          }
          enum "spoke-mode" {
            description
              "Spoke mode.";
          }
          enum "split-group-mode" {
            description
```

```
            "Split group mode.";
        }
      }
      description
        "The binding type of VNI.";
    }

    typedef vni-mode {
      type enumeration {
        enum "local" {
          description
            "Local mode.";
        }
        enum "global" {
          description
            "Global mode.";
        }
      }
      description
        "The mode of VNI.";
    }

    grouping nvo3-traffic-statistics {
      description
        "NVO3 tunnel traffic statistics collection.";
      leaf send-bits-rate {
        type uint64;
        units bit/s;
        description
          "Number of send bits per second.";
      }
      leaf send-pkts-rate {
        type uint64;
        units pps;
        description
          "Number of send packets per second.";
      }
      leaf send-unicast-pkts {
        type uint64;
        units packet;
        description
          "Number of send unicast packets.";
      }
      leaf send-multicast-pkts {
        type uint64;
        units packet;
        description
          "Number of send multicast packets.";
```

```
      }
      leaf send-broadcast-pkts {
        type uint64;
        units packet;
        description
          "Number of send broadcast packets.";
      }
      leaf send-total-bytes {
        type uint64;
        units Byte;
        description
          "Total number of send bytes.";
      }
      leaf send-total-pkts {
        type uint64;
        units packet;
        description
          "Total number of send packets.";
      }
      leaf receive-bits-rate {
        type uint64;
        units bit/s;
        description
          "Number of receive bits per second.";
      }
      leaf receive-pkts-rate {
        type uint64;
        units pps;
        description
          "Number of receive packets per second.";
      }
      leaf receive-unicast-pkts {
        type uint64;
        units packet;
        description
          "Number of receive unicast packets.";
      }
      leaf receive-multicast-pkts {
        type uint64;
        units packet;
        description
          "Number of receive multicast packets.";
      }
      leaf receive-broadcast-pkts {
        type uint64;
        units packet;
        description
          "Number of receive broadcast packets.";
```

```
      }
      leaf receive-total-bytes {
        type uint64;
        units Byte;
        description
          "Total number of receive bytes.";
      }
      leaf receive-total-pkts {
        type uint64;
        units packet;
        description
          "Total number of receive packets.";
      }
      leaf drop-unicast-pkts {
        type uint64;
        units packet;
        description
          "Number of discarded unicast packets.";
      }
      leaf drop-multicast-pkts {
        type uint64;
        units packet;
        description
          "Number of discarded multicast packets.";
      }
      leaf drop-broadcast-pkts {
        type uint64;
        units packet;
        description
          "Number of discarded broadcast packets.";
      }
    }

    container nvo3 {
      description
        "Management of NVO3.";
      container vni-instances {
        description
          "List of virtual network instances.";
        list vni-instance {
          key "vni-id";
          description
            "Configure the information of VNI.";
          leaf vni-id {
            type uint32 {
              range "1..16777215";
            }
            description
```

```
            "The id of VNI.";
          }
          leaf vni-mode {
            type vni-mode;
            default "local";
            description
              "The mode of VNI.";
          }
          leaf source-nve {
            type if:interface-ref;
            must "(/if:interfaces/if:interface
                    [if:name=current()]/if:type='Nve')";
            mandatory true;
            description
              "The name of the local NVE.";
          }
          leaf protocol-bgp {
            type boolean;
            default "false";
            description
              "Learn remote NVEs in the same VNI via BGP.";
          }
          leaf status {
            type vni-status-type;
            config false;
            description
              "The status of the VNI.";
          }
          container static-ipv4-peers {
            description
              "List of remote NVE address created by users in a VNI.";
            list static-peer {
              key "peer-ip";
              description
                "Configure remote NVE address in a same VNI.";
              leaf peer-ip {
                type inet:ipv4-address-no-zone;
                description
                  "The address of the remote NVE.";
              }
              leaf out-vni-id {
                type uint32 {
                  range "1..16777215";
                }
                description
                  "The ID of VNI for outbound.
                                  Do not support separate deletion.";
              }
```

```
        }
      }
      container static-ipv6-peers {
        description
          "List of remote NVE IPv6 address created by users
                      in a VNI.";
        list static-ipv6-peer {
          key "peer-ip";
          description
            "Configure remote NVE IPv6 address in a same VNI.";
          leaf peer-ip {
            type inet:ipv6-address-no-zone;
            description
              "The IPv6 address of the remote NVE.";
          }
          leaf out-vni-id {
            type uint32 {
              range "1..16777215";
            }
            description
              "The ID of VNI for outbound. Do not support separate
                             deletion.";
          }
        }
      }
      container flood-proxys {
        description
          "List of flood proxys for the VNI.";
        list flood-proxy {
          key "peer-ip";
          description
            "Configure flood proxys for the VNI.";
          leaf peer-ip {
            type inet:ip-address-no-zone;
            description
              "The address of flood proxy.";
          }
        }
      }
      container mcast-groups {
        description
          "List of multicast address for the VNI.";
        list mcast-group {
          key "mcast-ip";
          description
            "Configure multicast address in a same VNI.";
          leaf mcast-ip {
            type inet:ip-address-no-zone;
```

```
                description
                  "The mcast address of NVO3.";
              }
            }
          }
          container statistic {
            description
              "Configure VNI traffic statistics.";
            leaf enable {
              type boolean;
              default "false";
              description
                "Enable/disable VNI traffic statistics.";
            }
            container info {
              when "../enable='true'";
              config false;
              description
                "The information of vni instance traffic statistics.";
              uses nvo3-traffic-statistics;
            }
          }
        }
      }
    }
    container vni-peer-infos {
        config false;
        description
          "List of remote NVE addresses.";
        container peers {
          config false;
          description
            "Operational data of remote NVE address in a VNI.";
          list peer {
            key "vni-id source-ip peer-ip";
            config false;
            description
              "Operational data of remote NVE addresses in a VNI.";
            leaf vni-id {
              type uint32 {
                range "1..16777215";
              }
              config false;
              description
                "The ID of VNI.";
            }
            leaf source-ip {
              type inet:ip-address-no-zone;
```

```
            config false;
            description
              "Local NVE address, as NVO3 tunnel source point.";
          }
          leaf peer-ip {
            type inet:ip-address-no-zone;
            config false;
            description
              "Remote NVE address, as NVO3 tunnel end point.";
          }
          leaf type {
            type tunnel-type;
            config false;
            description
              "Tunnel type.";
          }
          leaf out-vni-id {
            type uint32 {
              range "1..16777215";
            }
            config false;
            description
              "The ID of VNI for outbound.";
          }
        }
      }
    }

  container tunnel-infos {
    config false;
    description
      "List of NVO3 tunnel information.";
    list tunnel-info {
      key "tunnel-id";
      config false;
      description
        "Operational data of NVO3 tunnel information.";
      leaf tunnel-id {
        type uint32 {
          range "1..4294967295";
        }
        config false;
        description
          "The ID of NVO3 tunnel.";
      }
      leaf source-ip {
        type inet:ip-address-no-zone;
        config false;
```

```
          description
            "Local NVE address, as NVO3 tunnel source point.";
        }
        leaf peer-ip {
          type inet:ip-address-no-zone;
          config false;
          description
            "Remote NVE address, as NVO3 tunnel end point.";
        }
        leaf status {
          type tunnel-status;
          config false;
          description
            "Tunnel status.";
        }
        leaf type {
          type tunnel-type;
          config false;
          description
            "Tunnel type.";
        }
        leaf up-time {
          type string {
            length "1..10";
          }
          config false;
          description
            "The continuous time as NVO3 tunnel is reachable.";
        }
        leaf vrf-name {
          type leafref {
            path "/ni:network-instances/ni:network-instance/ni:name";
          }
          default "_public_";
          config false;
          description
            "The name of VPN instance.";
        }
      }
    }

    identity Nve {
      base ianaift:iana-interface-type;
      description "A new interface type to be registered to IANA";
    }

    augment "/if:interfaces/if:interface" {
      when "(/if:interfaces/if:interface/if:type = 'nvo3:Nve')";
```

```
      description
        "Augment the interface, NVE as an interface.";
      container nvo3-nve {
        description
          "Local NVE.";
        leaf nve-ip {
          type inet:ipv4-address-no-zone;
          description
            "The address of local NVE.";
        }
        leaf nve-ipv6 {
          type inet:ipv6-address-no-zone;
          description
            "The IPv6 address of the local NVE.";
        }
        leaf bypass-nve-ip {
          type inet:ipv4-address-no-zone;
          description
            "The address of local NVE as bypass.";
        }
        leaf bypass-nve-ipv6 {
          type inet:ipv6-address-no-zone;
          description
            "The IPv6 address of local NVE as bypass.";
        }
        container statistics {
          description
            "List of NVO3 tunnel statistics.";
          list statistic {
            key "vni-id peer-ip direction";
            description
              "Configure NVO3 tunnel statistics information.";
            leaf vni-id {
              type uint32 {
                range "1..16777215";
              }
              description
                "The ID of the VNI.";
            }
            leaf peer-ip {
              type inet:ip-address-no-zone;
              description
                "The address of remote NVE.";
            }
            leaf direction {
              type direction-type;
              description
                "Traffic statistics direction for the tunnel.";
```

```
            }
            container info {
              config false;
              description
                "The information of tunnel traffic statistics.";
              uses nvo3-traffic-statistics;
            }
          }
        }
      }
    }

    augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
            "/l3vpn:l3vpn/l3vpn:l3vpn" {
      description "Augment for l3vpn instance";
      container vnis {
        description "Vni list for l3vpn.";
        list vni {
          key "vni-id";
          description
            "Vni for current l3vpn instance.";
          leaf vni-id {
            type uint32 {
              range "1..16777215";
            }
            description
              "The ID of the VNI.";
          }
        }
      }
    }

    augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
            "/l2vpn:l2vpn" {
      description "Augment for l2vpn instance.";
      container vnis {
        description "Vni list for l2vpn.";
        list vni {
          key "vni-id";
          description
            "Vni for current l2vpn instance.";
          leaf vni-id {
            type uint32 {
              range "1..16777215";
            }
            description
              "The ID of the VNI.";
          }
```

```
          container split-horizon {
            description "Configure NVO3 split-horizon information.";
            leaf split-horizon-mode {
              type vni-bind-type;
              default "hub-mode";
              description
                "Split horizon mode.";
            }
            leaf split-group {
              when "(../split-horizon-mode='split-group-mode')";
              type string {
                length "1..31";
              }
              description
                "Split group name.";
            }
          }
        }
      }

      rpc reset-vni-instance-statistic {
        description
          "Clear traffic statistics about the VNI.";
        input {
          leaf vni-id {
            type uint32 {
              range "1..16777215";
            }
            mandatory true;
            description
              "The ID of the VNI.";
          }
        }
      }
      rpc reset-vni-peer-statistic {
        description
          "Clear traffic statistics about the VXLAN tunnel.";
        input {
          leaf vni-id {
            type uint32 {
              range "1..16777215";
            }
            mandatory true;
            description
              "The ID of the VNI.";
          }
          leaf peer-ip {
```

```
            type inet:ip-address-no-zone;
            mandatory true;
            description
              "The address of the remote NVE.";
          }
          leaf direction{
            type direction-type;
            mandatory true;
            description
              "Traffic statistics direction for the tunnel.";
          }
        }
      }
    }
      <code end>
    <CODE ENDS>
```

3. Security Considerations

This document raises no new security issues.

4. 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-nvo3-base
    Registrant Contact: The IESG.
    XML: N/A, the requested URI is an XML namespace.
```

This document requests to register a YANG module in the YANG Module Names registry [RFC7950].

```
    Name: ietf-nvo3-base
    Namespace: urn:ietf:params:xml:ns:yang:ietf-nvo3-base
    Maintained by IANA: N
    Prefix: nvo3
    Reference: RFC XXXX
```

5. Acknowledgements

The authors would like to thank Haibo Wang, Yuan Gao, Guannan Shi, Gang Yan, Mingui Zhang, Yubao Wang, Ruixue Wang and Sijun Weng many others for their helpful comments and suggestions.

6. References

6.1. Normative References

[I-D.ietf-bess-l2vpn-yang]
Shah, H. C., Brissette, P., Chen, I., Hussain, I., Wen, B., and K. Tiruveedhula, "YANG Data Model for MPLS-based L2VPN", Work in Progress, Internet-Draft, draft-ietf-bess-l2vpn-yang-10, 2 July 2019, <https://datatracker.ietf.org/doc/html/draft-ietf-bess-l2vpn-yang-10>.

[I-D.ietf-bess-l3vpn-yang]
Jain, D., Patel, K., Brissette, P., Li, Z., Zhuang, S., Liu, X., Haas, J., Esale, S., and B. Wen, "Yang Data Model for BGP/MPLS L3 VPNs", Work in Progress, Internet-Draft, draft-ietf-bess-l3vpn-yang-05, 13 April 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-bess-l3vpn-yang-05>.

[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>.

[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>.

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

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

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

[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>.

[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>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8365] Sajassi, A., Ed., Drake, J., Ed., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8365, DOI 10.17487/RFC8365, March 2018, <https://www.rfc-editor.org/info/rfc8365>.

[RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <https://www.rfc-editor.org/info/rfc8529>.

[RFC9135] Sajassi, A., Salam, S., Thoria, S., Drake, J., and J. Rabadan, "Integrated Routing and Bridging in Ethernet VPN (EVPN)", RFC 9135, DOI 10.17487/RFC9135, October 2021, <https://www.rfc-editor.org/info/rfc9135>.

6.2. Informative References

[I-D.ietf-nvo3-vxlan-gpe] Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN (VXLAN-GPE)", Work in Progress, Internet-Draft, draft-ietf-nvo3-vxlan-gpe-13, 4 November 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-nvo3-vxlan-gpe-13>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <https://www.rfc-editor.org/info/rfc7365>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <https://www.rfc-editor.org/info/rfc7637>.

[RFC8014] Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <https://www.rfc-editor.org/info/rfc8014>.

[RFC8293] Ghanwani, A., Dunbar, L., McBride, M., Bannai, V., and R. Krishnan, "A Framework for Multicast in Network Virtualization over Layer 3", RFC 8293, DOI 10.17487/RFC8293, January 2018, <https://www.rfc-editor.org/info/rfc8293>.

[RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <https://www.rfc-editor.org/info/rfc8926>.

[RFC9469] Rabadan, J., Ed., Bocci, M., Boutros, S., and A. Sajassi, "Applicability of Ethernet Virtual Private Network (EVPN) to Network Virtualization over Layer 3 (NVO3) Networks", RFC 9469, DOI 10.17487/RFC9469, September 2023, <https://www.rfc-editor.org/info/rfc9469>.

Authors' Addresses

Ran Chen
ZTE Corporation
Nanjing
China
Email: chen.ran@zte.com.cn

Kunding Zhao
Huawei Technologies
Beijing
China
Email: zhaokun.ding@huawei.com

Fengwei Qin
China Mobile
Beijing
China
Email: qinfengwei@chinamobile.com

Reshad Rahman
Juniper networks
Email: reshad@yahoo.com

Bing Liu
Huawei Technologies
Email: remy.liubing@huawei.com