

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 3, 2015

X. Liu
A. Kyparlis
Ericsson
December 3, 2014

A YANG Data Model for Virtual Router Redundancy Protocol (VRRP)
draft-liu-rtgwg-yang-vrrp-00.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 January 1, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft [draft-liu-rtgwg-yang-vrrp-00.txt](#)

December 2014

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.

Table of Contents

1.	Introduction.....	2
1.1.	Terminology.....	2
2.	VRRP YANG model overview.....	3
3.	VRRP YANG module.....	6
4.	Security Considerations.....	21
5.	References.....	21
5.1.	Normative References.....	21
5.2.	Informative References.....	21

[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 (advertise-interval-choice)?
        | +--:(v2)
        | | +--rw advertise-interval-sec?      uint8
        | +--:(v3)
        |   +--rw advertise-interval-centi-sec? uint16
      +--rw is-owner?                           boolean
      +--rw init-wait?                           uint16
      +--rw log-state-change?                     boolean
      +--rw preempt!
        | +--rw hold-time?      uint16
      +--rw priority?                           uint8
      +--rw accept-mode?                         boolean
      +--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 (advertise-interval-choice)?
        | +--:(v2)
        | | +--rw advertise-interval-sec?      uint8
        | +--:(v3)
        |   +--rw advertise-interval-centi-sec? uint16
      +--rw is-owner?                           boolean
      +--rw init-wait?                           uint16

```

```

+---rw log-state-change?          boolean
+---rw preempt!
|   +---rw hold-time?    uint16
+---rw priority?                  uint8

```

```

+---rw accept-mode?                boolean
+---rw virtual-ipv4-addresses
    +---rw virtual-ipv4-address* [ipv4-address]
        +---rw ipv4-address    inet:ipv4-address
augment /if:interfaces-state/if:interface/ip:ipv4:
+--ro vrrp
    +--ro vrrp-instance* [vrid]
        +--ro vrid                uint8
        +--ro version?            enumeration
        +--ro (advertise-interval-choice)?
            | +--:(v2)
            | | +--ro advertise-interval-sec?    uint8
            | +--:(v3)
            | | +--ro advertise-interval-centi-sec?    uint16
        +--ro is-owner?            boolean
        +--ro init-wait?            uint16
        +--ro log-state-change?    boolean
        +--ro preempt!
            | +--ro hold-time?    uint16
        +--ro priority?            uint8
        +--ro accept-mode?        boolean
        +--ro virtual-ipv4-addresses
            | +--ro virtual-ipv4-address* [ipv4-address]
            | +--ro ipv4-address    inet:ipv4-address
        +--ro state?                enumeration
        +--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 statistics
            +--ro master-transitions?    yang:counter32
            +--ro new-master-reason?    enumeration
            +--ro advertisement-recv?    yang:counter64
            +--ro advertisement-sent?    yang:counter64
            +--ro interval-errors?    yang:counter64
            +--ro ip-ttl-errors?        yang:counter64
            +--ro riority-zero-pkts-rcvd?    yang:counter64

```

```

+---ro riority-zero-pkts-sent?      yang:counter64
+---ro invalid-type-pkts-rcvd?     yang:counter64
+---ro address-list-errors?        yang:counter64

```

```

+---ro packet-length-errors?      yang:counter64
+---ro wrong-owner-errors?        yang:counter64
+---ro self-orig-pkt-recv?        yang:counter64
augment /if:interfaces-state/if:interface/ip:ipv6:
+---ro vrrp
+---ro vrrp-instance* [vrid]
+---ro vrid                        uint8
+---ro version?                    enumeration
+---ro (advertise-interval-choice)?
| +---:(v2)
| | +---ro advertise-interval-sec?  uint8
| +---:(v3)
|   +---ro advertise-interval-centi-sec?  uint16
+---ro is-owner?                    boolean
+---ro init-wait?                    uint16
+---ro log-state-change?            boolean
+---ro preempt!
| +---ro hold-time?  uint16
+---ro priority?                    uint8
+---ro accept-mode?                boolean
+---ro virtual-ipv6-addresses
| +---ro virtual-ipv6-address* [ipv6-address]
|   +---ro ipv6-address  inet:ipv6-address
+---ro state?                        enumeration
+---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 statistics
+---ro master-transitions?          yang:counter32
+---ro new-master-reason?           enumeration
+---ro advertisement-recv?          yang:counter64
+---ro advertisement-sent?          yang:counter64
+---ro interval-errors?            yang:counter64
+---ro ip-ttl-errors?              yang:counter64
+---ro riority-zero-pkts-rcvd?      yang:counter64
+---ro riority-zero-pkts-sent?      yang:counter64
+---ro invalid-type-pkts-rcvd?      yang:counter64

```

```
+++ro address-list-errors?      yang:counter64
+++ro packet-length-errors?    yang:counter64
```

```
+++ro wrong-owner-errors?      yang:counter64
+++ro self-orig-pkt-recv?      yang:counter64

rpcs:
  +---x clear-vrrp-statistics
    +--ro input
      +--ro clear-type?         enumeration
      +--ro interface?         if:interface-ref
      +--ro ip-version?         enumeration
      +--ro vrid-v4?            leafref
      +--ro vrid-v6?            leafref
```

[3.](#) VRRP YANG module

```
<CODE BEGINS> file "ietf-vrrp@2014-11-24.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.";
}
```

Internet-Draft [draft-liu-rtgwg-yang-vrrp-00.txt](#)

December 2014

```
revision "2014-11-24" {
  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).";
}

grouping vrrp-common-attributes {
  description
    "Goup 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.";
  }

  choice advertise-interval-choice {
    description
      "The options of 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 at which 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 at which Virtual Router
       Redundancy Protocol version 3 (VRRPv3) advertisements
       are sent from the specified interface.";
  }
}
} // advertise-interval-choice

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


Internet-Draft [draft-liu-rtgwg-yang-vrrp-00.txt](#)

December 2014

```
leaf init-wait {
  type uint16 {
    range 1..3600;
  }
  units seconds;
  description
    "Time, in seconds, that a backup router in Initialize state
    delays before transitioning to Backup state.";
}

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
    "Goup of VRRP versin 3 attributes.";

  leaf accept-mode {
    type boolean;
    description
      "Configures a virtual router in Master state to accept
       packets addressed to the IPv4 or IPv6 address of the owner
       even if it is not the IPv4 or IPv6 address owner.";
  }
}

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

  uses vrrp-common-attributes;

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

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 interface IP addresses on which the owner VRRP
       is configured.";
  }
}
```

```
        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
        "Goup 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

    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 IPv4 address.";
            }
        } // virtual-ipv6-address
    } // virtual-ipv6-addresses
} // grouping vrrp-ipv6-attributes
```

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

  leaf state {
    type enumeration {
      enum none {
        description
          "Unknown state.";
      }
      enum init {
        description
          "Not initialized. Not ready to process events.";
      }
      enum wait {
        description
          "Indicates that all the virtual router is waiting
          for a startup event.";
      }
      enum backup {
        description
          "Indicates the virtual router is monitoring the
          availability of the master router.";
      }
      enum master {
        description
          "indicates that the virtual router is forwarding
          packets for IP addresses that are associated with
          this router.";
      }
    }
    description
      "Operational state.";
  }

  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 oper-state
    transitioned out of init state.";
}

leaf master-down-interval {
  type uint32;
  units centiseconds;
  description
    "Time interval for Backup to declare Master down.";
}

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

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

container statistics {
  description
    "VRRP statistics.";

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

  leaf new-master-reason {
```

```
type enumeration {
  enum not-master {
    description
      "The virtual router was 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 response.";
  }
}
description
  "Indicates the reason for the virtual router to transition
  to master state.";
}

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 {
  type yang:counter64;
  description
    "The total number of VRRP advertisement packets
    received for which the advertisement interval is
    different than the one configured for the local
```

```
        virtual router";
    }

    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";
    }

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

    leaf riority-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 {
        type yang:counter64;
        description
            "The total number of packets received for which the
            address list does not match the locally configured
            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.";
  }

  leaf wrong-owner-errors {
    type yang:counter64;
    description
      "The number of VRRP packets received by the virtual
      router with an invalid owner.";
  }

  leaf self-orig-pkt-recv {
    type yang:counter64;
    description
      "The number of VRRP packets received by the virtual
      router originated by itself.";
  }
} // container statistics
} // grouping vrrp-state-attributes

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

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

    list vrrp-instance {
      key vrid;
      description
        "Defines a virtual router, identified by a virtual router
         identifier (VRID), within IPv6 address space.";

      uses vrrp-ipv4-attributes;
    } // list vrrp-instance
  } // container vrrp
} // augment ipv6

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

  container vrrp {
    description
      "State information of the 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
}

rpc clear-vrrp-statistics {
  description
    "Clears Virtual Router Redundancy Protocol (VRRP) statistics.";

  input {
    description
      "Specify the information to be cleared";

    leaf clear-type {
      type enumeration {
        enum all {
          description
            "Clears all VRRP statistics.";
        }
        enum global {
          description
            "Clears global VRRP statistics.";
        }
        enum all-interface {
          description
            "Clears VRRP statistics for all interfaces.";
        }
        enum interface {
```

```
        description
            "Clears VRRP statistics for the specified interface.";
    }
    enum virtual-router {
        description
            "Clears VRRP statistics for the specified virtual
            router.";
    }
    enum scheduler {
        description "Clear VRRP scheduler statistics.";
    }
}
description
    "Specify the type of information to be cleared.";
}

leaf interface {
    when "../clear-type = interface or "
        + "../clear-type = virtual-router" {
        description
            "Valid only when clear-type is interface.";
    }
    type if:interface-ref;
    description
        "Specify the interface whose statistics to be cleared.";
}

leaf ip-version {
    when "../clear-type = virtual-router" {
        description
            "Valid only when clear-type is virtual-router.";
    }
    type enumeration {
        enum 4 {
            description "IPv4";
        }
        enum 6 {
            description "IPv6";
        }
    }
    description "Specify the IP version.";
}
```

```
}

leaf vrid-v4 {
  when "'../clear-type = virtual-router' and"
    + "'../ip-version = 4'" {
    description
      "Valid only when clear-type is virtual-router.";
  }
  type leafref {
    path "/if:interfaces/if:interface"
      + "[if:name = current()../interface]/ip:ipv4/vrrp/"
      + "vrrp-instance/vrid";
  }
  description
    "Specify the virtual router whose statistics to be
    cleared.";
}

leaf vrid-v6 {
  when "'../clear-type = virtual-router' and"
    + "'../ip-version = 4'" {
    description
      "Valid only when clear-type is virtual-router.";
  }
  type leafref {
    path "/if:interfaces/if:interface"
      + "[if:name = current()../interface]/ip:ipv6/vrrp/"
      + "vrrp-instance/vrid";
  }
  description
    "Specify the virtual router whose statistics to be
    cleared.";
}
} // input
} // rpc clear-vrrp-statistics
}
<CODE ENDS>
```

[4.](#) Security Considerations

The abstract protocol used for sending the topology data MUST support authentication and SHOULD support encryption. The data-model by itself does not create any security implications.

[5.](#) References

[5.1.](#) 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.

[5.2.](#) Informative References

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

Internet-Draft

[draft-liu-rtgwg-yang-vrrp-00.txt](#)

December 2014

Authors' Addresses

Xufeng Liu
Ericsson

Email: xufeng.liu@ericsson.com

Athanasios Kyparlis
Ericsson

Email: athanasios.kyparlis@ericsson.com

Liu

Expires June 3, 2015

[Page 22]