Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2018

X. Liu, Ed.
A. Kyparlis
Jabil
R. Parikh
VMware
A. Lindem
Cisco Systems
M. Zhang
Huawei Technologies
February 22, 2018

A YANG Data Model for Virtual Router Redundancy Protocol (VRRP)
draft-ietf-rtgwg-yang-vrrp-11

Abstract

This document describes a data model for Virtual Router Redundancy Protocol (VRRP). Both version 2 and version 3 of VRRP are covered.

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 http://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 August 26, 2018.

Table of Contents

1. Introduction

This document introduces a YANG [RFC6020][RFC7950] data model for Virtual Router Redundancy Protocol (VRRP) [RFC3768] [RFC5798]. VRRP provides higher resiliency by specifying an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN.

This YANG model supports both version 2 and version 3 of VRRP. VRRP version 2 defined in [RFC3768] supports IPv4. VRRP version 3 defined in [RFC5798] supports both IPv4 and IPv6.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following terms are defined in [RFC7950] and are not redefined here:

o augment

- data model
- data node

1.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is defined in [I-D.ietf-netmod-yang-tree-diagrams].

1.3. 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 it is clear from the context in which YANG module 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]
if	ietf-interfaces	[I-D.ietf-netmod-rfc7223bis]
ip	ietf-ip	[I-D.ietf-netmod-rfc7277bis]

Table 1: Prefixes and Corresponding YANG Modules

2. Design of the Data Model

2.1. Scope of the Model

The model covers VRRP version 2 [RFC3768] and VRRP version 3 [RFC5798] protocols. The model is designed to be implemented on a device where VRRP version 2 or version 3 is implemented. With the help of a proper management protocol, the defined model can be used to:

- Configure the VRRP version 2 or version 3 protocol.
- Manage the protocol operational behavior.
- Retrieve the protocol operational status.
- Receive the protocol notifications.

2.2. Relations with Interface Model and IP Model

This model augments the interface data model "ietf-interfaces" [I-D.ietf-netmod-rfc7223bis] and the IP management model "ietf-ip" [I-D.ietf-netmod-rfc7277bis]. The augmentation relations are shown as follows:

```
module: ietf-interfaces
   +--rw interfaces
      +--rw interface* [name]
            ...
         +--rw ip:ipv4!
         |  +--rw ip:address* [ip]
         |        ...
         |  +--rw vrrp:vrrp
         |     +--rw vrrp:vrrp-instance* [vrid]
         |        +--rw vrrp:vrid                          uint8
         |        +--rw vrrp:virtual-ipv4-addresses
         |              ...
         +--rw ip:ipv6!
            +--rw ip:address* [ip]
            |     ...
            +--rw vrrp:vrrp
               +--rw vrrp:vrrp-instance* [vrid]
                  +--rw vrrp:vrid                          uint8
                  +--rw vrrp:virtual-ipv6-addresses
                        ...
```

In the above figure, a tree node without a prefix is from the model "ietf-interfaces". A tree node with prefix "ip:" is from the model "ietf-ip". A tree node with prefix "vrrp:" is from the VRRP model specified in this document.

The "vrrp" container contains a list of vrrp-instance nodes, which are instantiated under an interface for a specified address family (IPv4 or IPv6).

Each vrrp-instance node represents a VRRP router state machine described in Section 6.4 of [RFC5798], providing the configuration and state information for the election process of a virtual router. The IP addresses on the augmented interface are the real addresses through which the VRRP router operates. The IPv4 or IPv6 address(es) associated with a virtual router (described in Section 1 of [RFC5798]) are modeled as a list of IPv4 or IPv6 addresses under the vrrp-instance.

2.3. Protocol Configuration

The model structure for the protocol configuration is as shown below:

```
  augment /if:interfaces/if:interface/ip:ipv4:
    +--rw vrrp
       +--rw vrrp-instance* [vrid]
          +--rw vrid                              uint8
          |     ...
          +--rw track
          |  +--rw interfaces
          |  |  +--rw interface* [interface]
          |  |     +--rw interface             if:interface-ref
          |  |           ...
          |  +--rw networks
          |     +--rw network* [prefix]
          |        +--rw prefix                inet:ipv4-prefix
          |              ...
          +--rw virtual-ipv4-addresses
             +--rw virtual-ipv4-address* [ipv4-address]
                +--rw ipv4-address    inet:ipv4-address

  augment /if:interfaces/if:interface/ip:ipv6:
    +--rw vrrp
       +--rw vrrp-instance* [vrid]
          +--rw vrid                              uint8
          |     ...
          +--rw track
          |  +--rw interfaces
          |  |  +--rw interface* [interface]
          |  |     +--rw interface             if:interface-ref
          |  |           ...
          |  +--rw networks
          |     +--rw network* [prefix]
          |        +--rw prefix                inet:ipv6-prefix
          |              ...
          +--rw virtual-ipv6-addresses
             +--rw virtual-ipv6-address* [ipv6-address]
                +--rw ipv6-address    inet:ipv6-address
```

The model allows to configure the following protocol entities:

- VRRP instance (version 2 or version 3), representing a VRRP router.

- Virtual IPv4 or IPv6 address associated with a virtual router.

- Tracking interface, to detect interface connectivity failures.

- Tracking network, to detect interface connectivity failures.

2.4. Protocol States

The model structure for the protocol states is as shown below:

```
module: ietf-vrrp
    +--ro vrrp
       |     // global operational states
       +--ro virtual-routers?   uint32
       +--ro interfaces?        uint32
       +--ro statistics                 // global statistics
          +--ro discontinuity-datetime?   yang:date-and-time
          +--ro checksum-errors?          yang:counter64
          +--ro version-errors?           yang:counter64
          +--ro vrid-errors?              yang:counter64
          +--ro ip-ttl-errors?            yang:counter64

  augment /if:interfaces/if:interface/ip:ipv4:
    +--rw vrrp
       +--rw vrrp-instance* [vrid]
          +--rw vrid                             uint8
          |     ...
          +--rw track
          |  +--rw interfaces
          |  |  +--rw interface* [interface]
          |  |     +--rw interface             if:interface-ref
          |  |           ...
          |  +--rw networks
          |     +--rw network* [prefix]
          |        +--rw prefix                inet:ipv4-prefix
          |              ...
          +--rw virtual-ipv4-addresses
          |  +--rw virtual-ipv4-address* [ipv4-address]
          |     +--rw ipv4-address    inet:ipv4-address
          |
          |     // per instance operational states
          +--ro state?                           identityref
          +--ro is-owner?                        boolean
          +--ro last-adv-source?                 inet:ip-address
          +--ro up-datetime?                     yang:date-and-time
          +--ro master-down-interval?            uint32
          +--ro skew-time?                       uint32
          +--ro last-event?                      identityref
          +--ro new-master-reason?               new-master-reason-type
```

```
            +--ro statistics                // per instance statistics
               +--ro discontinuity-datetime?    yang:date-and-time
               +--ro master-transitions?        yang:counter32
               +--ro advertisement-recv?        yang:counter64
               +--ro advertisement-sent?        yang:counter64
               +--ro interval-errors?           yang:counter64
               |       {validate-interval-errors}?
               +--ro priority-zero-pkts-rcvd?   yang:counter64
               +--ro priority-zero-pkts-sent?   yang:counter64
               +--ro invalid-type-pkts-rcvd?    yang:counter64
               +--ro address-list-errors?       yang:counter64
               |       {validate-address-list-errors}?
               +--ro packet-length-errors?      yang:counter64

     augment /if:interfaces/if:interface/ip:ipv6:
       +--rw vrrp
          +--rw vrrp-instance* [vrid]
             +--rw vrid                           uint8
             +     ...
             +--rw track
             |  +--rw interfaces
             |  |  +--rw interface* [interface]
             |  |     +--rw interface             if:interface-ref
             |  |           ...
             |  +--rw networks
             |     +--rw network* [prefix]
             |        +--rw prefix                inet:ipv6-prefix
             |              ...
             +--rw virtual-ipv6-addresses
             |  +--rw virtual-ipv6-address* [ipv6-address]
             |     +--rw ipv6-address    inet:ipv6-address
             |
             |     // per instance operational states
             +--ro state?                         identityref
             +--ro is-owner?                      boolean
             +--ro last-adv-source?               inet:ip-address
             +--ro up-datetime?                   yang:date-and-time
             +--ro master-down-interval?          uint32
             +--ro skew-time?                     uint32
             +--ro last-event?                    identityref
             +--ro new-master-reason?             new-master-reason-type
             +--ro statistics                // per instance statistics
                +--ro discontinuity-datetime?    yang:date-and-time
                +--ro master-transitions?        yang:counter32
                +--ro advertisement-recv?        yang:counter64
                +--ro advertisement-sent?        yang:counter64
                +--ro interval-errors?           yang:counter64
                |       {validate-interval-errors}?
```

```
            +--ro priority-zero-pkts-rcvd?   yang:counter64
            +--ro priority-zero-pkts-sent?   yang:counter64
            +--ro invalid-type-pkts-rcvd?    yang:counter64
            +--ro address-list-errors?       yang:counter64
            |       {validate-address-list-errors}?
            +--ro packet-length-errors?      yang:counter64
```

This model conforms to the Network Management Datastore Architecture (NMDA) [I-D.ietf-netmod-revised-datastores]. The operational state data is combined with the associated configuration data in the same hierarchy [I-D.ietf-netmod-rfc6087bis]. When protocol states are retrieved from the NMDA operational state datastore, the returned states cover all "config true" (rw) and "config false" (ro) nodes defined in the schema.

The model allows to retrieve protocol states at the following levels:

- VRRP instance (version 2 or version 3), representing a VRRP router.
- Virtual IPv4 or IPv6 address associated with a virtual router.
- Tracking interface, to detect interface connectivity failures.
- Tracking network, to detect interface connectivity failures.
- Global states and statistics summarizing all instances.

2.5. Notifications

This model defines the following VRRP specific notifications:

```
notifications:
   +---n vrrp-new-master-event
   |  +--ro master-ip-address    inet:ip-address
   |  +--ro new-master-reason    new-master-reason-type
   +---n vrrp-protocol-error-event
   |  +--ro protocol-error-reason    identityref
   +---n vrrp-virtual-router-error-event
      +--ro interface                     if:interface-ref
      +--ro (ip-version)
      |  +--:(ipv4)
      |  |  +--ro ipv4
      |  |     +--ro vrid    leafref
      |  +--:(ipv6)
      |     +--ro ipv6
      |        +--ro vrid    leafref
      +--ro virtual-router-error-reason    identityref
```

Each notification type is used to indicate a type of VRRP state changes or error occurances:

vrrp-new-master-event
: VRRP new master event, indicating that a new master has been elected.

vrrp-protocol-error-event
: VRRP protocol error event for a message that fails to reach a VRRP instance to be processed.

vrrp-virtual-router-error-event
: VRRP virtual router error event for a message processed on a VRRP instance.

In addition to the notifications specified above, the mechanism defined in [I-D.ietf-netconf-subscribed-notifications] and [I-D.ietf-netconf-yang-push] can be used for other general notifications. This mechanism currently allows the user to:

- Subscribe notifications on a per client basis.
- Specify subtree filters or xpath filters so that only interested contents will be sent.
- Specify either periodic or on-demand notifications.

3. Tree Structure

The VRRP YANG data model defined in this document has the following tree structure:

```
module: ietf-vrrp
    +--ro vrrp
       +--ro virtual-routers?   uint32
       +--ro interfaces?        uint32
       +--ro statistics
          +--ro discontinuity-datetime?   yang:date-and-time
          +--ro checksum-errors?          yang:counter64
          +--ro version-errors?           yang:counter64
          +--ro vrid-errors?              yang:counter64
          +--ro ip-ttl-errors?            yang:counter64
  augment /if:interfaces/if:interface/ip:ipv4:
    +--rw vrrp
       +--rw vrrp-instance* [vrid]
          +--rw vrid                              uint8
          +--rw version                           identityref
          +--rw log-state-change?                 boolean
          +--rw preempt
          |  +--rw enabled?     boolean
          |  +--rw hold-time?   uint16
          +--rw priority?                         uint8
          +--rw accept-mode?                      boolean
          +--rw (advertise-interval-choice)?
          |  +--:(v2)
          |  |  +--rw advertise-interval-sec?         uint8
          |  +--:(v3)
          |     +--rw advertise-interval-centi-sec?   uint16
          +--rw track
          |  +--rw interfaces
          |  |  +--rw interface* [interface]
          |  |     +--rw interface             if:interface-ref
          |  |     +--rw priority-decrement?   uint8
          |  +--rw networks
          |     +--rw network* [prefix]
          |        +--rw prefix                inet:ipv4-prefix
          |        +--rw priority-decrement?   uint8
          +--rw virtual-ipv4-addresses
          |  +--rw virtual-ipv4-address* [ipv4-address]
          |     +--rw ipv4-address    inet:ipv4-address
          +--ro state?                            identityref
          +--ro is-owner?                         boolean
          +--ro last-adv-source?                  inet:ip-address
          +--ro up-datetime?                      yang:date-and-time
          +--ro master-down-interval?             uint32
```

```
            +--ro skew-time?                      uint32
            +--ro last-event?                     identityref
            +--ro new-master-reason?
    new-master-reason-type
            +--ro statistics
               +--ro discontinuity-datetime?    yang:date-and-time
               +--ro master-transitions?        yang:counter32
               +--ro advertisement-recv?        yang:counter64
               +--ro advertisement-sent?        yang:counter64
               +--ro interval-errors?           yang:counter64
               |       {validate-interval-errors}?
               +--ro priority-zero-pkts-rcvd?   yang:counter64
               +--ro priority-zero-pkts-sent?   yang:counter64
               +--ro invalid-type-pkts-rcvd?    yang:counter64
               +--ro address-list-errors?       yang:counter64
               |       {validate-address-list-errors}?
               +--ro packet-length-errors?      yang:counter64
     augment /if:interfaces/if:interface/ip:ipv6:
       +--rw vrrp
          +--rw vrrp-instance* [vrid]
             +--rw vrid                           uint8
             +--rw version                        identityref
             +--rw log-state-change?              boolean
             +--rw preempt
             |  +--rw enabled?     boolean
             |  +--rw hold-time?   uint16
             +--rw priority?                      uint8
             +--rw accept-mode?                   boolean
             +--rw advertise-interval-centi-sec?  uint16
             +--rw track
             |  +--rw interfaces
             |  |  +--rw interface* [interface]
             |  |     +--rw interface             if:interface-ref
             |  |     +--rw priority-decrement?   uint8
             |  +--rw networks
             |     +--rw network* [prefix]
             |        +--rw prefix                inet:ipv6-prefix
             |        +--rw priority-decrement?   uint8
             +--rw virtual-ipv6-addresses
             |  +--rw virtual-ipv6-address* [ipv6-address]
             |     +--rw ipv6-address    inet:ipv6-address
             +--ro state?                         identityref
             +--ro is-owner?                      boolean
             +--ro last-adv-source?               inet:ip-address
             +--ro up-datetime?                   yang:date-and-time
             +--ro master-down-interval?          uint32
             +--ro skew-time?                     uint32
             +--ro last-event?                    identityref
```

```
          +--ro new-master-reason?
    new-master-reason-type
          +--ro statistics
             +--ro discontinuity-datetime?    yang:date-and-time
             +--ro master-transitions?        yang:counter32
             +--ro advertisement-recv?        yang:counter64
             +--ro advertisement-sent?        yang:counter64
             +--ro interval-errors?           yang:counter64
             |       {validate-interval-errors}?
             +--ro priority-zero-pkts-rcvd?   yang:counter64
             +--ro priority-zero-pkts-sent?   yang:counter64
             +--ro invalid-type-pkts-rcvd?    yang:counter64
             +--ro address-list-errors?       yang:counter64
             |       {validate-address-list-errors}?
             +--ro packet-length-errors?      yang:counter64

     notifications:
       +---n vrrp-new-master-event
       |  +--ro master-ip-address    inet:ip-address
       |  +--ro new-master-reason    new-master-reason-type
       +---n vrrp-protocol-error-event
       |  +--ro protocol-error-reason    identityref
       +---n vrrp-virtual-router-error-event
          +--ro interface                      if:interface-ref
          +--ro (ip-version)
          |  +--:(ipv4)
          |  |  +--ro ipv4
          |  |     +--ro vrid    leafref
          |  +--:(ipv6)
          |     +--ro ipv6
          |        +--ro vrid    leafref
          +--ro virtual-router-error-reason    identityref
```

4. YANG Module

This module references [RFC2787], [RFC3768], [RFC5798] and [RFC6527].

```
<CODE BEGINS> file "ietf-vrrp@2018-01-09.yang"
module ietf-vrrp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vrrp";
  prefix "vrrp";

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

```
    import ietf-yang-types {
      prefix "yang";
    }

    import ietf-interfaces {
      prefix "if";
    }

    import ietf-ip {
      prefix "ip";
    }

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

       Editor:   Xufeng Liu
                 <mailto:Xufeng_Liu@jabil.com>

       Editor:   Athanasios Kyparlis
                 <mailto:Athanasios_Kyparlis@jabil.com>

       Editor:   Ravi Parikh
                 <mailto:parikhr@vmware.com>

       Editor:   Acee Lindem
                 <mailto:acee@cisco.com>

       Editor:   Mingui Zhang
                 <mailto:zhangmingui@huawei.com>";

    description
      "This YANG module defines a model for managing Virtual Router
       Redundancy Protocol (VRRP) version 2 and version 3.

       Copyright (c) 2018 IETF Trust and the persons identified as
       authors of the code.  All rights reserved.

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

       This version of this YANG module is part of RFC XXXX; see the
```

```
        RFC itself for full legal notices.";

    revision 2018-01-09 {
      description "Initial revision";
      reference
        "RFC XXXX: A YANG Data Model for Virtual Router Redundancy
         Protocol (VRRP).
         RFC 2787: Definitions of Managed Objects for the Virtual
         Router Redundancy Protocol.
         RFC 3768: Virtual Router Redundancy Protocol (VRRP).
         RFC 5798: Virtual Router Redundancy Protocol (VRRP) Version 3.
         RFC 6527: Definitions of Managed Objects for the Virtual
         Router Redundancy Protocol Version 3 (VRRPv3).";
    }

    /*
     * Features
     */

    feature validate-interval-errors {
      description
        "This feature indicates that the system validates that
         the advertisement interval from advertisement packets
         received is the same as the one configured for the local
         VRRP router.";
    }

    feature validate-address-list-errors {
      description
        "This feature indicates that the system validates that
         the address list from received packets matches the
         locally configured list for the VRRP router.";
    }

    /*
     * Typedefs
     */

    typedef new-master-reason-type {
      type enumeration {
        enum not-master {
          description
            "The virtual router has never transitioned to master
             state,";
        }
        enum priority {
          description "Priority was higher.";
        }
```

```
        enum preempted {
          description "The master was preempted.";
        }
        enum no-response {
          description "Previous master did not respond.";
        }
      }
      description
        "The reason for the virtual router to transition to master
         state.";
    } // new-master-reason-type

    /*
     * Identities
     */

    /* vrrp-event-type identity and its derivatives. */
    identity vrrp-event-type {
      description
        "The type to indicate the type of a VRRP protocol event.";
    }
    identity vrrp-event-none {
      base vrrp-event-type;
      description
        "Indicates a non-meaningful event.";
    }
    identity vrrp-event-startup {
      base vrrp-event-type;
      description
        "Indicates that a VRRP router has initiated the protocol.";
    }
    identity vrrp-event-shutdown {
      base vrrp-event-type;
      description
        "Indicates that a VRRP router has closed down the protocol.";
    }
    identity vrrp-event-higher-priority-backup {
      base vrrp-event-type;
      description
        "Indicates that a backup router has a higher priority than
         the current master.";
    }
    identity vrrp-event-master-timeout {
      base vrrp-event-type;
      description
        "Indicates that the current master has not sent an
         advertisement within the limit of master-down-interval.";
    }
```

```
    identity vrrp-event-interface-up {
      base vrrp-event-type;
      description
        "Indicates that the VRRP enabled interface has become
         operational up.";
    }
    identity vrrp-event-interface-down {
      base vrrp-event-type;
      description
        "Indicates that the VRRP enabled interface has become
         operational down.";
    }
    identity vrrp-event-no-primary-ip-address {
      base vrrp-event-type;
      description
        "Indicates that the primary IP address on the VRRP enabled
         interface has become unavailable.";
    }
    identity vrrp-event-primary-ip-address {
      base vrrp-event-type;
      description
        "Indicates that the primary IP address on the VRRP enabled
         interface has become available.";
    }
    identity vrrp-event-no-virtual-ip-addresses {
      base vrrp-event-type;
      description
        "Indicates that there are no virtual IP addresses on the
         virtual router.";
    }
    identity vrrp-event-virtual-ip-addresses {
      base vrrp-event-type;
      description
        "Indicates that there are virtual IP addresses on the
         virtual router.";
    }
    identity vrrp-event-preempt-hold-timeout {
      base vrrp-event-type;
      description
        "Indicates that the configured preemption hold time has
         passed.";
    }
    identity vrrp-event-lower-priority-master {
      base vrrp-event-type;
      description
        "Indicates that there is a lower priority VRRP master.";
    }
    identity vrrp-event-owner-preempt {
```

```
      base vrrp-event-type;
      description
        "Indicates that the owner has preempted another router to
         become the master.";
    }

    /* vrrp-error-global identity and its derivatives. */
    identity vrrp-error-global {
      description
        "The type to indicate the type of a VRRP error that occurs
         for a packet before it reaches a VRRP router.";
    }
    identity checksum-error {
      base vrrp-error-global;
      description
        "A packet has been received with an invalid VRRP checksum
         value.";
    }
    identity ip-ttl-error {
      base vrrp-error-global;
      description
        "A packet has been received with IP TTL (Time-To-Live)
         not equal to 255.";
    }
    identity version-error {
      base vrrp-error-global;
      description
        "A packet has been received with an unknown or unsupported
         version number.";
    }
    identity vrid-error {
      base vrrp-error-global;
      description
        "A packet has been received with a VRID that is not valid
         for any virtual router on this router.";
    }

    /* vrrp-error-virtual-router identity and its derivatives. */
    identity vrrp-error-virtual-router {
      description
        "The type to indicate the type of a VRRP error that occurs
         after a packet reaches a VRRP router.";
    }
    identity address-list-error {
      base vrrp-error-virtual-router;
      description
        "A packet has been received with an address list that
         does not match the locally configured address list for
```

```
         the virtual router.";
      }
      identity interval-error {
        base vrrp-error-virtual-router;
        description
          "A packet has been received with an advertisement
           interval different than the one configured for the local
           virtual router";
      }
      identity packet-length-error {
        base vrrp-error-virtual-router;
        description
          "A packet has been received with a packet length less
           than the length of the VRRP header.";
      }

      /* vrrp-state-type identity and its derivatives. */
      identity vrrp-state-type {
        description
          "The type to indicate the state of a virtual router.";
      }
      identity initialize {
        base vrrp-state-type;
        description
          "Indicates that the virtual router is waiting
           for a startup event.";
      }
      identity backup {
        base vrrp-state-type;
        description
          "Indicates that the virtual router is monitoring the
           availability of the master router.";
      }
      identity master {
        base vrrp-state-type;
        description
          "Indicates that the virtual router is forwarding
           packets for IP addresses that are associated with
           this virtual router.";
      }

      /* vrrp-version identity and its derivatives. */
      identity vrrp-version {
        description
          "The version of the VRRP protocol.";
      }
      identity vrrp-v2 {
        base vrrp-version;
```

```
      description
        "Indicates version 2 of the VRRP protocol.";
    }
    identity vrrp-v3 {
      base vrrp-version;
      description
        "Indicates version 3 of the VRRP protocol.";
    }

    /*
     * Groupings
     */

    grouping vrrp-common-attributes {
      description
        "Group of VRRP attributes common to version 2 and version 3";

      leaf vrid {
        type uint8 {
          range 1..255;
        }
        description "Virtual router ID.";
      }

      leaf version {
        type identityref {
          base vrrp:vrrp-version;
        }
        mandatory true;
        description "Version 2 or version 3 of VRRP.";
      }

      leaf log-state-change {
        type boolean;
        default "false";
        description
          "Generates VRRP state change messages each time the VRRP
           instance changes state (from up to down or down to up).";
      }

      container preempt {
        description
          "Enables a higher priority Virtual Router Redundancy
           Protocol (VRRP) backup router to preempt a lower priority
           VRRP master.";
        leaf enabled {
          type boolean;
          default "true";
```

```
          description
            "'true' if preemption is enabled.";
        }
        leaf hold-time {
          type uint16;
          units seconds;
          default 0;
          description
            "Hold time, in seconds, for which a higher priority VRRP
             backup router must wait before preempting a lower priority
             VRRP master.";
        }
      }

      leaf priority {
        type uint8 {
          range 1..254;
        }
        default 100;
        description
          "Configures the Virtual Router Redundancy Protocol (VRRP)
           election priority for the backup virtual router.";
      }

      leaf accept-mode {
        when "derived-from-or-self(current()/../version, 'vrrp-v3')" {
          description "Applicable only to version 3.";
        }
        type boolean;
        default "false";
        description
          "Controls whether a virtual router in Master state will
           accept packets addressed to the address owner's IPvX address
           as its own if it is not the IPvX address owner. The default
           is false. Deployments that rely on, for example, pinging the
           address owner's IPvX address may wish to configure
           accept-mode to true.

           Note: IPv6 Neighbor Solicitations and Neighbor
           Advertisements MUST NOT be dropped when accept-mode is
           false.";
      }
    } // vrrp-common-attributes

    grouping vrrp-ipv4-attributes {
      description
        "Group of VRRP attributes for IPv4.";
```

```
      uses vrrp-common-attributes;

      choice advertise-interval-choice {
        description
          "The options for the advertisement interval at which VRRPv2
           or VRRPv3 advertisements are sent from the specified
           interface.";

        case v2 {
          when "derived-from-or-self(version, 'vrrp-v2')" {
            description "Applicable only to version 2.";
          }
          leaf advertise-interval-sec {
            type uint8 {
              range 1..254;
            }
            units seconds;
            default 1;
            description
              "Configures the interval that Virtual Router
               Redundancy Protocol Version 2 (VRRPv2) advertisements
               are sent from the specified interface.";
          }
        }

        case v3 {
          when "derived-from-or-self(version, 'vrrp-v3')" {
            description "Applicable only to version 3.";
          }
          leaf advertise-interval-centi-sec {
            type uint16 {
              range 1..4095;
            }
            units centiseconds;
            default 100;
            description
              "Configures the interval that Virtual Router
               Redundancy Protocol version 3 (VRRPv3) advertisements
               are sent from the specified interface.";
          }
        }
      } // advertise-interval-choice

      container track {
        description
          "Enables the specified VRRP instance to track interfaces
           or networks.";
        container interfaces {
```

```
      description
        "Enables the specified Virtual Router Redundancy Protocol
         version 2 (VRRP) or version 3 (VRRPv3) instance to track
         interfaces.
         Interface tracking prevents traffic loss by detecting the
         availability of interfaces. The operational states of
         other interfaces are associated with the priority of a
         VRRP router. When a tracked interface becomes unavailable
         (or operational down), the priority of the backup router
         decrements. When an unavailable interface becomes
         available again, the priority of the backup VRRP router is
         incremented by the same amount.";

      list interface {
        key "interface";
        description
          "Interface to track.";

        leaf interface {
          type if:interface-ref;
          must "/if:interfaces/if:interface[if:name=current()]/"
             + "ip:ipv4" {
            description "Interface is IPv4.";
          }
          description
            "Interface to track.";
        }

        leaf priority-decrement {
          type uint8 {
            range 1..254;
          }
          default 10;
          description
            "Specifies how much to decrement the priority of the
             VRRP instance if the interface goes down.";
        }
      } // interface
    } // interfaces

    container networks {
      description
        "Enables the backup Virtual Router Redundancy Protocol
         version 2 (VRRP) or version 3 (VRRPv3) router to track
         specified networks through the IP network prefixes of
         these networks.
         Network tracking prevents traffic loss by detecting
         network connectivity failure. The states of connectivity
```

```
            to some networks are associated with the priority of a
            VRRP router. When connectivity to a tracked network
            represented by its prefix is lost, the priority of the
            backup VRRP router decrements. When an unavailable network
            is again reachable, the priority of the backup VRRP router
            is incremented by the same amount.";
          list network {
            key "prefix";
            description
              "Enables the specified Virtual Router Redundancy
               Protocol version 2 (VRRP) or version 3 (VRRPv3)
               instance to track an IP network, by specifying the
               prefix of the IP network.";

            leaf prefix {
              type inet:ipv4-prefix;
              description
                "The prefix of the network to track.";
            }

            leaf priority-decrement {
              type uint8 {
                range 1..254;
              }
              default 10;
              description
                "Specifies how much to decrement the priority of the
                 backup VRRP router if there is a failure in the IP
                 network.";
            }
          } // track-network
        } // track-networks
      } // track

      container virtual-ipv4-addresses {
        description
          "Configures the virtual IP address for the Virtual Router
           Redundancy Protocol (VRRP) interface.";

        list virtual-ipv4-address {
          key "ipv4-address";
          max-elements 16;
          description
            "Virtual IP addresses for a single VRRP instance. For a
             VRRP owner router, the virtual address must match one
             of the IP addresses configured on the interface
             corresponding to the virtual router.";
```

```
          leaf ipv4-address {
            type inet:ipv4-address;
            description
              "An IPv4 address associated with a virtual router.";
            reference
              "RFC 5798: Virtual Router Redundancy Protocol (VRRP)
               Version 3. Section 1.2.";
          }
        } // virtual-ipv4-address
      } // virtual-ipv4-addresses
    } // grouping vrrp-ipv4-attributes

    grouping vrrp-ipv6-attributes {
      description
        "Group of VRRP attributes for IPv6.";

      uses vrrp-common-attributes;

      leaf advertise-interval-centi-sec {
        type uint16 {
          range 1..4095;
        }
        units centiseconds;
        default 100;
        description
          "Configures the interval that Virtual Router
           Redundancy Protocol version 3 (VRRPv3) advertisements
           are sent from the specified interface.";
      }

      container track {
        description
          "Enables the specified VRRP instance to track interfaces
           or networks.";
        container interfaces {
          description
            "Enables the specified Virtual Router Redundancy Protocol
             version 2 (VRRP) or version 3 (VRRPv3) instance to track
             interfaces.
             Interface tracking prevents traffic loss by detecting the
             availability of interfaces. The operational states of
             other interfaces are associated with the priority of a
             VRRP router. When a tracked interface becomes unavailable
             (or operational down), the priority of the backup router
             decrements. When an unavailable interface becomes
             available again, the priority of the backup VRRP router is
             incremented by the same amount.";
          list interface {
```

```
        key "interface";
        description
          "Interface to track.";

        leaf interface {
          type if:interface-ref;
          must "/if:interfaces/if:interface[if:name=current()]/"
             + "ip:ipv6" {
            description "Interface is IPv6.";
          }
          description
            "Interface to track.";
        }

        leaf priority-decrement {
          type uint8 {
            range 1..254;
          }
          default 10;
          description
            "Specifies how much to decrement the priority of the
             VRRP instance if the interface goes down.";
        }
      } // interface
    } // interfaces

    container networks {
      description
        "Enables the backup Virtual Router Redundancy Protocol
         version 2 (VRRP) or version 3 (VRRPv3) router to track
         specified networks through the IP network prefixes of
         these networks.
         Network tracking prevents traffic loss by detecting
         network connectivity failure. The states of connectivity
         to some networks are associated with the priority of a
         VRRP router. When connectivity to a tracked network
         represented by its prefix is lost, the priority of the
         backup VRRP router decrements. When an unavailable network
         is again reachable, the priority of the backup VRRP router
         is incremented by the same amount.";
      list network {
        key "prefix";
        description
          "Enables the specified Virtual Router Redundancy
           Protocol version 2 (VRRP) or version 3 (VRRPv3)
           instance to track an IP network, by specifying the
           prefix of the IP network.";
```

```
            leaf prefix {
              type inet:ipv6-prefix;
              description
                "The prefix of the network to track.";
            }

            leaf priority-decrement {
              type uint8 {
                range 1..254;
              }
              default 10;
              description
                "Specifies how much to decrement the priority of the
                 backup VRRP router if there is a failure in the IP
                 network.";
            }
          } // track-network
        } // track-networks
      } // track

      container virtual-ipv6-addresses {
        description
          "Configures the virtual IP address for the Virtual Router
           Redundancy Protocol (VRRP) interface.";
        list virtual-ipv6-address {
          key "ipv6-address";
          max-elements 2;
          description
            "Two IPv6 addresses are allowed. The first one must be
             a link-local address and the second one can be a
             link-local or global address.";

          leaf ipv6-address {
            type inet:ipv6-address;
            description
              "An IPv6 address associated with a virtual router.";
            reference
              "RFC 5798: Virtual Router Redundancy Protocol (VRRP)
               Version 3. Section 1.3.";
          }
        } // virtual-ipv6-address
      } // virtual-ipv6-addresses
    } // grouping vrrp-ipv6-attributes

    grouping vrrp-state-attributes {
      description
        "Group of VRRP state attributes.";
```

```
      leaf state {
        type identityref {
          base vrrp:vrrp-state-type;
        }
        config false;
        description
          "Operational state.";
      }

      leaf is-owner {
        type boolean;
        config false;
        description
          "Set to true if this virtual router is owner.";
      }

      leaf last-adv-source {
        type inet:ip-address;
        config false;
        description
          "Last advertised IPv4/IPv6 source address";
      }

      leaf up-datetime {
        type yang:date-and-time;
        config false;
        description
          "The date and time when this virtual router
           transitioned out of init state.";
      }

      leaf master-down-interval {
        type uint32;
        units centiseconds;
        config false;
        description
          "Time interval for backup virtual router to declare
           Master down.";
      }

      leaf skew-time {
        type uint32;
        units microseconds;
        config false;
        description
          "Calculated based on the priority and advertisement
           interval configuration command parameters. See RFC 3768.";
      }
```

```
      leaf last-event {
        type identityref {
          base vrrp:vrrp-event-type;
        }
        config false;
        description
          "Last reported event.";
      }

      leaf new-master-reason {
        type new-master-reason-type;
        config false;
        description
          "Indicates the reason for the virtual router to transition
           to master state.";
      }

      container statistics {
        config false;
        description
          "VRRP statistics.";

        leaf discontinuity-datetime {
          type yang:date-and-time;
          description
            "The time on the most recent occasion at which any one or
             more of the VRRP statistic counters suffered a
             discontinuity.  If no such discontinuities have occurred
             since the last re-initialization of the local management
             subsystem, then this node contains the time that the
             local management subsystem re-initialized itself.";
        }

        leaf master-transitions {
          type yang:counter32;
          description
            "The total number of times that this virtual router's
             state has transitioned to master";
        }

        leaf advertisement-recv {
          type yang:counter64;
          description
            "The total number of VRRP advertisements received by
             this virtual router.";
        }

        leaf advertisement-sent {
```

```
          type yang:counter64;
          description
            "The total number of VRRP advertisements sent by
             this virtual router.";
        }

        leaf interval-errors {
          if-feature validate-interval-errors;
          type yang:counter64;
          description
            "The total number of VRRP advertisement packets
             received with an advertisement interval
             different than the one configured for the local
             virtual router";
        }

        leaf priority-zero-pkts-rcvd {
          type yang:counter64;
          description
            "The total number of VRRP packets received by the
             virtual router with a priority of 0.";
        }

        leaf priority-zero-pkts-sent {
          type yang:counter64;
          description
            "The total number of VRRP packets sent by the
             virtual router with a priority of 0.";
        }

        leaf invalid-type-pkts-rcvd {
          type yang:counter64;
          description
            "The number of VRRP packets received by the virtual
             router with an invalid value in the 'type' field.";
        }

        leaf address-list-errors {
          if-feature validate-address-list-errors;
          type yang:counter64;
          description
            "The total number of packets received with an
             address list that does not match the locally
             configured address list for the virtual router.";
        }

        leaf packet-length-errors {
          type yang:counter64;
```

```
          description
            "The total number of packets received with a packet
             length less than the length of the VRRP header.";
        }
      } // container statistics
    } // grouping vrrp-state-attributes

    grouping vrrp-global-state-attributes {
      description
        "Group of VRRP global state attributes.";

      leaf virtual-routers {
        type uint32;
        description "Number of configured virtual routers.";
      }

      leaf interfaces {
        type uint32;
        description "Number of interface with VRRP configured.";
      }

      container statistics {
        description
          "VRRP global statistics.";

        leaf discontinuity-datetime {
          type yang:date-and-time;
          description
            "The time on the most recent occasion at which one of
             checksum-errors, version-errors, vrid-errors, and
             ip-ttl-errors suffered a discontinuity.

             If no such discontinuities have occurred since the last
             re-initialization of the local management subsystem,
             then this node contains the time that the local management
             subsystem re-initialized itself.";
        }

        leaf checksum-errors {
          type yang:counter64;
          description
            "The total number of VRRP packets received with an invalid
             VRRP checksum value.";
          reference "RFC 5798, Section 5.2.8";
        }

        leaf version-errors {
          type yang:counter64;
```

```
          description
            "The total number of VRRP packets received with an unknown
             or unsupported version number.";
          reference "RFC 5798, Section 5.2.1";
        }

        leaf vrid-errors {
          type yang:counter64;
          description
            "The total number of VRRP packets received with a VRID that
             is not valid for any virtual router on this router.";
          reference "RFC 5798, Section 5.2.3";
        }

        leaf ip-ttl-errors {
          type yang:counter64;
          description
            "The total number of VRRP packets received by the
             virtual router with IP TTL (Time-To-Live) not equal
             to 255.";
          reference "RFC 5798, Sections 5.1.1.3 and 5.1.2.3.";
        }
      } // statistics
    } // vrrp-global-state-attributes

    /*
     * Configuration data and operational state data nodes
     */

    augment "/if:interfaces/if:interface/ip:ipv4" {
      description "Augment IPv4 interface.";

      container vrrp {
        description
          "Configures the Virtual Router Redundancy Protocol (VRRP)
           version 2 or version 3 for IPv4.";

        list vrrp-instance {
          key "vrid";
          description
            "Defines a virtual router, identified by a virtual router
             identifier (VRID), within IPv4 address space.";

          uses vrrp-ipv4-attributes;
          uses vrrp-state-attributes;
        }
      }
    } // augment ipv4
```

```
    augment "/if:interfaces/if:interface/ip:ipv6" {
      description "Augment IPv6 interface.";

      container vrrp {
        description
          "Configures the Virtual Router Redundancy Protocol (VRRP)
           version 3 for IPv6.";

        list vrrp-instance {
          must "derived-from-or-self(version, 'vrrp-v3')" {
            description
              "IPv6 is only supported by version 3.";
          }
          key "vrid";
          description
            "Defines a virtual router, identified by a virtual router
             identifier (VRID), within IPv6 address space.";

          uses vrrp-ipv6-attributes;
          uses vrrp-state-attributes;
        } // list vrrp-instance
      } // container vrrp
    } // augment ipv6

    container vrrp {
      config false;
      description "VRRP data at the global level.";

      uses vrrp-global-state-attributes;
    }

    /*
     * Notifications
     */

    notification vrrp-new-master-event {
      description
        "Notification event for a change of VRRP new master.";
      leaf master-ip-address {
        type inet:ip-address;
        mandatory true;
        description
          "IPv4 or IPv6 address of the new master.";
      }
      leaf new-master-reason {
        type new-master-reason-type;
        mandatory true;
        description
```

```
          "Indicates the reason for the virtual router to transition
           to master state.";
      }
    }

    notification vrrp-protocol-error-event {
      description
        "Notification event for a VRRP protocol error.";
      leaf protocol-error-reason {
        type identityref {
          base vrrp:vrrp-error-global;
        }
        mandatory true;
        description
          "Indicates the reason for the protocol error.";
      }
    }

    notification vrrp-virtual-router-error-event {
      description
        "Notification event for an error happened on a virtual
         router.";
      leaf interface {
        type if:interface-ref;
        mandatory true;
        description
          "Indicates the interface for which statistics area
           to be cleared.";
      }

      choice ip-version {
        mandatory true;
        description
          "The error may have happened on either an IPv4 virtual
           router or an IPv6 virtual router. The information
           related to a specific IP version is provided by one of
           the following cases.";
        case ipv4 {
          description "IPv4";
          container ipv4 {
            description
              "Error information for IPv4.";
            leaf vrid {
              type leafref {
                path "/if:interfaces/if:interface"
                  + "[if:name = current()/../../vrrp:interface]/"
                  + "ip:ipv4/vrrp:vrrp/vrrp:vrrp-instance/vrrp:vrid";
              }
```

```
              mandatory true;
              description
                "Indicates the virtual router on which the event has
                 occurred.";
            }
          }
        }
        case ipv6 {
          description "IPv6";
          container ipv6 {
            description
              "Error information for IPv6.";
            leaf vrid {
              type leafref {
                path "/if:interfaces/if:interface"
                  + "[if:name = current()/../../vrrp:interface]/"
                  + "ip:ipv6/vrrp:vrrp/vrrp:vrrp-instance/vrrp:vrid";
              }
              mandatory true;
              description
                "Indicates the virtual router on which the event has
                 occurred.";
            }
          }
        }
      }

      leaf virtual-router-error-reason {
        type identityref {
          base vrrp:vrrp-error-virtual-router;
        }
        mandatory true;
        description
          "Indicates the reason for the virtual router error.";
      }
    }
  }
  <CODE ENDS>
```

5. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```
---------------------------------------------------------------------
URI: urn:ietf:params:xml:ns:yang:ietf-vrrp
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
---------------------------------------------------------------------
```

This document registers the following YANG modules in the YANG Module Names registry [RFC7950]:

```
---------------------------------------------------------------------
name:         ietf-vrrp
namespace:    urn:ietf:params:xml:ns:yang:ietf-vrrp
prefix:       vrrp
reference:    RFC XXXX
---------------------------------------------------------------------
```

6. Security Considerations

The YANG module specified in this document defines 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 [RFC5246].

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

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/if:interfaces/if:interface/ip:ipv4/vrrp:vrrp/vrrp:vrrp-instance

/if:interfaces/if:interface/ip:ipv6/vrrp:vrrp/vrrp:vrrp-instance

Unauthorized access to any data node of these subtrees 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.

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

/ietf-vrrp:vrrp

/if:interfaces/if:interface/ip:ipv4/vrrp:vrrp/vrrp:vrrp-instance

/if:interfaces/if:interface/ip:ipv6/vrrp:vrrp/vrrp:vrrp-instance

Unauthorized access to any data node of these subtrees can disclose the operational state information of VRRP on this device.

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>.

[RFC2787] Jewell, B. and D. Chuang, "Definitions of Managed Objects for the Virtual Router Redundancy Protocol", RFC 2787, DOI 10.17487/RFC2787, March 2000, <https://www.rfc-editor.org/info/rfc2787>.

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <https://www.rfc-editor.org/info/rfc5798>.

[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>.

[RFC6527] Tata, K., "Definitions of Managed Objects for Virtual Router Redundancy Protocol Version 3 (VRRPv3)", RFC 6527, DOI 10.17487/RFC6527, March 2012, <https://www.rfc-editor.org/info/rfc6527>.

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

[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>.

[I-D.ietf-netmod-rfc7223bis] Bjorklund, M., "A YANG Data Model for Interface Management", draft-ietf-netmod-rfc7223bis-03 (work in progress), January 2018.

[I-D.ietf-netmod-rfc7277bis] Bjorklund, M., "A YANG Data Model for IP Management", draft-ietf-netmod-rfc7277bis-03 (work in progress), January 2018.

[I-D.ietf-netmod-revised-datastores] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-10 (work in progress), January 2018.

7.2. Informative References

[RFC3768] Hinden, R., Ed., "Virtual Router Redundancy Protocol (VRRP)", RFC 3768, DOI 10.17487/RFC3768, April 2004, <https://www.rfc-editor.org/info/rfc3768>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[I-D.ietf-netconf-subscribed-notifications] Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Custom Subscription to Event Streams", draft-ietf-netconf-subscribed-notifications-09 (work in progress), January 2018.

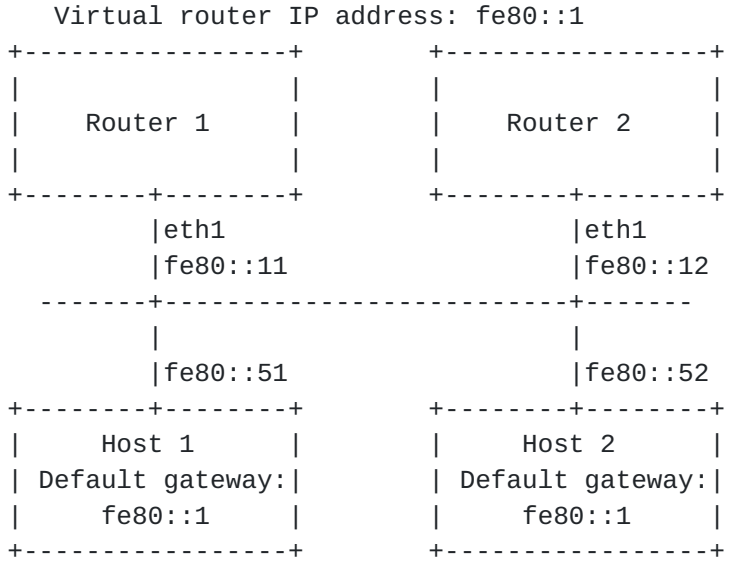
[I-D.ietf-netconf-yang-push] Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", draft-ietf-netconf-yang-push-14 (work in progress), February 2018.

[I-D.ietf-netmod-rfc6087bis] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-17 (work in progress), February 2018.

[I-D.ietf-netmod-yang-tree-diagrams] Bjorklund, M. and L. Berger, "YANG Tree Diagrams", draft-ietf-netmod-yang-tree-diagrams-06 (work in progress), February 2018.

Appendix A. Data Tree Example

This section contains an example of an instance data tree in the JSON encoding [RFC7951], containing both configuration and state data.



The configuration instance data for Router 1 in the above figure could be as follows:

```
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1",
        "description": "An interface with VRRP enabled.",
        "type": "iana-if-type:ethernetCsmacd",
        "ietf-ip:ipv6": {
          "address": [
            {
              "ip": "2001:db8:0:1::1",
              "prefix-length": 64
            },
            {
              "ip": "fe80::11",
              "prefix-length": 64
            }
          ],
          "forwarding": true,
          "ietf-vrrp:vrrp": {
            "vrrp-instance": [
              {
                "vrid": 1,
                "version": "vrrp-v3",
                "priority": 200,
                "advertise-interval-centi-sec": 50,
                "virtual-ipv6-addresses": {
                  "virtual-ipv6-address": [
                    "ipv6-address": "fe80::1"
                  ]
                }
              }
            ]
          }
        }
      }
    ]
  }
}
```

The cooresponding operational state data for Router 1 could be as follows:

```
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1",
```

```
            "description": "An interface with VRRP enabled.",
            "type": "iana-if-type:ethernetCsmacd",
            "phys-address": "00:00:5e:00:53:01",
            "oper-status": "up",
            "statistics": {
              "discontinuity-time": "2016-10-24T17:11:27+02:00"
            },
            "ietf-ip:ipv6": {
              "forwarding": true,
              "mtu": 1500,
              "address": [
                {
                  "ip": "2001:db8:0:1::1",
                  "prefix-length": 64,
                  "origin": "static",
                  "status": "preferred"
                },
                {
                  "ip": "fe80::11",
                  "prefix-length": 64,
                  "origin": "static",
                  "status": "preferred"
                }
              ]
              "ietf-vrrp:vrrp": {
                "vrrp-instance": [
                  {
                    "vrid": 1,
                    "version": "vrrp-v3",
                    "log-state-change": false,
                    "preempt": {
                      "enabled": true,
                      "hold-time": 0
                    }
                    "priority": 200,
                    "accept-mode": false,
                    "advertise-interval-centi-sec": 50,
                    "virtual-ipv6-addresses": {
                      "virtual-ipv6-address": [
                        "ipv6-address": "fe80::1"
                      ]
                    },
                    "state": "master",
                    "is-owner": false,
                    "last-adv-source": "fe80::11",
                    "up-datetime": "2016-10-24T17:11:27+02:00",
                    "master-down-interval": 161,
                    "skew-time": 11,
```

```
                  "last-event": "vrrp-event-interface-up",
                  "new-master-reason": "priority",
                  "statistics": {
                    "discontinuity-datetime":
                      "2016-10-24T17:11:27+02:00",
                    "master-transitions": 2,
                    "advertisement-recv": 20,
                    "advertisement-sent": 12,
                    "interval-errors": 0,
                    "priority-zero-pkts-rcvd": 0,
                    "priority-zero-pkts-sent": 0,
                    "invalid-type-pkts-rcvd": 0,
                    "address-list-errors": 0,
                    "packet-length-errors": 1
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }

  {
    "ietf-vrrp:vrrp": {
      "virtual-routers": 3,
      "interfaces": 2,
      "statistics": {
        "discontinuity-datetime": "2016-10-24T17:11:27+02:00",
        "checksum-errors": 2,
        "version-errors": 0,
        "vrid-errors": 0,
        "ip-ttl-errors": 1
      }
    }
  }
```

Authors' Addresses

Xufeng Liu (editor)
Jabil
8281 Greensboro Drive, Suite 200
McLean VA 22102
USA

EMail: Xufeng_Liu@jabil.com

Athanasios Kyparlis
Jabil
8281 Greensboro Drive, Suite 200
McLean VA 22102
USA

EMail: Athanasios_Kyparlis@jabil.com

Ravi Parikh
VMware
3425 Hillview Avenue
Palo Alto CA 94304
USA

EMail: parikhr@vmware.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary NC 27513
USA

EMail: acee@cisco.com

Mingui Zhang
Huawei Technologies
No. 156 Beiqing Rd. Haidian District
Beijing 100095
P.R. China

EMail: zhangmingui@huawei.com