                   YANG Data Model for Network Instances

Abstract

   This document defines a network instance module. This module can be
   used to manage the virtual resource partitioning that may be present
   on a network device. Examples of common industry terms for virtual
   resource partitioning are VPN Routing and Forwarding (VRF) instances
   and Virtual Switch Instances (VSIs).

   The YANG data model in this document conforms to the Network
   Management Datastore Architecture (NMDA) defined in RFC 8342.

Status of This Memo

Table of Contents

1. Introduction

   This document defines the second of two new modules that are defined
   to support the configuration and operation of network devices that
   allow for the partitioning of resources from both, or either,
   management and networking perspectives. Both leverage the YANG
   functionality enabled by YANG Schema Mount [RFC8528].

   The YANG data model in this document conforms to the Network
   Management Datastore Architecture defined in [RFC8342].

   The first form of resource partitioning provides a logical
   partitioning of a network device where each partition is separately
   managed as essentially an independent network element that is
   "hosted" by the base network device. These hosted network elements
   are referred to as logical network elements, or LNEs, and are
   supported by the logical-network-element module defined in [RFC8530].
   That module is used to identify LNEs and associate resources from the
   network device with each LNE. LNEs themselves are represented in
   YANG as independent network devices; each is accessed independently.
   Examples of vendor terminology for an LNE include logical system or
   logical router and virtual switch, chassis, or fabric.

   The second form, which is defined in this document, provides support
   for what are commonly referred to as VPN Routing and Forwarding (VRF)
   instances as well as Virtual Switch Instances (VSI); see [RFC4026]
   and [RFC4664]. In this form of resource partitioning, multiple
   control-plane and forwarding/bridging instances are provided by and
   managed through a single (physical or logical) network device. This
   form of resource partitioning is referred to as a Network Instance
   (NI) and is supported by the network instance module defined below.
   Configuration and operation of each network instance is always via
   the network device and the network instance module.

   One notable difference between the LNE model and the NI model is that
   the NI model provides a framework for VRF and VSI management. This
   document envisions the separate definition of models specific to VRF
   and VSI -- i.e., L3 and L2 VPN -- technology. An example of such can
   be found in the emerging L3VPN model defined in [YANG-L3VPN] and the
   examples discussed below.

1.1. Terminology

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

Readers are expected to be familiar with terms and concepts of YANG
[RFC7950] and YANG Schema Mount [RFC8528].

This document uses the graphical representation of data models
defined in [RFC8340].

2. Overview

In this document, we consider network devices that support protocols
and functions defined within the IETF -- e.g., routers, firewalls,
and hosts. Such devices may be physical or virtual, e.g., a classic
router with custom hardware or one residing within a server-based
virtual machine implementing a virtual network function (VNF). Each
device may subdivide their resources into logical network elements
(LNEs), each of which provides a managed logical device. Examples of
vendor terminology for an LNE include logical system or logical
router and virtual switch, chassis, or fabric. Each LNE may also
support VRF and VSI functions, which are referred to below as network
instances (NIs). This breakdown is represented in Figure 1.

```
           ,'''''''''''''''''''''''''''''''''''''''''''''`.
         ,'                                                 `.
         |        Network Device (Physical or Virtual)       |
         | ....................    .....................     |
         | :  Logical Network :    :  Logical Network  :     |
         | :      Element      :    :      Element      :     |
         | :+-----+-----+-----+:    :+-----+-----+-----+:     |
         | :| Net | Net | Net |:    :| Net | Net | Net |:     |
         | :|Inst.|Inst.|Inst.|:    :|Inst.|Inst.|Inst.|:     |
         | :+-----+-----+-----+:    :+-----+-----+-----+:     |
         | :  | |   | |   | |  :    :  | |   | |   | |   :     |
         | :..|.|...|.|...|.|..:    :..|.|...|.|...|.|..:     |
         | |  | |   | |   | |        | |   | |   | |    |     |
          `'''|'|'''|'|'''|'|'''''''''|'|'''|'|'''|'|'''''
              | |   | |   | |         | |   | |   | |
              | |   | |   | |         | |   | |   | |
                 Interfaces              Interfaces
```
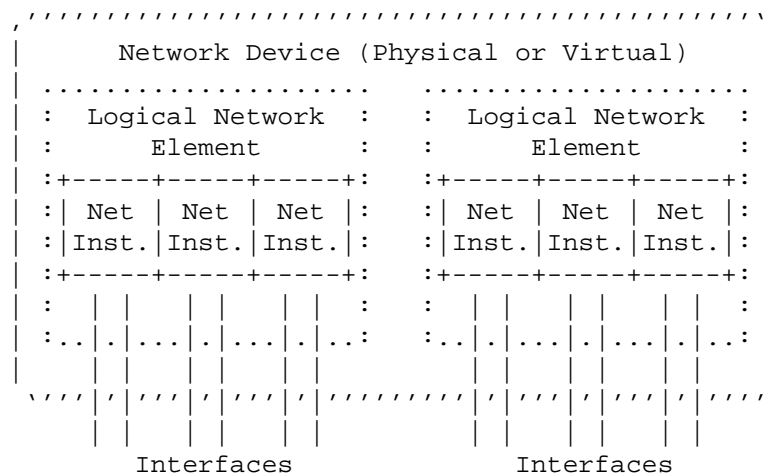
Figure 1: Module Element Relationships

A model for LNEs is described in [RFC8530], and the model for NIs is
covered in Section 3 of this document.

The current interface management model [RFC8343] is impacted by the
definition of LNEs and NIs. This document and [RFC8530] define
augmentations to the interface module to support LNEs and NIs.

The network instance model supports the configuration of VRFs and
VSIs. Each instance is supported by information that relates to the
device -- for example, the route target used when advertising VRF
routes via the mechanisms defined in [RFC4364], and information that
relates to the internal operation of the NI, such as for routing
protocols [RFC8349] and OSPF [YANG-OSPF]. This document defines the
network instance module that provides a basis for the management of
both types of information.

NI information that relates to the device, including the assignment
of interfaces to NIs, is defined as part of this document. The
defined module also provides a placeholder for the definition of
NI-technology-specific information both at the device level and for
NI internal operation. Information related to NI internal operation
is supported via schema mount [RFC8528] and mounting appropriate
modules under the mount point. Well-known mount points are defined
for L3VPN, L2VPN, and L2+L3VPN NI types.

3. Network Instances

   The network instance container is used to represent VRFs and VSIs.
   VRFs and VSIs are commonly used to isolate routing and switching
   domains -- for example, to create virtual private networks, each with
   their own active protocols and routing/switching policies. The model
   supports both core/provider and virtual instances. Core/provider
   instance information is accessible at the top level of the server,
   while virtual instance information is accessible under the root
   schema mount points.

```
   module: ietf-network-instance
     +--rw network-instances
        +--rw network-instance* [name]
           +--rw name            string
           +--rw enabled?        boolean
           +--rw description?    string
           +--rw (ni-type)?
           +--rw (root-type)
              +--:(vrf-root)
              |  +--mp vrf-root
              +--:(vsi-root)
              |  +--mp vsi-root
              +--:(vv-root)
                 +--mp vv-root
   augment /if:interfaces/if:interface:
     +--rw bind-ni-name?   -> /network-instances/network-instance/name
   augment /if:interfaces/if:interface/ip:ipv4:
     +--rw bind-ni-name?   -> /network-instances/network-instance/name
   augment /if:interfaces/if:interface/ip:ipv6:
     +--rw bind-ni-name?   -> /network-instances/network-instance/name

   notifications:
     +---n bind-ni-name-failed
        +--ro name           -> /if:interfaces/interface/name
        +--ro interface
        |  +--ro bind-ni-name?
        |              -> /if:interfaces/interface/ni:bind-ni-name
        +--ro ipv4
        |  +--ro bind-ni-name?
        |        -> /if:interfaces/interface/ip:ipv4/ni:bind-ni-name
        +--ro ipv6
        |  +--ro bind-ni-name?
        |        -> /if:interfaces/interface/ip:ipv6/ni:bind-ni-name
        +--ro error-info?   string
```

A network instance is identified by a "name" string. This string is
used both as an index within the network instance module and to
associate resources with a network instance, as shown above in the
interface augmentation. The ni-type and root-type choice statements
are used to support different types of L2 and L3 VPN technologies.
The bind-ni-name-failed notification is used in certain failure
cases.

3.1. NI Types and Mount Points

The network instance module is structured to facilitate the
definition of information models for specific types of VRFs and VSIs
using augmentations. For example, the information needed to support
L2VPN, such as VPLS and EVPN, are likely to be quite different.
Example models under development that could be restructured to take
advantage on NIs include models for L3VPNs [YANG-L3VPN] and L2VPNs
[YANG-L2VPN].

Documents defining new YANG data models for the support of specific
types of network instances should augment the network instance
module. The basic structure that should be used for such
augmentations includes a case statement with containers for
configuration and state data and, when needed, a type-specific mount
point. Generally, NI types are expected to not need to define type-
specific mount points but rather reuse one of the well-known mount
points, as defined in the next section. The following is an example
type-specific augmentation:

```
 augment "/ni:network-instances/ni:network-instance/ni:ni-type" {
   case l3vpn {
     container l3vpn {
         ...
     }
     container l3vpn-state {
         ...
     }
   }
 }
```

3.1.1. Well-Known Mount Points

   YANG Schema Mount [RFC8528] identifies mount points by name within a
   module. This definition allows for the definition of mount points
   whose schema can be shared across NI types. As discussed above,
   ni-types largely differ in the configuration information needed in
   the core/top-level instance to support the NI, rather than in the
   information represented within an NI. This allows the use of shared
   mount points across certain NI types.

   The expectation is that there are actually very few different schemas
   that need to be defined to support NIs for an implementation. In
   particular, it is expected that the following three forms of NI
   schema are needed, and each can be defined with a well-known mount
   point that can be reused by future modules defining NI types.

   The three well-known mount points are:

   vrf-root
      vrf-root is intended for use with L3VPN-type NI types.

   vsi-root
      vsi-root is intended for use with L2VPN-type Ni types.

   vv-root
      vv-root is intended for use with NI types that simultaneously
      support L2VPN bridging and L3VPN routing capabilities.

   Future model definitions should use the above mount points whenever
   possible. When a well-known mount point isn't appropriate, a model
   may define a type-specific mount point via augmentation.

3.1.2. NI Type Example

   The following is an example of an L3VPN VRF using a hypothetical
   augmentation to the network instance schema defined in [YANG-L3VPN].
   More detailed examples can be found in Appendix A.

```
   module: ietf-network-instance
     +--rw network-instances
        +--rw network-instance* [name]
           +--rw name            string
           +--rw enabled?        boolean
           +--rw description?    string
           +--rw (ni-type)?
           |  +--:(l3vpn)
           |     +--rw l3vpn:l3vpn
           |     |  ... // config data
           |     +--ro l3vpn:l3vpn-state
           |     |  ... // state data
           +--rw (root-type)
              +--:(vrf-root)
                 +--mp vrf-root
                    +--rw rt:routing/
                    |  +--rw router-id?                yang:dotted-quad
                    |  +--rw control-plane-protocols
                    |     +--rw control-plane-protocol* [type name]
                    |     +--rw ospf:ospf
                    |        +--rw area* [area-id]
                    |           +--rw interfaces
                    |              +--rw interface* [name]
                    |                 +--rw name if:interface-ref
                    |                 +--rw cost?   uint16
                    +--ro if:interfaces@
                    |  ...
```

   This shows YANG Routing Management [RFC8349] and YANG OSPF
   [YANG-OSPF] as mounted modules. The mounted modules can reference
   interface information via a parent-reference to the containers
   defined in [RFC8343].

3.2. NIs and Interfaces

   Interfaces are a crucial part of any network device's configuration
   and operational state. They generally include a combination of raw
   physical interfaces, link-layer interfaces, addressing configuration,
   and logical interfaces that may not be tied to any physical
   interface. Several system services and Layer 2 and Layer 3 protocols

may also associate configuration or operational state data with
different types of interfaces (these relationships are not shown for
simplicity). The interface management model is defined by [RFC8343].

As shown below, the network instance module augments the existing
interface management model by adding a name that is used on interface
or sub-interface types to identify an associated network instance.
Similarly, this name is also added for IPv4 and IPv6 types, as
defined in [RFC8344].

The following is an example of envisioned usage. The interfaces
container includes a number of commonly used components as examples:

```
module: ietf-interfaces
   +--rw interfaces
   |  +--rw interface* [name]
   |     +--rw name                        string
   |     +--rw ip:ipv4!
   |     |  +--rw ip:enabled?                   boolean
   |     |  +--rw ip:forwarding?                boolean
   |     |  +--rw ip:mtu?                       uint16
   |     |  +--rw ip:address* [ip]
   |     |  |  +--rw ip:ip              inet:ipv4-address-no-zone
   |     |  |  +--rw (ip:subnet)
   |     |  |     +--:(ip:prefix-length)
   |     |  |     |  +--rw ip:prefix-length?   uint8
   |     |  |     +--:(ip:netmask)
   |     |  |        +--rw ip:netmask?         yang:dotted-quad
   |     |  +--rw ip:neighbor* [ip]
   |     |  |  +--rw ip:ip              inet:ipv4-address-no-zone
   |     |  |  +--rw ip:link-layer-address yang:phys-address
   |     |  +--rw ni:bind-network-instance-name?   string
   |     +--rw ni:bind-network-instance-name?   string
```

The "ietf-interfaces" module [RFC8343] is structured to include all
interfaces in a flat list, without regard to virtual instances (e.g.,
VRFs) supported on the device. The bind-network-instance-name leaf
provides the association between an interface and its associated NI
(e.g., VRF or VSI). Note that as currently defined, to assign an
interface to both an LNE and an NI, the interface would first be
assigned to the LNE using the mechanisms defined in [RFC8530] and
then, within that LNE's interface module, the LNE's representation of
that interface would be assigned to an NI.

3.3. Network Instance Management

   Modules that may be used to represent network instance information
   will be available under the "root" mount point specific to the
   ni-type. The "shared-schema" method defined in the "ietf-yang-
   schema-mount" module [RFC8528] MUST be used to identify accessible
   modules. A future version of this document could relax this
   requirement. Mounted modules SHOULD be defined with access, via the
   appropriate schema mount parent-references [RFC8528], to device
   resources such as interfaces. An implementation MAY choose to
   restrict parent-referenced information to information related to a
   specific instance. For example, it might only allow references to
   interfaces that have a "bind-network-instance-name" that is identical
   to the instance's "name".

   All modules that represent control-plane and data-plane information
   may be present at the "root" mount point and accessible via paths
   modified per [RFC8528]. The list of available modules is expected to
   be implementation dependent, as is the method used by an
   implementation to support NIs.

   For example, the following could be used to define the data
   organization of the example NI shown in Section 3.1.2:

```
     "ietf-yang-schema-mount:schema-mounts": {
       "mount-point": [
         {
           "module": "ietf-network-instance",
           "label": "vrf-root",
           "shared-schema": {
             "parent-reference": [
               "/*[namespace-uri() = 'urn:ietf:...:ietf-interfaces']"
             ]
           }
         }
       ]
     }
```

   Module data identified according to the ietf-yang-schema-mount module
   will be instantiated under the mount point identified under
   "mount-point". These modules will be able to reference information
   for nodes belonging to top-level modules that are identified under
   "parent-reference". Parent-referenced information is available to
   clients via their top-level paths only and not under the associated
   mount point.

To allow a client to understand the previously mentioned instance
restrictions on parent-referenced information, an implementation MAY
represent such restrictions in the "parent-reference" leaf-list. For
example:

```
   "namespace": [
     {
       "prefix": "if",
       "uri": "urn:ietf:params:xml:ns:yang:ietf-interfaces"
     },
     {
       "prefix": "ni",
       "uri": "urn:ietf:params:xml:ns:yang:ietf-network-instance"
     }
   ],
   "mount-point": [
     {
       "module": "ietf-network-instance",
       "label": "vrf-root",
       "shared-schema": {
         "parent-reference": [
           "/if:interfaces/if:interface
              [ni:bind-network-instance-name = current()/../ni:name]",
           "/if:interfaces/if:interface/ip:ipv4
              [ni:bind-network-instance-name = current()/../ni:name]",
           "/if:interfaces/if:interface/ip:ipv6
              [ni:bind-network-instance-name = current()/../ni:name]"
         ]
       }
     }
   ],
```

   The same such "parent-reference" restrictions for non-NMDA
   implementations can be represented based on [RFC8343] and [RFC8344]
   as:

```
     "namespace": [
       {
         "prefix": "if",
         "uri": "urn:ietf:params:xml:ns:yang:ietf-interfaces"
       },
       {
         "prefix": "ni",
         "uri": "urn:ietf:params:xml:ns:yang:ietf-network-instance"
       }
     ],
     "mount-point": [
       {
         "module": "ietf-network-instance",
         "label": "vrf-root",
         "shared-schema": {
           "parent-reference": [
             "/if:interfaces/if:interface
                [ni:bind-network-instance-name = current()/../ni:name]",
             "/if:interfaces-state/if:interface
                [if:name = /if:interfaces/if:interface
                   [ni:bind-ni-name = current()/../ni:name]/if:name]",
             "/if:interfaces/if:interface/ip:ipv4
                [ni:bind-network-instance-name = current()/../ni:name]",
             "/if:interfaces-state/if:interface/ip:ipv4
                [if:name = /if:interfaces/if:interface/ip:ipv4
                   [ni:bind-ni-name = current()/../ni:name]/if:name]",
             "/if:interfaces/if:interface/ip:ipv6
                [ni:bind-network-instance-name = current()/../ni:name]",
             "/if:interfaces-state/if:interface/ip:ipv6
                [if:name = /if:interfaces/if:interface/ip:ipv6
                   [ni:bind-ni-name = current()/../ni:name]/if:name]"
           ]
         }
       }
     ],
```

3.4. Network Instance Instantiation

   Network instances may be controlled by clients using existing list
   operations. When a list entry is created, a new instance is
   instantiated. The models mounted under an NI root are expected to be
   dependent on the server implementation. When a list entry is
   deleted, an existing network instance is destroyed. For more
   information, see Section 7.8.6 of [RFC7950].

   Once instantiated, host network device resources can be associated
   with the new NI. As previously mentioned, this document augments
   ietf-interfaces with the bind-ni-name leaf to support such
   associations for interfaces. When a bind-ni-name is set to a valid
   NI name, an implementation MUST take whatever steps are internally
   necessary to assign the interface to the NI or provide an error
   message (defined below) with an indication of why the assignment
   failed. It is possible for the assignment to fail while processing
   the set operation or after asynchronous processing. Error
   notification in the latter case is supported via a notification.

4. Security Considerations

   The YANG modules specified in this document define a schema for data
   that is designed to be accessed via network management protocols such
   as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer
   is the secure transport layer, and the mandatory-to-implement secure
   transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer
   is HTTPS, and the mandatory-to-implement secure transport is TLS
   [RFC8446].

   The Network Configuration Access Control Model (NACM) [RFC8341]
   provides the means to restrict access for particular NETCONF or
   RESTCONF users to a preconfigured subset of all available NETCONF or
   RESTCONF protocol operations and content.

   There are two different sets of security considerations to consider
   in the context of this document. One set is security related to
   information contained within mounted modules. The security
   considerations for mounted modules are not substantively changed
   based on the information being accessible within the context of an
   NI. For example, when considering the modules defined in [RFC8349],
   the security considerations identified in that document are equally
   applicable, whether those modules are accessed at a server's root or
   under an NI instance's root node.

The second area for consideration is information contained in the NI
module itself. NI information represents network configuration and
route distribution policy information. As such, the security of this
information is important, but it is fundamentally no different than
any other interface or routing configuration information that has
already been covered in [RFC8343] and [RFC8349].

The vulnerable "config true" parameters and subtrees are the
following:

/network-instances/network-instance: This list specifies the network
   instances and the related control plane protocols configured on a
   device.

/if:interfaces/if:interface/*/bind-network-instance-name: This leaf
   indicates the NI instance to which an interface is assigned.

Unauthorized access to any of these lists can adversely affect the
routing subsystem of both the local device and the network. This may
lead to network malfunctions, delivery of packets to inappropriate
destinations, and other problems.

5. IANA Considerations

   This document registers a URI in the "IETF XML Registry" [RFC3688].

       URI: urn:ietf:params:xml:ns:yang:ietf-network-instance

       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-network-instance
     namespace: urn:ietf:params:xml:ns:yang:ietf-network-instance
     prefix: ni
     reference: RFC 8529

6. Network Instance Model

   The structure of the model defined in this document is described by
   the YANG module below.

```
<CODE BEGINS> file "ietf-network-instance@2019-01-21.yang"
module ietf-network-instance {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-instance";
  prefix ni;

  // import some basic types

  import ietf-interfaces {
    prefix if;
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }
  import ietf-ip {
    prefix ip;
    reference
      "RFC 8344: A YANG Data Model for IP Management";
  }
  import ietf-yang-schema-mount {
    prefix yangmnt;
    reference
      "RFC 8528: YANG Schema Mount";
  }

  organization
    "IETF Routing Area (rtgwg) Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/rtgwg>
     WG List:  <mailto:rtgwg@ietf.org>

     Author:   Lou Berger
               <mailto:lberger@labn.net>
     Author:   Christian Hopps
               <mailto:chopps@chopps.org>
     Author:   Acee Lindem
               <mailto:acee@cisco.com>
     Author:   Dean Bogdanovic
               <mailto:ivandean@gmail.com>";
  description
    "This module is used to support multiple network instances
     within a single physical or virtual device.  Network
     instances are commonly known as VRFs (VPN Routing and
     Forwarding) and VSIs (Virtual Switching Instances).
```

```
        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.

        Copyright (c) 2019 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

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

        This version of this YANG module is part of RFC 8529; see
        the RFC itself for full legal notices.";

     revision 2019-01-21 {
       description
         "Initial revision.";
       reference
         "RFC 8529";
     }

     // top-level device definition statements

     container network-instances {
       description
         "Network instances, each of which consists of
          VRFs and/or VSIs.";
       reference
         "RFC 8349: A YANG Data Model for Routing Management";
       list network-instance {
         key "name";
         description
           "List of network instances.";
         leaf name {
           type string;
           mandatory true;
           description
             "device-scoped identifier for the network
              instance.";
         }
         leaf enabled {
           type boolean;
```

```
          default "true";
          description
            "Flag indicating whether or not the network
             instance is enabled.";
        }
        leaf description {
          type string;
          description
            "Description of the network instance
             and its intended purpose.";
        }
        choice ni-type {
          description
            "This node serves as an anchor point for different types
             of network instances.  Each 'case' is expected to
             differ in terms of the information needed in the
             parent/core to support the NI and may differ in their
             mounted-schema definition.  When the mounted schema is
             not expected to be the same for a specific type of NI,
             a mount point should be defined.";
        }
        choice root-type {
          mandatory true;
          description
            "Well-known mount points.";
          container vrf-root {
            description
              "Container for mount point.";
            yangmnt:mount-point "vrf-root" {
              description
                "Root for L3VPN-type models.  This will typically
                 not be an inline-type mount point.";
            }
          }
          container vsi-root {
            description
              "Container for mount point.";
            yangmnt:mount-point "vsi-root" {
              description
                "Root for L2VPN-type models.  This will typically
                 not be an inline-type mount point.";
            }
          }
          container vv-root {
            description
              "Container for mount point.";
            yangmnt:mount-point "vv-root" {
              description
```

```
                    "Root models that support both L2VPN-type bridging
                     and L3VPN-type routing.  This will typically
                     not be an inline-type mount point.";
              }
            }
          }
        }
      }

      // augment statements

      augment "/if:interfaces/if:interface" {
        description
          "Add a node for the identification of the network
           instance associated with the information configured
           on a interface.

           Note that a standard error will be returned if the
           identified leafref isn't present.  If an interface cannot
           be assigned for any other reason, the operation SHALL fail
           with an error-tag of 'operation-failed' and an
           error-app-tag of 'ni-assignment-failed'.  A meaningful
           error-info that indicates the source of the assignment
           failure SHOULD also be provided.";
        leaf bind-ni-name {
          type leafref {
            path "/network-instances/network-instance/name";
          }
          description
            "Network instance to which an interface is bound.";
        }
      }
      augment "/if:interfaces/if:interface/ip:ipv4" {
        description
          "Add a node for the identification of the network
           instance associated with the information configured
           on an IPv4 interface.

           Note that a standard error will be returned if the
           identified leafref isn't present.  If an interface cannot
           be assigned for any other reason, the operation SHALL fail
           with an error-tag of 'operation-failed' and an
           error-app-tag of 'ni-assignment-failed'.  A meaningful
           error-info that indicates the source of the assignment
           failure SHOULD also be provided.";
        leaf bind-ni-name {
          type leafref {
            path "/network-instances/network-instance/name";
```

```
      }
        description
          "Network instance to which IPv4 interface is bound.";
      }
    }
    augment "/if:interfaces/if:interface/ip:ipv6" {
      description
        "Add a node for the identification of the network
         instance associated with the information configured
         on an IPv6 interface.

         Note that a standard error will be returned if the
         identified leafref isn't present.  If an interface cannot
         be assigned for any other reason, the operation SHALL fail
         with an error-tag of 'operation-failed' and an
         error-app-tag of 'ni-assignment-failed'.  A meaningful
         error-info that indicates the source of the assignment
         failure SHOULD also be provided.";
      leaf bind-ni-name {
        type leafref {
          path "/network-instances/network-instance/name";
        }
        description
          "Network instance to which IPv6 interface is bound.";
      }
    }

    // notification statements

    notification bind-ni-name-failed {
      description
        "Indicates an error in the association of an interface to an
         NI.  Only generated after success is initially returned when
         bind-ni-name is set.

         Note: Some errors may need to be reported for multiple
         associations, e.g., a single error may need to be reported
         for an IPv4 and an IPv6 bind-ni-name.

         At least one container with a bind-ni-name leaf MUST be
         included in this notification.";
      leaf name {
        type leafref {
          path "/if:interfaces/if:interface/if:name";
        }
        mandatory true;
        description
          "Contains the interface name associated with the
```

```
             failure.";
         }
       container interface {
         description
           "Generic interface type.";
         leaf bind-ni-name {
           type leafref {
             path "/if:interfaces/if:interface"
                 + "/ni:bind-ni-name";
           }
           description
             "Contains the bind-ni-name associated with the
              failure.";
         }
       }
       container ipv4 {
         description
           "IPv4 interface type.";
         leaf bind-ni-name {
           type leafref {
             path "/if:interfaces/if:interface/ip:ipv4/ni:bind-ni-name";
           }
           description
             "Contains the bind-ni-name associated with the
              failure.";
         }
       }
       container ipv6 {
         description
           "IPv6 interface type.";
         leaf bind-ni-name {
           type leafref {
             path "/if:interfaces/if:interface/ip:ipv6"
                 + "/ni:bind-ni-name";
           }
           description
             "Contains the bind-ni-name associated with the
              failure.";
         }
       }
       leaf error-info {
         type string;
         description
           "Optionally, indicates the source of the assignment
            failure.";
       }
     }
   }
```

   <CODE ENDS>

7. References

7.1. Normative References

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

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

   [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF)", RFC 6020,
              DOI 10.17487/RFC6020, October 2010,
              <https://www.rfc-editor.org/info/rfc6020>.

   [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
              and A. Bierman, Ed., "Network Configuration Protocol
              (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
              <https://www.rfc-editor.org/info/rfc6241>.

   [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure
              Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,
              <https://www.rfc-editor.org/info/rfc6242>.

   [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>.

   [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration
              Access Control Model", STD 91, RFC 8341,
              DOI 10.17487/RFC8341, March 2018,
              <https://www.rfc-editor.org/info/rfc8341>.

   [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>.

   [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management",
              RFC 8344, DOI 10.17487/RFC8344, March 2018,
              <https://www.rfc-editor.org/info/rfc8344>.

   [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol
              Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,
              <https://www.rfc-editor.org/info/rfc8446>.

   [RFC8528] Bjorklund, M. and L. Lhotka, "YANG Schema Mount",
              RFC 8528, DOI 10.17487/RFC8528, March 2019,
              <https://www.rfc-editor.org/info/rfc8528>.

7.2. Informative References

   [RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual
              Private Network (VPN) Terminology", RFC 4026,
              DOI 10.17487/RFC4026, March 2005,
              <https://www.rfc-editor.org/info/rfc4026>.

   [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private
              Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February
              2006, <https://www.rfc-editor.org/info/rfc4364>.

   [RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer
              2 Virtual Private Networks (L2VPNs)", RFC 4664,
              DOI 10.17487/RFC4664, September 2006,
              <https://www.rfc-editor.org/info/rfc4664>.

   [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>.

   [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>.

   [RFC8530] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X.
              Liu, "YANG Model for Logical Network Elements", RFC 8530,
              DOI 10.17487/RFC8530, March 2019.

   [YANG-L2VPN]
              Shah, H., Brissette, P., Chen, I., Hussain, I., Wen, B.,
              and K. Tiruveedhula, "YANG Data Model for MPLS-based
              L2VPN", Work in Progress, draft-ietf-bess-l2vpn-yang-09,
              October 2018.

   [YANG-L3VPN]
              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, draft-ietf-bess-
              l3vpn-yang-04, October 2018.

   [YANG-NETWORK]
              Lindem, A., Berger, L., Bogdanovic, D., and C. Hopps,
              "Network Device YANG Logical Organization", Work in
              Progress, draft-ietf-rtgwg-device-model-02, March 2017.

   [YANG-OSPF]
              Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem,
              "YANG Data Model for OSPF Protocol", Work in Progress,
              draft-ietf-ospf-yang-21, January 2019.

Appendix A. Example NI Usage

   The following subsections provide example uses of NIs.

A.1. Configuration Data

   The following shows an example where two customer-specific network
   instances are configured:

```
{
  "ietf-network-instance:network-instances": {
    "network-instance": [
      {
        "name": "vrf-red",
        "vrf-root": {
          "ietf-routing:routing": {
            "router-id": "192.0.2.1",
            "control-plane-protocols": {
              "control-plane-protocol": [
                {
                  "type": "ietf-routing:ospf",
                  "name": "1",
                  "ietf-ospf:ospf": {
                    "af": "ipv4",
                    "areas": {
                      "area": [
                        {
                          "area-id": "203.0.113.1",
                          "interfaces": {
                            "interface": [
                              {
                                "name": "eth1",
                                "cost": 10
                              }
                            ]
                          }
                        }
                      ]
                    }
                  }
                }
              ]
            }
          }
        }
      },
      {
        "name": "vrf-blue",
```

```
            "vrf-root": {
              "ietf-routing:routing": {
                "router-id": "192.0.2.2",
                "control-plane-protocols": {
                  "control-plane-protocol": [
                    {
                      "type": "ietf-routing:ospf",
                      "name": "1",
                      "ietf-ospf:ospf": {
                        "af": "ipv4",
                        "areas": {
                          "area": [
                            {
                              "area-id": "203.0.113.1",
                              "interfaces": {
                                "interface": [
                                  {
                                    "name": "eth2",
                                    "cost": 10
                                  }
                                ]
                              }
                            }
                          ]
                        }
                      }
                    }
                  ]
                }
              }
            }
          ]
        },

        "ietf-interfaces:interfaces": {
          "interface": [
            {
              "name": "eth0",
              "type": "iana-if-type:ethernetCsmacd",
              "ietf-ip:ipv4": {
                "address": [
                  {
                    "ip": "192.0.2.10",
                    "prefix-length": 24
                  }
                ]
              },
```

```
          "ietf-ip:ipv6": {
            "address": [
              {
                "ip": "2001:db8:0:2::10",
                "prefix-length": 64
              }
            ]
          }
        },
        {
          "name": "eth1",
          "type": "iana-if-type:ethernetCsmacd",
          "ietf-ip:ipv4": {
            "address": [
              {
                "ip": "192.0.2.11",
                "prefix-length": 24
              }
            ]
          },
          "ietf-ip:ipv6": {
            "address": [
              {
                "ip": "2001:db8:0:2::11",
                "prefix-length": 64
              }
            ]
          },
          "ietf-network-instance:bind-network-instance-name": "vrf-red"
        },
        {
          "name": "eth2",
          "type": "iana-if-type:ethernetCsmacd",
          "ietf-ip:ipv4": {
            "address": [
              {
                "ip": "192.0.2.11",
                "prefix-length": 24
              }
            ]
          },
          "ietf-ip:ipv6": {
            "address": [
              {
                "ip": "2001:db8:0:2::11",
                "prefix-length": 64
              }
            ]
```

```
            },
            "ietf-network-instance:bind-network-instance-name":
            "vrf-blue"
          }
        ]
      },

      "ietf-system:system": {
        "authentication": {
          "user": [
            {
              "name": "john",
              "password": "$0$password"
            }
          ]
        }
      }
    }
  }
```

A.2. State Data - Non-NMDA Version

   The following shows state data for the configuration example above
   based on [RFC8343], [RFC8344], and [RFC8349].

```
{
  "ietf-network-instance:network-instances": {
    "network-instance": [
      {
        "name": "vrf-red",
        "vrf-root": {
          "ietf-yang-library:modules-state": {
            "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
            "module": [
              {
                "name": "ietf-yang-library",
                "revision": "2019-01-04",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-yang-library",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-ospf",
                "revision": "2019-01-24",
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-routing",
```

```
                "revision": "2018-03-13",
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
                "conformance-type": "implement"
              }
            ]
          },
          "ietf-routing:routing-state": {
            "router-id": "192.0.2.1",
            "control-plane-protocols": {
              "control-plane-protocol": [
                {
                  "type": "ietf-routing:ospf",
                  "name": "1",
                  "ietf-ospf:ospf": {
                    "af": "ipv4",
                    "areas": {
                      "area": [
                        {
                          "area-id": "203.0.113.1",
                          "interfaces": {
                            "interface": [
                              {
                                "name": "eth1",
                                "cost": 10
                              }
                            ]
                          }
                        }
                      ]
                    }
                  }
                }
              ]
            }
          }
        }
      },
      {
        "name": "vrf-blue",
        "vrf-root": {
          "ietf-yang-library:modules-state": {
            "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
            "module": [
              {
                "name": "ietf-yang-library",
                "revision": "2019-01-04",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-yang-library",
```

```
                "conformance-type": "implement"
              },
              {
                "name": "ietf-ospf",
                "revision": "2019-01-24",
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-routing",
                "revision": "2018-03-13",
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
                "conformance-type": "implement"
              }
            ]
          },
          "ietf-routing:routing-state": {
            "router-id": "192.0.2.2",
            "control-plane-protocols": {
              "control-plane-protocol": [
                {
                  "type": "ietf-routing:ospf",
                  "name": "1",
                  "ietf-ospf:ospf": {
                    "af": "ipv4",
                    "areas": {
                      "area": [
                        {
                          "area-id": "203.0.113.1",
                          "interfaces": {
                            "interface": [
                              {
                                "name": "eth2",
                                "cost": 10
                              }
                            ]
                          }
                        }
                      ]
                    }
                  }
                }
              ]
            }
          }
        }
      }
    }
  ]
```

```
      },

      "ietf-interfaces:interfaces-state": {
        "interface": [
          {
            "name": "eth0",
            "type": "iana-if-type:ethernetCsmacd",
            "oper-status": "up",
            "phys-address": "00:01:02:A1:B1:C0",
            "statistics": {
              "discontinuity-time": "2017-06-26T12:34:56-05:00"
            },
            "ietf-ip:ipv4": {
              "address": [
                {
                  "ip": "192.0.2.10",
                  "prefix-length": 24
                }
              ]
            },
            "ietf-ip:ipv6": {
              "address": [
                {
                  "ip": "2001:db8:0:2::10",
                  "prefix-length": 64
                }
              ]
            }
          },
          {
            "name": "eth1",
            "type": "iana-if-type:ethernetCsmacd",
            "oper-status": "up",
            "phys-address": "00:01:02:A1:B1:C1",
            "statistics": {
              "discontinuity-time": "2017-06-26T12:34:56-05:00"
            },
            "ietf-ip:ipv4": {
              "address": [
                {
                  "ip": "192.0.2.11",
                  "prefix-length": 24
                }
              ]
            },
            "ietf-ip:ipv6": {
              "address": [
                {
```

```
              "ip": "2001:db8:0:2::11",
              "prefix-length": 64
            }
          ]
        }
      },
      {
        "name": "eth2",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "phys-address": "00:01:02:A1:B1:C2",
        "statistics": {
          "discontinuity-time": "2017-06-26T12:34:56-05:00"
        },
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "192.0.2.11",
              "prefix-length": 24
            }
          ]
        },
        "ietf-ip:ipv6": {
          "address": [
            {
              "ip": "2001:db8:0:2::11",
              "prefix-length": 64
            }
          ]
        }
      }
    ]
  },

  "ietf-system:system-state": {
    "platform": {
      "os-name": "NetworkOS"
    }
  },

  "ietf-yang-library:modules-state": {
    "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
    "module": [
      {
        "name": "iana-if-type",
        "revision": "2018-07-03",
        "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
        "conformance-type": "import"
```

```
      },
      {
        "name": "ietf-inet-types",
        "revision": "2013-07-15",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
        "conformance-type": "import"
      },
      {
        "name": "ietf-interfaces",
        "revision": "2018-02-20",
        "feature": [
          "arbitrary-names",
          "pre-provisioning"
        ],
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-ip",
        "revision": "2018-01-09",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-network-instance",
        "revision": "2018-02-03",
        "feature": [
          "bind-network-instance-name"
        ],
        "namespace":
          "urn:ietf:params:xml:ns:yang:ietf-network-instance",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-ospf",
        "revision": "2019-01-24",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-routing",
        "revision": "2018-03-13",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-routing",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-system",
        "revision": "2014-08-06",
```

```
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-system",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-yang-library",
        "revision": "2019-01-04",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-library",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-yang-schema-mount",
        "revision": "2019-01-14",
        "namespace":
          "urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount",
        "conformance-type": "implement"
      },
      {
        "name": "ietf-yang-types",
        "revision": "2013-07-15",
        "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
        "conformance-type": "import"
      }
    ]
  },

  "ietf-yang-schema-mount:schema-mounts": {
    "mount-point": [
      {
        "module": "ietf-network-instance",
        "label": "vrf-root",
        "shared-schema": {
          "parent-reference": [
            "/*[namespace-uri() = 'urn:ietf:...:ietf-interfaces']"
          ]
        }
      }
    ]
  }
}
```

A.3. State Data - NMDA Version

   The following shows state data for the configuration example above
   based on [RFC8343], [RFC8344], and [RFC8349].

```
  {
    "ietf-network-instance:network-instances": {
      "network-instance": [
        {
          "name": "vrf-red",
          "vrf-root": {
            "ietf-yang-library:yang-library": {
              "content-id": "41e2ab5dc325f6d86f743e8da3de323f1a61a801",
              "module-set": [
                {
                  "name": "ni-modules",
                  "module": [
                    {
                      "name": "ietf-yang-library",
                      "revision": "2019-01-04",
                      "namespace":
                        "urn:ietf:params:xml:ns:yang:ietf-yang-library"
                    },
                    {
                      "name": "ietf-ospf",
                      "revision": "2019-01-24",
                      "namespace":
                        "urn:ietf:params:xml:ns:yang:ietf-ospf"
                    },
                    {
                      "name": "ietf-routing",
                      "revision": "2018-03-13",
                      "namespace":
                        "urn:ietf:params:xml:ns:yang:ietf-routing"
                    }
                  ],
                  "import-only-module": [
                    {
                      "name": "ietf-inet-types",
                      "revision": "2013-07-15",
                      "namespace":
                        "urn:ietf:params:xml:ns:yang:ietf-inet-types"
                    },
                    {
                      "name": "ietf-yang-types",
                      "revision": "2013-07-15",
                      "namespace":
                        "urn:ietf:params:xml:ns:yang:ietf-yang-types"
```

```
              },
              {
                "name": "ietf-datastores",
                "revision": "2018-02-14",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-datastores"
              }
            ]
          }
        ],
        "schema": [
          {
            "name": "ni-schema",
            "module-set": [ "ni-modules" ]
          }
        ],
        "datastore": [
          {
            "name": "ietf-datastores:running",
            "schema": "ni-schema"
          },
          {
            "name": "ietf-datastores:operational",
            "schema": "ni-schema"
          }
        ]
      },
      "ietf-routing:routing": {
        "router-id": "192.0.2.1",
        "control-plane-protocols": {
          "control-plane-protocol": [
            {
              "type": "ietf-routing:ospf",
              "name": "1",
              "ietf-ospf:ospf": {
                "af": "ipv4",
                "areas": {
                  "area": [
                    {
                      "area-id": "203.0.113.1",
                      "interfaces": {
                        "interface": [
                          {
                            "name": "eth1",
                            "cost": 10
                          }
                        ]
                      }
```

```
                        }
                      ]
                    }
                  }
                }
              ]
            }
          }
        }
      },
      {
        "name": "vrf-blue",
        "vrf-root": {
          "ietf-yang-library:yang-library": {
            "checksum": "41e2ab5dc325f6d86f743e8da3de323f1a61a801",
            "module-set": [
              {
                "name": "ni-modules",
                "module": [
                  {
                    "name": "ietf-yang-library",
                    "revision": "2019-01-04",
                    "namespace":
                      "urn:ietf:params:xml:ns:yang:ietf-yang-library",
                    "conformance-type": "implement"
                  },
                  {
                    "name": "ietf-ospf",
                    "revision": "2019-01-24",
                    "namespace":
                      "urn:ietf:params:xml:ns:yang:ietf-ospf",
                    "conformance-type": "implement"
                  },
                  {
                    "name": "ietf-routing",
                    "revision": "2018-03-13",
                    "namespace":
                      "urn:ietf:params:xml:ns:yang:ietf-routing",
                    "conformance-type": "implement"
                  }
                ],
                "import-only-module": [
                  {
                    "name": "ietf-inet-types",
                    "revision": "2013-07-15",
                    "namespace":
                      "urn:ietf:params:xml:ns:yang:ietf-inet-types"
                  },
```

```
                      {
                        "name": "ietf-yang-types",
                        "revision": "2013-07-15",
                        "namespace":
                          "urn:ietf:params:xml:ns:yang:ietf-yang-types"
                      },
                      {
                        "name": "ietf-datastores",
                        "revision": "2018-02-14",
                        "namespace":
                          "urn:ietf:params:xml:ns:yang:ietf-datastores"
                      }
                    ]
                  }
                ],
                "schema": [
                  {
                    "name": "ni-schema",
                    "module-set": [ "ni-modules" ]
                  }
                ],
                "datastore": [
                  {
                    "name": "ietf-datastores:running",
                    "schema": "ni-schema"
                  },
                  {
                    "name": "ietf-datastores:operational",
                    "schema": "ni-schema"
                  }
                ]
              },
              "ietf-routing:routing": {
                "router-id": "192.0.2.2",
                "control-plane-protocols": {
                  "control-plane-protocol": [
                    {
                      "type": "ietf-routing:ospf",
                      "name": "1",
                      "ietf-ospf:ospf": {
                        "af": "ipv4",
                        "areas": {
                          "area": [
                            {
                              "area-id": "203.0.113.1",
                              "interfaces": {
                                "interface": [
                                  {
```

```
                               "name": "eth2",
                               "cost": 10
                             }
                           ]
                         }
                       }
                     ]
                   }
                 }
               }
             ]
           }
         }
       }
     }
   ]
 },

 "ietf-interfaces:interfaces": {
   "interface": [
     {
       "name": "eth0",
       "type": "iana-if-type:ethernetCsmacd",
       "oper-status": "up",
       "phys-address": "00:01:02:A1:B1:C0",
       "statistics": {
         "discontinuity-time": "2017-06-26T12:34:56-05:00"
       },
       "ietf-ip:ipv4": {
         "address": [
           {
             "ip": "192.0.2.10",
             "prefix-length": 24
           }
         ]
       },
       "ietf-ip:ipv6": {
         "address": [
           {
             "ip": "2001:db8:0:2::10",
             "prefix-length": 64
           }
         ]
       }
     },
     {
       "name": "eth1",
       "type": "iana-if-type:ethernetCsmacd",
```

```
        "oper-status": "up",
        "phys-address": "00:01:02:A1:B1:C1",
        "statistics": {
          "discontinuity-time": "2017-06-26T12:34:56-05:00"
        },
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "192.0.2.11",
              "prefix-length": 24
            }
          ]
        },
        "ietf-ip:ipv6": {
          "address": [
            {
              "ip": "2001:db8:0:2::11",
              "prefix-length": 64
            }
          ]
        }
      },
      {
        "name": "eth2",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "phys-address": "00:01:02:A1:B1:C2",
        "statistics": {
          "discontinuity-time": "2017-06-26T12:34:56-05:00"
        },
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "192.0.2.11",
              "prefix-length": 24
            }
          ]
        },
        "ietf-ip:ipv6": {
          "address": [
            {
              "ip": "2001:db8:0:2::11",
              "prefix-length": 64
            }
          ]
        }
      }
    }
  ]
```

```
      },

      "ietf-system:system-state": {
        "platform": {
          "os-name": "NetworkOS"
        }
      },

      "ietf-yang-library:yang-library": {
        "content-id": "75a43df9bd56b92aacc156a2958fbe12312fb285",
        "module-set": [
          {
            "name": "host-modules",
            "module": [
              {
                "name": "ietf-interfaces",
                "revision": "2018-02-20",
                "feature": [
                  "arbitrary-names",
                  "pre-provisioning"
                ],
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-interfaces"
              },
              {
                "name": "ietf-ip",
                "revision": "2018-01-09",
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip"
              },
              {
                "name": "ietf-network-instance",
                "revision": "2018-02-03",
                "feature": [
                  "bind-network-instance-name"
                ],
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-network-instance"
              },
              {
                "name": "ietf-ospf",
                "revision": "2019-01-24",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-ospf"
              },
              {
                "name": "ietf-routing",
                "revision": "2018-03-13",
                "namespace":
```

```
              "urn:ietf:params:xml:ns:yang:ietf-routing"
            },
            {
              "name": "ietf-system",
              "revision": "2014-08-06",
              "namespace": "urn:ietf:params:xml:ns:yang:ietf-system"
            },
            {
              "name": "ietf-yang-library",
              "revision": "2019-01-04",
              "namespace":
                "urn:ietf:params:xml:ns:yang:ietf-yang-library"
            },
            {
              "name": "ietf-yang-schema-mount",
              "revision": "2019-01-14",
              "namespace":
                "urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount"
            }
          ],
          "import-only-module": [
            {
              "name": "iana-if-type",
              "revision": "2018-07-03",
              "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type"
            },
            {
              "name": "ietf-inet-types",
              "revision": "2013-07-15",
              "namespace":
                "urn:ietf:params:xml:ns:yang:ietf-inet-types"
            },
            {
              "name": "ietf-yang-types",
              "revision": "2013-07-15",
              "namespace":
                "urn:ietf:params:xml:ns:yang:ietf-yang-types"
            },
            {
              "name": "ietf-datastores",
              "revision": "2018-02-14",
              "namespace":
                "urn:ietf:params:xml:ns:yang:ietf-datastores"
            }
          ]
        }
      ],
      "schema": [
```

```
          {
            "name": "host-schema",
            "module-set": [ "host-modules" ]
          }
        ],
        "datastore": [
          {
            "name": "ietf-datastores:running",
            "schema": "host-schema"
          },
          {
            "name": "ietf-datastores:operational",
            "schema": "host-schema"
          }
        ]
      },

      "ietf-yang-schema-mount:schema-mounts": {
        "mount-point": [
          {
            "module": "ietf-network-instance",
            "label": "vrf-root",
            "shared-schema": {
              "parent-reference": [
                "/*[namespace-uri() = 'urn:ietf:...:ietf-interfaces']"
              ]
            }
          }
        ]
      }
    }
```

Acknowledgments

Authors' Addresses

   Lou Berger
   LabN Consulting, L.L.C.

   Email: lberger@labn.net

   Christian Hopps
   LabN Consulting, L.L.C.

   Email: chopps@chopps.org

   Acee Lindem
   Cisco Systems
   301 Midenhall Way
   Cary, NC 27513
   United States of America

   Email: acee@cisco.com

   Dean Bogdanovic
   Volta Networks

   Email: ivandean@gmail.com

   Xufeng Liu
   Volta Networks

   Email: xufeng.liu.ietf@gmail.com