Network Working Group
Internet-Draft
Obsoletes: 8347 (if approved)
Intended status: Standards Track
Expires: 11 December 2025

A. Lindem, Ed.
Arrcus, Inc.
X. Liu
Alef Edge
A. Kyparlis
Jabil
R. Parikh
Zoox
M. Zhang
China Tower
9 June 2025

A YANG Data Model for the Virtual Router Redundancy Protocol (VRRP)
draft-ietf-rtgwg-vrrp-rfc8347bis-06

Abstract

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

The VRRP terminology has been updated conform to inclusive language guidelines for IETF technologies.

This document obsoletes RFC 8347.

Status of This Memo

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

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

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

This Internet-Draft will expire on 11 December 2025.

Copyright Notice

Table of Contents

1. Introduction

This document introduces a YANG data model [RFC6020] [RFC7950] for the Virtual Router Redundancy Protocol (VRRP) [RFC3768], [RFC9568]. 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.

The YANG module specified in this document supports both versions 2 and 3 of VRRP. VRRP version 2 (defined in [RFC3768]) supports IPv4. VRRP version 3 (defined in [RFC9568]) supports both IPv4 and IPv6.

The VRRP terminology has been updated conform to inclusive language guidelines for IETF technologies. This document obsoletes VRRP Version 3 [RFC8347].

1.1. Changes from ~~RF~~RFC 8347

The following changes have been made consistent with IETF inclusive language guidelines:

* The typedef "~~typedef~~ new-master-reason-type" was changed to "~~typedef~~ new-active-reason-type". The associated descriptive text was also updated.

* The enum "~~enum~~ not-master" was changed to "~~enum~~ not-active". The associated descriptive text was also updated.

* The descriptive text for enum "~~enum~~ preempted" was updated.

* The descriptive text for enum "~~enum~~ no-response" was updated.

* The identity "~~identity~~ vrrp-event-master-timeout" was changed to "~~identity~~ vrrp-event-active-timeout". The associated descriptive text was also updated.

* The identity "~~identity~~ vrrp-event-lower-priority-master" was changed to "~~identity~~ vrrp-event-lower-priority-active". The associated descriptive text was also updated.

* The descriptive text for identity "~~identity~~ vrrp-event-higher-priority-backup" was updated.

* The descriptive text for identity "~~identity~~ vrrp-event-owner-prempt" was updated.

* The descriptive text for identity "~~identity~~ backup" was updated.

* The identity "~~identity~~ master" was changed to "~~identity~~ active". The associated descriptive text was also updated.

* The descriptive text for container "~~container~~ preempt" was updated.

* The descriptive text for leaf "~~leaf~~ hold-time" was updated.

* The descriptive text for leaf "~~leaf~~ accept-mode" was updated.

* The leaf "~~leaf~~ master-down-interval" was changed to "~~leaf~~ active-down-interval". The associated descriptive text was also updated.

* The leaf "~~leaf~~ new-master-reason" was changed to "~~leaf~~ new-active-reason". The associated descriptive text was also updated.

* The leaf "~~leaf~~ new-master-reason" was changed to "~~leaf~~ new-active-reason". The associated descriptive text was also updated.

* The leaf "~~leaf~~ master-transitions" was changed to "~~leaf~~ active-transitions". The associated descriptive text was also updated.

* The notification "~~notification~~ vrrp-new-master-event" was changed to "~~notification~~ vrrp-new-active-event". The associated descriptive text was also updated.

* The notification leaf "master-ip-address" was changed to "active-ip-address". The associated descriptive text was also updated.

* The notification leaf "~~leaf~~ new-master-reason" was changed to "~~leaf~~ new-active-reason". The associated descriptive text was also updated.

The following additions have been made to the "ietf-vrrp" module:

* The leaf "~~leaf~~ effective-priority" was add to the grouping "~~grouping~~ vrrp-state-attributes" to reflect the effective priority

due to the VRRP router owning the address or any local policy adjusting the priority.

1.2. 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.

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

* augment

* data node

1.3. 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 [RFC8340].

1.4. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as 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	[RFC8343]
ip	ietf-ip	[RFC8344]

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 [RFC9568]. The model is designed to be implemented on a device where VRRP version 2 or 3 is implemented. With the help of a proper management protocol, the defined model can be used to:

* Configure VRRP version 2 or 3.

* Manage the protocol operational behavior.

* Retrieve the protocol operational status.

* Receive the protocol notifications.

2.2. Relationships with the Interface Model and IP Model

This model augments the interface data model "ietf-interfaces" [RFC8343] and the IP management model "ietf-ip" [RFC8344]. The augmentation relationships 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
                        ...
```

Figure 1

In ~~the above figure~~Figure 1, 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, as described in Section 6.4 of [RFC9568], 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 or addresses associated with a virtual router (described in Section 1 of [RFC9568]) 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:~~depicted in Figure 2.

```
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
```

Figure 2

The model allows the following protocol entities to be ~~configured~~managed:

* VRRP instance (version 2 or 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 network connectivity failures.

2.4. Protocol States

The model structure for the protocol states is ~~as shown below~~depicted in Figure 3.~~:~~

a mis en forme : Anglais (États-Unis)

```
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 effective-priority?             uint8
          +--ro last-adv-source?                inet:ip-address
          +--ro up-datetime?                    yang:date-and-time
          +--ro active-down-interval?           uint32
          +--ro skew-time?                      uint32
          +--ro last-event?                     identityref
          +--ro new-active-reason?              new-active-reason-type
          +--ro statistics                 // per-instance statistics
             +--ro discontinuity-datetime?    yang:date-and-time
             +--ro active-transitions?        yang:counter32
             +--ro advertisement-rcvd?        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 effective-priority?             uint8
+--ro last-adv-source?                inet:ip-address
+--ro up-datetime?                    yang:date-and-time
+--ro active-down-interval?           uint32
+--ro skew-time?                      uint32
+--ro last-event?                     identityref
+--ro new-active-reason?              new-active-reason-type
+--ro statistics                 // per-instance statistics
   +--ro discontinuity-datetime?    yang:date-and-time
   +--ro active-transitions?        yang:counter32
   +--ro advertisement-rcvd?        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
```

Figure 3

This model conforms to the Network Management Datastore Architecture (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [I-D.ietf-netmod-rfc8407bis]. 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 the retrieval of protocol states at the following levels:

* VRRP instance (version 2 or 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 network connectivity failures.

* Global states and statistics summarizing all instances.

2.5. Notifications

This model defines the following VRRP-specific notifications (Figure 4):

```
notifications:
   +---n vrrp-new-active-event
   |  +--ro active-ip-address    inet:ip-address
   |  +--ro new-active-reason    new-active-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
```

Figure 4

Each notification type is used to indicate a type of VRRP state change or error occurrence:

"vrrp-new-active-event": VRRP new active event, indicating that a new active virtual router 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 mechanisms defined in [RFC8639] and [RFC8641] can be used for other general notifications. These mechanisms currently allow the user to:

* Subscribe notifications on a per-client basis.

* Specify subtree filters or XML Path Language (XPath) filters so that only contents of interest 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 effective-priority?               uint8
          +--ro last-adv-source?                  inet:ip-address
          +--ro up-datetime?                      yang:date-and-time
          +--ro active-down-interval?             uint32
          +--ro skew-time?                        uint32
          +--ro last-event?                       identityref
          +--ro new-active-reason?
 new-active-reason-type
          +--ro statistics
             +--ro discontinuity-datetime?    yang:date-and-time
             +--ro active-transitions?        yang:counter32
             +--ro advertisement-rcvd?        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 effective-priority?             uint8
         +--ro last-adv-source?                inet:ip-address
         +--ro up-datetime?                    yang:date-and-time
         +--ro active-down-interval?           uint32
         +--ro skew-time?                      uint32
         +--ro last-event?                     identityref
         +--ro new-active-reason?
new-active-reason-type
         +--ro statistics
            +--ro discontinuity-datetime?    yang:date-and-time
            +--ro active-transitions?        yang:counter32
            +--ro advertisement-rcvd?        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-active-event
   |  +--ro active-ip-address    inet:ip-address
   |  +--ro new-active-reason    new-active-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
```

Figure 5

4. YANG Module

This module references [RFC3768], [RFC9568], and [RFC6527].

```
<CODE BEGINS> file "ietf-vrrp@2025-06-09.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;
    reference
      "RFC 6991: Common YANG Data Types, Section 4";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types, Section 3";
  }
  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";
  }

  organization
    "IETF Routing Area Working Group (RTGWG)";

  contact
    "WG Web:   <https://datatracker.ietf.org/wg/rtgwg/>
     WG List:  <mailto:rtgwg@ietf.org>

     Editor:   Acee Lindem
               <mailto:acee.ietf@gmail.com>

     Editor:   Xufeng Liu
               <mailto:xufeng.liu.ietf@gmail.com>

     Editor:   Athanasios Kyparlis
```

```
             <mailto:Athanasios_Kyparlis@jabil.com>

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

   Editor:   Mingui Zhang
             <mailto:zhangmingui@huawei.com>";
description
  "This YANG module defines a model for managing Virtual Router
   Redundancy Protocol (VRRP) versions 2 and 3.

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

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

   All revisions of IETF and IANA published modules can be found
   at the YANG Parameters registry group
   (https://www.iana.org/assignments/yang-parameters).

   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.

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

revision 2025-06-09 {
  description
    "This revision includes the following changes:
     - Update some identifiers and descriptions to conform
       to the changes in RFC 9586
     - Add a new effective-priority leaf
     - Update some reference statements";
  reference
    "RFC XXXX: A YANG Data Model for the Virtual Router Redundancy
               Protocol (VRRP)";
}
revision 2018-03-13 {
  description
    "Initial revision.";
  reference
    "RFC 8347: A YANG Data Model for the Virtual Router Redundancy
               Protocol (VRRP)";
}

/*
 * 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 interval 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-active-reason-type {
  type enumeration {
    enum not-active {
      description
        "The virtual router has never transitioned to active
         state.";
    }
    enum priority {
      description
        "Priority was higher.";
    }
    enum preempted {
      description
        "The active virtual router was preempted.";
    }
    enum no-response {
      description
        "Previous active virutalvirtual router did not respond.";
    }
  }
  description
    "Indicates why the virtual router has transitioned to
     active state.

     The identifier of this typedef is changed to to reflect
     the updated terminology used in RFC 9568.";
} // new-active-reason-type

/*
 * Identities
 */
/* vrrp-event-type identity and its derivatives. */

identity vrrp-event-type {
  description
    "Indicates 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 active.";
}

identity vrrp-event-active-timeout {
  base vrrp-event-type;
  description
    "Indicates that the current active virtual router has not sent
     an advertisement within the limit of active-down-interval.

     The identity is changed to to reflect the updated terminology
     used in RFC 9568.";
}

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-active {
  base vrrp-event-type;
  description
    "Indicates that there is a lower-priority VRRP active router.

     The identity is changed to ~~to~~ reflect the updated terminology
     used in RFC 9568.";
}

identity vrrp-event-owner-preempt {
  base vrrp-event-type;
  description
    "Indicates that the owner has preempted another virtual router
     to become the active virtual router.";
}

/* vrrp-error-global identity and its derivatives. */

identity vrrp-error-global {
  description
    "Indicates the type of a VRRP error that occurred
     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 Virtual Router Identifier
     (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
    "Indicates the type of a VRRP error that occurred
     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 interval 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
    "Indicates 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 active virtual router.";
}

identity active {
  base vrrp-state-type;
  description
    "Indicates that the virtual router is forwarding
     packets for IP addresses that are associated with
     this virtual router.

     The identity is changed to ~~to~~ reflect the updated terminology
     used in RFC 9568.";
}

/* vrrp-version identity and its derivatives. */

identity vrrp-version {
  description
    "The version of VRRP.";
}

identity vrrp-v2 {
  base vrrp-version;
  description
    "Indicates version 2 of VRRP.";

}

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

/*
 * Groupings
 */

grouping vrrp-common-attributes {
  description
    "Group of VRRP attributes common to versions 2 and 3.";
  leaf vrid {
    type uint8 {
      range "1..255";
    }
    description
```

```
          "Virtual Router ID (i.e., VRID).";
        }
        leaf version {
          type identityref {
            base vrrp:vrrp-version;
          }
          mandatory true;
          description
            "Version 2 or 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 VRRP backup router to preempt a
             lower-priority VRRP active router.";
          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 active router.";
          }
        }
        leaf priority {
          type uint8 {
            range "1..254";
          }
          default "100";
          description
            "Configures the VRRP election priority for the backup
             virtual router.";
        }
        leaf accept-mode {
          when "derived-from-or-self(current()/../version, 'vrrp:vrrp-
v3')" {
            description
              "Applicable only to version 3.";
          }
          type boolean;
          default "false";
          description
            "Controls whether a virtual router in active 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: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 VRRPv2 advertisements
             are sent from the specified interface.";
        }
      }
      case v3 {
        when "derived-from-or-self(version, 'vrrp: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 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 VRRPv2 or 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 VRRP router decrements.  When an unavailable
     interface becomes available again, the priority of the

     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 VRRPv2 or VRRPv3 router instance to track the
     specified networks through their IPv4 network prefixes.
     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 VRRP router decrements.  When an
     unavailable network is again reachable, the priority of
     the VRRP router is incremented by the same amount.";
  list network {
    key "prefix";
    description
      "Enables the specified VRRPv2 or VRRPv3 instance to
       track an IPv4 network by specifying the prefix of the
       IPv4 network.";
    leaf prefix {
      type inet:ipv4-prefix;
      description
```

```
            "The IPv4 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
             VRRP router if there is a failure in the IPv4
             network.";
        }
      } // network
    } // networks
  } // track
  container virtual-ipv4-addresses {
    description
      "Configures the virtual IPv4 address for the
       VRRP interface.";
    list virtual-ipv4-address {
      key "ipv4-address";
      max-elements 16;
      description
        "Virtual IPv4 addresses for a single VRRP instance.  For a
         VRRP owner router, the virtual address must match one
         of the IPv4 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 9568: Virtual Router Redundancy Protocol (VRRP)
                     Version 3 for IPv4 and IPv6,  Section 1.3";
      }
    } // virtual-ipv4-address
  } // virtual-ipv4-addresses
} // 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
      "Includes the interval that 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 VRRPv2 or 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 VRRP router decrements.  When an unavailable
     interface becomes available again, the priority of the
     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 VRRPv2 or VRRPv3 router instance to track the
     specified networks through their IPv6 network prefixes.
     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 VRRP router decrements.  When an
     unavailable network is again reachable, the priority of
     the VRRP router is incremented by the same amount.";
  list network {
    key "prefix";
    description
      "Enables the specified VRRPv2 or VRRPv3 instance to
       track an IPv6 network by specifying the prefix of the
       IPv6 network.";
    leaf prefix {
      type inet:ipv6-prefix;
      description
        "The IPv6 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
             VRRP router if there is a failure in the IPv6
             network.";
        }
      } // network
    } // networks
  } // track
  container virtual-ipv6-addresses {
    description
      "Configures the virtual IPv6 address for the
       VRRP interface.";
    list virtual-ipv6-address {
      key "ipv6-address";
      max-elements 2;
      description
        "Two IPv6 addresses are allowed.  The first address must
         be a link-local address.  The second address 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 9568: Virtual Router Redundancy Protocol (VRRP)
                     Version 3 for IPv4 and IPv6, Section 1.4";
      }
    } // virtual-ipv6-address
  } // virtual-ipv6-addresses
} // 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 the owner.";
  }
  leaf effective-priority {
    type uint8 {
      range "1..255";
    }
    config false;
```

```
  description
    "The effect priority of the virtual router taking account
     address ownership and any modifications due to
     local policy.";
}
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 active-down-interval {
  type uint32;
  units "centiseconds";
  config false;
  description
    "Time interval for the backup virtual router to declare
     'active down'.

     The identifier of this leaf is changed to to reflect
     the updated terminology used in RFC 9568.";
}
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-active-reason {
  type new-active-reason-type;
  config false;
  description
    "Indicates why the virtual router has transitioned to
     active state.

     The identifier of this leaf is changed to to reflect
     the updated terminology used in RFC 9568.";
}
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 statistics 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 active-transitions {
  type yang:counter32;
  description
    "The total number of times that this virtual router's
     state has transitioned to 'Active'.

     The identifier of this leaf is changed to to reflect
     the updated terminology used in RFC 9568.";
}
leaf advertisement-rcvd {
  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
     interval 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.";
    }
  } // statistics
} // 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 interfaces 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 any
         one or more of checksum-errors, version-errors,
         vrid-errors, or 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 9568: Virtual Router Redundancy Protocol (VRRP)
                   Version 3 for IPv4 and IPv6, 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 9568: Virtual Router Redundancy Protocol (VRRP)
                   Version 3 for IPv4 and IPv6, 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 9568: Virtual Router Redundancy Protocol (VRRP)
                   Version 3 for IPv4 and IPv6, 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 (IPv4) or Hop Limit (IPv6)
         not equal to 255.";
      reference
        "RFC 9568: Virtual Router Redundancy Protocol (VRRP)
                   Version 3 for IPv4 and IPv6,
                   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
    "Augments IPv4 interface.";
  container vrrp {
    description
      "Configures VRRP version 2 or 3 for IPv4.";
    list vrrp-instance {
      key "vrid";
      description
        "Defines a virtual router, identified by a VRID, within the
         IPv4 address space.";
      uses vrrp-ipv4-attributes;
      uses vrrp-state-attributes;
    }
  }
} // augments ipv4

augment "/if:interfaces/if:interface/ip:ipv6" {
  description
    "Augments IPv6 interface.";
  container vrrp {
    description
      "Configures 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 VRID, within the
         IPv6 address space.";
      uses vrrp-ipv6-attributes;
      uses vrrp-state-attributes;
    }
  }
} // augments ipv6

container vrrp {
  config false;
  description
    "VRRP data at the global level.";
  uses vrrp-global-state-attributes;
}

/*
 * Notifications
 */

notification vrrp-new-active-event {
  description
    "Notification event for the election of a new VRRP
     active router.

     The identifier of the notification is changed to
     reflect the updated terminology used in RFC 9568.";
  leaf active-ip-address {
    type inet:ip-address;
    mandatory true;
    description
      "IPv4 or IPv6 address of the new VRRP active router.

       The identifier of the leaf is changed to
       reflect the updated terminology used in RFC 9568.";
  }
  leaf new-active-reason {
    type new-active-reason-type;
    mandatory true;
    description
      "Indicates why the virtual router has transitioned to
       active state.

       The identifier of the leaf is changed to
       reflect the updated terminology used in RFC 9568.";
  }
}

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 that happened on a
     virtual router.";
  leaf interface {
    type if:interface-ref;
    mandatory true;
    description
      "Indicates the interface on which the event has occurred.";
  }

  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>
```

~~Figure 6~~

5. IANA Considerations

IANA is requested to update the following registration in the "ns" registry within the "IETF XML Registry" group [RFC3688] to reference this document:

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

IANA is requested to register the following YANG module in the "YANG Module Names" registry [RFC6020] within the "YANG Parameters" registry group.

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

6. Operations and ~~Managment~~Management Considerations

This document defines ~~the~~ a YANG data model [RFC7950] for the VRRP protocol [RFC9568]. Hence, the entire document can be considered devoted to operations and management.

6.1. Migration Path

The changes to use inclusive language (Section 1.1)~~Section 1.1~~ are non-backward compatibly based on the rules in [RFC7950]. However, there are no

exant existing implementations of the "ietf-vrrp" YANG module based on numerournumerous implementation polls. Hence, it was agreed upon to change the YANG data nodes with non-inclusive language in a single iteration rather than going through the deprecation, obsoletion, and removal cycle.

7. Security Considerations

This section is modeled after the template described in Section 3.7 of [I-D.ietf-netmod-rfc8407bis].

The "ietf-vrrp" YANG module defines a data model that is designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. These YANG-based protocols (1) have to use a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and (2) have to use mutual authentication.

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 a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., "config true", which is the default). All writable data nodes are likely to be reasonably sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) and delete operations to these data nodes without proper protection or authentication can have a negative effect on network operations. The following subtrees and data nodes have particular sensitivities/vulnerabilities:

/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. Specifically, the following subtrees and data nodes have particular sensitivities/ vulnerabilities:

/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.

8. Acknowledgments

Thanks to Nicola Serafini for his suggestion to add effective-priority to the virtual router operational state.

Thanks to Med Boucadair for an extensive review and suggested edits.

Thanks to Carlos Pignataro for review and comments.

9. 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>.

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

[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>.

[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>.

[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>.

[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>.

[RFC8347] Liu, X., Ed., Kyparlis, A., Parikh, R., Lindem, A., and M.

Zhang, "A YANG Data Model for the Virtual Router Redundancy Protocol (VRRP)", RFC 8347, DOI 10.17487/RFC8347, March 2018, <https://www.rfc-editor.org/info/rfc8347>.

[RFC9568] Lindem, A. and A. Dogra, "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 9568, DOI 10.17487/RFC9568, April 2024, <https://www.rfc-editor.org/info/rfc9568>.

10. Informative References

[I-D.ietf-netmod-rfc8407bis] Bierman, A., Boucadair, M., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", Work in Progress, Internet-Draft, draft-ietf-netmod-rfc8407bis-28, 5 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-rfc8407bis-28>.

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <https://www.rfc-editor.org/info/rfc4252>.

[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>.

[RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", RFC 7224, DOI 10.17487/RFC7224, May 2014, <https://www.rfc-editor.org/info/rfc7224>.

[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>.

[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>.

[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>.

[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>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

Appendix A. Data Tree Example

This section contains an example of an instance data tree in JSON encoding [RFC7951], containing both configuration and state data. (This example includes "iana-if-type", which is initially defined in [RFC7224].)

Commenté [MB1]: As the authoritative version is the recent one in the IANA website. Better to cite https://www.iana.org/assignments/iana-if-type/iana-if-type.xhtml, per 8407bs reco.

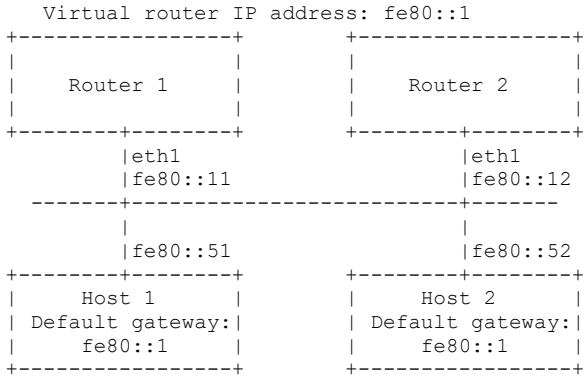

Figure 7

The configuration instance data for Router 1 in ~~the above figure~~Figure 7 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": "ietf-vrrp:vrrp-v3",
                "priority": 200,
                "advertise-interval-centi-sec": 50,
                "virtual-ipv6-addresses": {
                  "virtual-ipv6-address": [
                    "ipv6-address": "fe80::1"
                  ]
                }
              }
            ]
          }
        }
      }
    ]
  }
}
```

Figure 8

The corresponding 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": "ietf-vrrp: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": "active",
              "is-owner": false,
              "effective-priority": 200,
              "last-adv-source": "fe80::11",
              "up-datetime": "2016-10-24T17:11:27+02:00",
              "active-down-interval": 161,
              "skew-time": 11,
              "last-event": "vrrp-event-interface-up",
              "new-active-reason": "priority",
              "statistics": {
                "discontinuity-datetime":
                  "2016-10-24T17:11:27+02:00",
                "active-transitions": 2,
                "advertisement-rcvd": 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
        }
      }
    }
```

Figure 9

Authors' Addresses

Acee Lindem (editor)
Arrcus, Inc.
301 Midenhall Way
Cary, NC 27513
United States of America
Email: acee.ietf@gmail.com

Xufeng Liu
Alef Edge
United States of America
Email: xufeng.liu.ietf@gmail.com

Athanasios Kyparlis
Jabil
8281 Greensboro Drive, Suite 200
McLean, VA 22102
United States of America
Email: Athanasios_Kyparlis@jabil.com

Ravi Parikh
Zoox
United States of America
Email: parikh_ravi@ymail.com

Mingui Zhang
China Tower
9 Dongranbeijie Haidian District
Beijing
China
Email: zhangmg3@chinatowercom.cn