

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 5, 2016

X. Liu, Editor
A. Kyparlis
R. Parikh
Ericsson
A. Lindem
Cisco Systems
M. Zhang
Huawei Technologies
April 5, 2016

A YANG Data Model for Virtual Router Redundancy Protocol (VRRP)
draft-liu-rtgwg-yang-vrrp-04.txt

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on October 5, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction.....	2
1.1. Terminology.....	2
2. VRRP YANG model overview.....	3
3. VRRP YANG module.....	7
4. Security Considerations.....	28
5. Contributors.....	28
6. References.....	29
6.1. Normative References.....	29
6.2. Informative References.....	29

1. Introduction

This document introduces a YANG [RFC6020] 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 keywords "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].

The following terms are defined in [RFC6020] and are not redefined here:

- o augment

- o data model
- o data node

2. VRRP YANG model overview

This document defines the YANG module "ietf-vrrp", which has the following structure:

```
module: ietf-vrrp
augment /if:interfaces/if:interface/ip:ipv4:
  +--rw vrrp
    +--rw vrrp-instance* [vrid]
      +--rw vrid                               uint8
      +--rw version?                           enumeration
      +--rw log-state-change?                  boolean
      +--rw preempt!
        | +--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* [network]
        | | | +--rw network                  inet:ipv4-prefix
        | | | +--rw priority-decrement?      uint8
      +--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 version?                           enumeration
```

```

    +--rw log-state-change?                boolean
    +--rw preempt!
    |   +--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* [network]
    |   |   |   +--rw network            inet:ipv6-prefix
    |   |   |   +--rw priority-decrement?  uint8
    +--rw virtual-ipv6-addresses
    |   +--rw virtual-ipv6-address* [ipv6-address]
    |   +--rw ipv6-address    inet:ipv6-address
augment /if:interfaces-state/if:interface/ip:ipv4:
    +--ro vrrp
    |   +--ro vrrp-instance* [vrid]
    |   |   +--ro vrid                uint8
    |   |   +--ro version?            enumeration
    |   |   +--ro log-state-change?    boolean
    |   |   +--ro preempt!
    |   |   |   +--ro hold-time?    uint16
    |   |   +--ro priority?                uint8
    |   |   +--ro accept-mode?            boolean
    |   |   +--ro (advertise-interval-choice)?
    |   |   |   +--:(v2)
    |   |   |   |   +--ro advertise-interval-sec?    uint8
    |   |   |   +--:(v3)
    |   |   |   |   +--ro advertise-interval-centi-sec?    uint16
    |   +--ro track
    |   |   +--ro interfaces
    |   |   |   +--ro interface* [interface]
    |   |   |   |   +--ro interface            if:interface-ref
    |   |   |   |   +--ro priority-decrement?  uint8
    |   |   +--ro networks
    |   |   |   +--ro network* [network]
    |   |   |   |   +--ro network            inet:ipv4-prefix

```

```

|         +--ro priority-decrement?    uint8
+--ro virtual-ipv4-addresses
|   +--ro virtual-ipv4-address* [ipv4-address]
|   |   +--ro ipv4-address    inet:ipv4-address
+--ro state?                      identityref
+--ro is-owner?                    boolean
+--ro last-adv-source?              inet:ip-address
+--ro up-time?                     yang:date-and-time
+--ro master-down-interval?        uint32
+--ro skew-time?                   uint32
+--ro last-event?                  string
+--ro new-master-reason?            new-master-reason-type
+--ro statistics
|   +--ro discontinuity-time?        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-state/if:interface/ip:ipv6:
+--ro vrrp
|   +--ro vrrp-instance* [vrid]
|   |   +--ro vrid                    uint8
|   |   +--ro version?                enumeration
|   |   +--ro log-state-change?        boolean
|   |   +--ro preempt!
|   |   |   +--ro hold-time?    uint16
|   |   +--ro priority?            uint8
|   |   +--ro accept-mode?          boolean
|   |   +--ro advertise-interval-centi-sec?    uint16
|   +--ro track
|   |   +--ro interfaces
|   |   |   +--ro interface* [interface]
|   |   |   |   +--ro interface    if:interface-ref
|   |   |   |   +--ro priority-decrement?    uint8

```

```

|   +--ro networks
|   |   +--ro network* [network]
|   |   |   +--ro network          inet:ipv6-prefix
|   |   |   +--ro priority-decrement?  uint8
+--ro virtual-ipv6-addresses
|   +--ro virtual-ipv6-address* [ipv6-address]
|   |   +--ro ipv6-address      inet:ipv6-address
+--ro state?                      identityref
+--ro is-owner?                   boolean
+--ro last-adv-source?            inet:ip-address
+--ro up-time?                   yang:date-and-time
+--ro master-down-interval?      uint32
+--ro skew-time?                 uint32
+--ro last-event?                string
+--ro new-master-reason?         new-master-reason-type
+--ro statistics
|   +--ro discontinuity-time?      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-state:
+--ro vrrp-global
|   +--ro virtual-routers?         uint32
|   +--ro interfaces?              uint32
|   +--ro checksum-errors?         yang:counter64
|   +--ro version-errors?          yang:counter64
|   +--ro vrid-errors?             yang:counter64
|   +--ro ip-ttl-errors?           yang:counter64
|   +--ro global-statistics-discontinuity-time? yang:date-and-
time
notifications:
+---n vrrp-new-master-event
|   +--ro master-ipaddr?          inet:ipv4-address

```

```
|  +--ro new-master-reason?  new-master-reason-type
+---n vrrp-protocol-error-event
|  +--ro protocol-error-reason?  enumeration
+---n vrrp-virtual-router-error-event
    +--ro interface?            if:interface-ref
    +--ro ip-version?           enumeration
    +--ro vrid-v4?              leafref
    +--ro vrid-v6?              leafref
    +--ro virtual-router-error-reason?  enumeration
```

3. VRRP YANG module

```
<CODE BEGINS> file "ietf-vrrp@2015-09-28.yang"
module ietf-vrrp {
  namespace "urn:ietf:params:xml:ns:yang:ietf-vrrp";
  // replace with IANA namespace when assigned
  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 "TBD";
  contact "TBD";
  description
    "This YANG module defines a model for managing Virtual Router
    Redundancy Protocol (VRRP) version 2 and version 3.";

  revision "2015-09-28" {
```

```
description "Initial revision";
reference
  "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 master-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
 */

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.";
```

```
}

/*
 * 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 enumeration {
      enum 2 {
        description "VRRP version 2.";
      }
      enum 3 {
        description "VRRP version 3.";
      }
    }
    description "Version 2 or version 3 of VRRP.";
  }

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

  container preempt {
    presence "Present if preempt is enabled.";
    description
      "Enables a higher priority Virtual Router Redundancy
       Protocol (VRRP) backup router to preempt a lower priority
```

```
        VRRP master.";
    leaf hold-time {
        type uint16;
        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.";
}
} // vrrp-common-attributes

grouping vrrp-v3-attributes {
    description
        "Group of VRRP versin 3 attributes.";

    leaf accept-mode {
        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.";
    }
}
```

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

  uses vrrp-common-attributes;

  uses vrrp-v3-attributes {
    when "version = 3" {
      description "Applicable only to version 3.";
    }
  }

  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 "version = 2" {
        description "Applicable only to version 2.";
      }
      leaf advertise-interval-sec {
        type uint8 {
          range 1..254;
        }
        default 1;
        description
          "Configures the interval that Virtual Router
          Redundancy Protocol Version 2 (VRRPv2) advertisements
          are sent from the specified interface.";
      }
    }

    case v3 {
      when "version = 3" {
        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
            an interface.";

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

            leaf interface {
                type if:interface-ref;
                must "../../../../../../../../../../../ipv4" {
                    description "Interface is IPv4.";
                }
            }
            description
                "Interface to track.";
        }

        leaf priority-decrement {
            type uint8 {
                range 1..254;
            }
            description
                "Specifies how much to decrement the priority of the
```

```
        VRRP instance if the interface goes down.";
    }
} // track-interface
} // track-interfaces

container networks {
  description
    "Enables the backup Virtual Router Redundancy Protocol
    version 2 (VRRP) or version 3 (VRRPv3) router to track a
    specified network through the IP network prefix of that
    network.";
  list network {
    key "network";
    description
      "Enables the specified Virtual Router Redundancy
      Protocol version 2 (VRRP) or version 3 (VRRPv3)
      instance to track an interface.";

    leaf network {
      type inet:ipv4-prefix;
      description
        "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
      "Virtual IPv4 address.";
  }
} // 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;

  uses vrrp-v3-attributes {
    when "version = 3" {
      description "Uses VRRP version 3 attributes.";
    }
  }
} // uses vrrp-v3-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
                an interface.";
            list interface {
                key "interface";
                description
                    "Interface to track.";

                leaf interface {
                    type if:interface-ref;
                    must "../../../..../ipv6" {
                        description "Interface is IPv6.";
                    }
                }
                description
                    "Interface to track.";
            }

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

    container networks {
        description
            "Enables the backup Virtual Router Redundancy Protocol
            version 2 (VRRP) or version 3 (VRRPv3) router to track a
```



```
    specified network through the IP network prefix of that
    network.";
list network {
  key "network";
  description
    "Enables the specified Virtual Router Redundancy
    Protocol version 2 (VRRP) or version 3 (VRRPv3)
    instance to track an interface.";

  leaf network {
    type inet:ipv6-prefix;
    description
      "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
                "Virtual IPv6 address.";
        }
    } // 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-state-type;
        }
        description
            "Operational state.";
    }

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

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

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

    leaf master-down-interval {
```

```
    type uint32;
    units centiseconds;
    description
        "Time interval for backup virtual router to declare
        Master down.";
}

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

leaf last-event {
    type string;
    description
        "Last reported event.";
}

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

container statistics {
    description
        "VRRP statistics.";

    leaf discontinuity-time {
        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.";
}

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.";
}

leaf global-statistics-discontinuity-time {
    type yang:date-and-time;
    description
        "The time on the most recent occasion at which one of
```

router-checksum-errors, router-version-errors,
router-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 object will be 0.";

```
    }  
  } // vrrp-global-state-attributes  
  
/*  
 * Configuration 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;  
    }  
  }  
} // 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 "version = 3" {
    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;
} // list vrrp-instance
} // container vrrp
} // augment ipv6

/*
 * Operational state data nodes
 */

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

  container vrrp {
    description
      "State information for Virtual Router Redundancy Protocol
      (VRRP) version 2 for IPv4.";

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

      uses vrrp-ipv4-attributes;
      uses vrrp-state-attributes;
    } // list vrrp-instance
  }
}

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



```
container vrrp {
  description
    "State information of the Virtual Router Redundancy Protocol
    (VRRP) version 2 or version 3 for IPv6.";

  list vrrp-instance {
    key vrid;
    description
      "States of 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
}

augment "/if:interfaces-state" {
  description "Specify VRRP state data at the global level.";

  container vrrp-global {
    description
      "State information of the Virtual Router Redundancy Protocol
      (VRRP) 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-ipaddr {
    type inet:ipv4-address;
    description
      "IPv4 or IPv6 address of the new master.";
  }
}
```

```
    }
    leaf new-master-reason {
        type new-master-reason-type;
        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 enumeration {
            enum checksum-error {
                description
                    "A packet has been received with an invalid VRRP checksum
                    value.";
            }
            enum version-error {
                description
                    "A packet has been received with an unknown or
                    unsupported version number.";
            }
            enum vrid-error {
                description
                    "A packet has been received with a VRID that is not valid
                    for any virtual router on this router.";
            }
            enum ip-ttl-error {
                description
                    "A packet has been received with IP TTL (Time-To-Live)
                    not equal to 255.";
            }
        }
        description
            "Indicates the reason for the protocol error.";
    }
}

notification vrrp-virtual-router-error-event {
```

```
description
  "Notification event for a error happened on a virtual router.";
leaf interface {
  type if:interface-ref;
  description
    "Indicates the interface for which statistics area
    to be cleared.";
}
leaf ip-version {
  type enumeration {
    enum 4 {
      description "IPv4";
    }
    enum 6 {
      description "IPv6";
    }
  }
  description "Indicates the IP version.";
}
leaf vrid-v4 {
  type leafref {
    path "/if:interfaces/if:interface"
      + "[if:name = current()/../interface]/ip:ipv4/vrrp/"
      + "vrrp-instance/vrid";
  }
  description
    "Indicates the virtual router on which the event has
    occured.";
}
leaf vrid-v6 {
  type leafref {
    path "/if:interfaces/if:interface"
      + "[if:name = current()/../interface]/ip:ipv6/vrrp/"
      + "vrrp-instance/vrid";
  }
  description
    "Indicates the virtual router on which the event has
    occured.";
}
leaf virtual-router-error-reason {
```

```
type enumeration {
  enum interval-error {
    description
      "A packet has been received with an advertisement
      interval different than the one configured for the local
      virtual router";
  }
  enum address-list-error {
    description
      "A packet has been received with an address list that
      does not match the locally configured address list for
      the virtual router.";
  }
  enum packet-length-error {
    description
      "A packet has been received with a packet length less
      than the length of the VRRP header.";
  }
}
description
  "Indicates the reason for the virtual router error.";
}
}
}
<CODE ENDS>
```

4. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The data-model by itself does not create any security implications. The security considerations for the NETCONF protocol are applicable. The NETCONF protocol used for sending the data supports authentication and encryption.

5. Contributors

Yuyang Xie
Huawei Technologies
No. 156 Beiqing Rd. Haidian District
Beijing 100095
P.R. China

Email: xieyuyang@huawei.com

6. References

6.1. Normative References

- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.
- [RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.
- [RFC2338] Knight, S., Weaver, D., Whipple, D., Hinden, R., Mitzel, D., Hunt, P., Higginson, P., Shand, M., and A. Lindem, "Virtual Router Redundancy Protocol", RFC 2338, April 1998.
- [RFC2787] Jewell, B. and D. Chuang, "Definitions of Managed Objects for the Virtual Router Redundancy Protocol", RFC 2787, March 2000.
- [RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, March 2010.
- [RFC6527] Tata, K., Ed., "Definitions of Managed Objects for the Virtual Router Redundancy Protocol Version 3 (VRRPv3)", RFC 6527, March 2012.

6.2. Informative References

- [RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, January 2011.

Authors' Addresses

Xufeng Liu (Editor)
Ericsson
1595 Spring Hill Road, Suite 500
Vienna, VA 22182
USA

Email: xliu@kuatrotech.com

Athanasios Kyparlis
Ericsson
1595 Spring Hill Road, Suite 500
Vienna, VA 22182
USA

Email: athanasios.kyparlis@ericsson.com

Ravi Parikh
Ericsson
300 Holger Way
San Jose, CA 95134
USA

Email: ravi.parikh@ericsson.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

