

Network Working Group
Request for Comments: 2787
Category: Standards Track

B. Jewell
Copper Mountain Networks, Inc.
D. Chuang
CoSine Communications
March 2000

Definitions of Managed Objects for the
Virtual Router Redundancy Protocol

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

Abstract

This specification defines an extension to the Management Information Base (MIB) for use with SNMP-based network management. In particular, it defines objects for configuring, monitoring, and controlling routers that employ the Virtual Router Redundancy Protocol (VRRP) [17].

This memo specifies a MIB module in a manner that is compliant with SMIV2 [5], and semantically identical to the SMIV1 definitions [2].

Table of Contents

1	The SNMP Network Management Framework	2
2	Overview	3
2.1	VRRP MIB Structure	3
2.2	Virtual Router Redundancy Protocol	4
2.3	VRRP MIB Table Design	4
2.3.1	Relation to Interface Group	5
2.4	VRRP Scenarios	5
2.4.1	Scenario #1	5
2.4.2	Scenario #2	8
3	Definitions	11
4	Security Considerations	27
5	Acknowledgements	28
6	References	28
7	Authors' Addresses	30
8	Intellectual Property Statement.....	30
9	Full Copyright Statement.....	31

1. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2571 [1].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIV2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].

- o A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [16].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

2. Overview

This memo identifies the set of objects for configuring, monitoring, and controlling the Virtual Router Redundancy Protocol (VRRP), as defined in RFC 2338 [17].

VRRP specifies an election protocol that will allow one or more associated IP addresses to be assumed by another router in the event of a failure of the IP address(es) owner. Thus, IP traffic from a host using a failed router as a default gateway is transparently forwarded by the VRRP router that has assumed control. VRRP provides redundancy in routed networks without requiring configuration of dynamic routing or router discovery protocols on every end-host.

Since the VRRP protocol is intended for use with IPv4 routers only, this MIB uses the SYNTAX for IP addresses which is specific to IPv4. Thus, changes will be required for this MIB to interoperate in an IPv6 environment.

2.1. VRRP MIB Structure

The VRRP MIB contains three conformance groups:

- vrrpOperations Group: Objects related to VRRP router's configuration and control.
- vrrpStatistics Group: Objects containing information useful in monitoring the operation of VRRP routers.

- vrrpNotifications Group: Consists of objects and definitions for use in SNMP notifications sent by VRRP routers.

Tables in the MIB include the following:

- (1) The vrrpOperTable, which contains objects that define the operational characteristics of a VRRP router. Rows in this table correspond to instances of virtual routers.
- (2) The vrrpAssoIpAddrTable, which contains the addresses of the virtual router(s) that a given VRRP router is backing up.
- (3) The vrrpRouterStatsTable which contains the operating statistics for a VRRP router.

2.2. Virtual Router Redundancy Protocol

This MIB is based on the following characteristics of VRRP as defined in the VRRP specification [17].

- A "VRRP router" is one that is configured to run the VRRP protocol in conjunction with one or more other VRRP routers attached to a LAN.
- A VRRP router can be running one or more instances of a virtual router.
- A "virtual router" is an abstraction which consists of two or more physical routers associated by a Virtual Router Identifier (VRID).
- An instance of a virtual router (on a physical VRRP router), can be uniquely identified by a combination of the 'ifIndex' [18] and "Virtual Router Identifier" (VRID).
- For each VRID there is a set of one or more "associated IP addresses" that are backed-up by the virtual router.

2.3. VRRP MIB Table Design

The tables in the VRRP MIB are structured with the assumption that a VRRP network management application would likely be designed to display information or provide configuration about a VRRP router on a "per-virtual-router basis". Thus, the tables defined in the MIB consist of conceptual rows which are grouped in a manner to present a view of individual virtual routers with a minimal number of SNMP operations.

2.3.1. Relation to Interface Group (RFC 2233) [18].

Since a router can be participating in VRRP on one or more physical interfaces, "ifIndex" is used as an index into the tables defined in the VRRP MIB.

2.4. VRRP Scenarios

The following section provides examples of how some of the objects in this MIB are instantiated for two different VRRP scenarios.

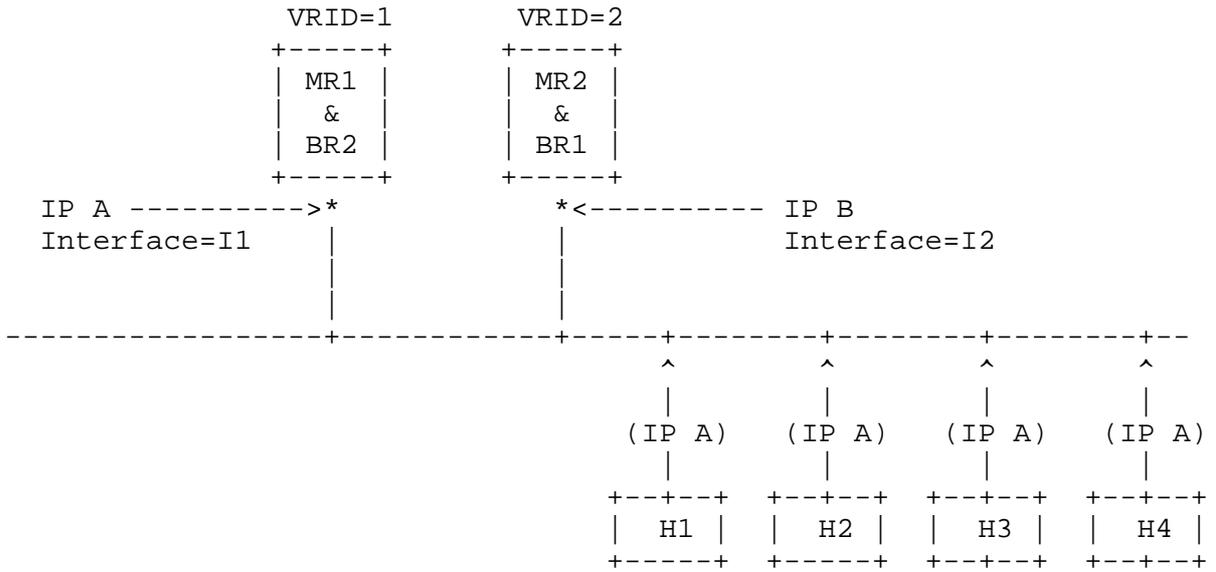
KEY:

The labels in the following tables and diagrams correspond to the actual MIB objects as follows:

```
if      = vrrpOperIfIndex
VrId    = vrrpOperVrId
State   = vrrpOperState
Prior   = vrrpOperPriority
AddrCnt = vrrpOperIpAddrCount
IpAddr  = vrrpOperMasterIpAddr
RowStat = vrrpOperRowStatus
```

2.4.1. VRRP Scenario #1

The following figure shows a simple network with two VRRP routers configured with two virtual routers. This sample topology is taken from the VRRP specification [17]. Addresses in '()' indicate the IP address of the default gateway for a given host, H1 - H4. In the diagram, "Interface" is used in the context defined in IF-MIB [18].



----- MIB Tables For VRRP Router "IP A": -----

vrrpOperTable

if	VrId	State	Prior	AddrCnt	IpAddr	...	RowStat
I1	01	M	255	1	A	()	active
I1	02	B	1-254	1	B	()	active

vrrpAssoIpAddrTable

```

-----
| if | VrId | IP | RowStat |
+---+---+---+---+
| I1 | 01 | A | active |
+---+---+---+---+
| I1 | 02 | B | active |
+---+---+---+---+

```

----- MIB Tables For VRRP Router "IP B": -----

vrrpOperTable

```

-----
| if | VrId | State | Prior | AddrCnt | IpAddr | ... | RowStat |
+---+---+---+---+---+---+---+---+
| I2 | 01 | B | 1-254 | 1 | A | ( ) | active |
+---+---+---+---+---+---+---+---+
| I2 | 02 | M | 255 | 1 | B | ( ) | active |
+---+---+---+---+---+---+---+---+

```

vrrpAssoIpAddrTable

```

-----
| if | VrId | IP | RowStat |
+---+---+---+---+
| I2 | 01 | A | active |
+---+---+---+---+
| I2 | 02 | B | active |
+---+---+---+---+

```


----- MIB Tables For VRRP Router "IP A": -----

vrrpOperTable

if	VrId	State	Prior	AddrCnt	IpAddr	...	RowStat
I1	01	M	255	2	A	()	active
I1	02	B	1-254	1	B	()	active

vrrpAssoIpAddrTable

if	VrId	IP	RowStat
I1	01	A	active
I1	01	C	active
I1	02	B	active

----- MIB Tables For VRRP Router "IP B": -----

vrrpOperTable

if	VrId	State	Prior	AddrCnt	IpAddr	...	RowStat
I2	01	B	1-254	2	A	()	active
I2	02	M	255	1	B	()	active

vrrpAssoIpAddrTable

if	VrId	IP	RowStat
I2	01	A	active
I2	01	C	active
I2	02	B	active

3. Definitions

VRRP-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY, OBJECT-TYPE,
NOTIFICATION-TYPE, Counter32,
Integer32, IpAddress, mib-2 FROM SNMPv2-SMI

TEXTUAL-CONVENTION, RowStatus,
MacAddress, TruthValue, TimeStamp FROM SNMPv2-TC

MODULE-COMPLIANCE, OBJECT-GROUP,
NOTIFICATION-GROUP FROM SNMPv2-CONF
ifIndex FROM IF-MIB;

vrrpMIB MODULE-IDENTITY

LAST-UPDATED "200003030000Z"
ORGANIZATION "IETF VRRP Working Group"
CONTACT-INFO
"Brian R. Jewell
Postal: Copper Mountain Networks, Inc.
2470 Embarcadero Way
Palo Alto, California 94303
Tel: +1 650 687 3367
E-Mail: bjewell@coppermountain.com"

DESCRIPTION

"This MIB describes objects used for managing Virtual Router
Redundancy Protocol (VRRP) routers."
REVISION "200003030000Z" -- 03 Mar 2000
DESCRIPTION "Initial version as published in RFC 2787."
::= { mib-2 68 }

-- *****
-- Textual Conventions
-- *****

VrId ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A number which, along with an interface index (ifIndex),
serves to uniquely identify a virtual router on a given VRRP
router. A set of one or more associated addresses is assigned
to a VRID."

SYNTAX Integer32 (1..255)

```
-- *****
-- VRRP MIB Groups
-- *****
```

```
vrrpOperations      OBJECT IDENTIFIER ::= { vrrpMIB 1 }
vrrpStatistics      OBJECT IDENTIFIER ::= { vrrpMIB 2 }
vrrpConformance     OBJECT IDENTIFIER ::= { vrrpMIB 3 }
```

```
-- *****
-- Start of MIB objects
-- *****
```

```
vrrpNodeVersion    OBJECT-TYPE
SYNTAX              Integer32
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION
    "This value identifies the particular version of the VRRP
    supported by this node."
 ::= { vrrpOperations 1 }
```

```
vrrpNotificationCntl OBJECT-TYPE
SYNTAX              INTEGER {
    enabled          (1),
    disabled         (2)
}
MAX-ACCESS          read-write
STATUS              current
DESCRIPTION
    "Indicates whether the VRRP-enabled router will generate
    SNMP traps for events defined in this MIB. 'Enabled'
    results in SNMP traps; 'disabled', no traps are sent."
DEFVAL { enabled }
 ::= { vrrpOperations 2 }
```

```
-- *****
-- VRRP Operations Table
-- *****
```

```
vrrpOperTable      OBJECT-TYPE
SYNTAX              SEQUENCE OF VrrpOperEntry
MAX-ACCESS          not-accessible
STATUS              current
DESCRIPTION
    "Operations table for a VRRP router which consists of a
    sequence (i.e., one or more conceptual rows) of
    'vrrpOperEntry' items."
```

```
::= { vrrpOperations 3 }
```

```
vrrpOperEntry OBJECT-TYPE
```

```
SYNTAX          VrrpOperEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
```

"An entry in the vrrpOperTable containing the operational characteristics of a virtual router. On a VRRP router, a given virtual router is identified by a combination of the IF index and VRID.

Rows in the table cannot be modified unless the value of 'vrrpOperAdminState' is 'disabled' and the 'vrrpOperState' has transitioned to 'initialize'."

```
INDEX          { ifIndex, vrrpOperVrId }
::= { vrrpOperTable 1 }
```

```
VrrpOperEntry ::=
```

```
SEQUENCE {
    vrrpOperVrId
        VrId,
    vrrpOperVirtualMacAddr
        MacAddress,
    vrrpOperState
        INTEGER,
    vrrpOperAdminState
        INTEGER,
    vrrpOperPriority
        Integer32,
    vrrpOperIpAddrCount
        Integer32,
    vrrpOperMasterIpAddr
        IpAddress,
    vrrpOperPrimaryIpAddr
        IpAddress,
    vrrpOperAuthType
        INTEGER,
    vrrpOperAuthKey
        OCTET STRING,
    vrrpOperAdvertisementInterval
        Integer32,
    vrrpOperPreemptMode
        TruthValue,
    vrrpOperVirtualRouterUpTime
        TimeStamp,
    vrrpOperProtocol
```

```

        INTEGER,
        vrrpOperRowStatus
        RowStatus
    }

vrrpOperVrId OBJECT-TYPE
    SYNTAX      VrId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This object contains the Virtual Router Identifier (VRID)."
```

::= { vrrpOperEntry 1 }

```

vrrpOperVirtualMacAddr OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The virtual MAC address of the virtual router. Although this
        object can be derived from the 'vrrpOperVrId' object, it is
        defined so that it is easily obtainable by a management
        application and can be included in VRRP-related SNMP traps."
```

::= { vrrpOperEntry 2 }

```

vrrpOperState OBJECT-TYPE
    SYNTAX      INTEGER {
        initialize(1),
        backup(2),
        master(3)
    }
    MAX-ACCESS  read-only
    STATUS      current

    DESCRIPTION
        "The current state of the virtual router. This object has
        three defined values:

        - 'initialize', which indicates that all the
          virtual router is waiting for a startup event.

        - 'backup', which indicates the virtual router is
          monitoring the availability of the master router.

        - 'master', which indicates that the virtual router
          is forwarding packets for IP addresses that are
          associated with this router.

        Setting the 'vrrpOperAdminState' object (below) initiates
```

transitions in the value of this object."
 ::= { vrrpOperEntry 3 }

vrrpOperAdminState OBJECT-TYPE

SYNTAX INTEGER {

up(1),
 down(2)

}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object will enable/disable the virtual router function. Setting the value to 'up', will transition the state of the virtual router from 'initialize' to 'backup' or 'master', depending on the value of 'vrrpOperPriority'. Setting the value to 'down', will transition the router from 'master' or 'backup' to 'initialize'. State transitions may not be immediate; they sometimes depend on other factors, such as the interface (IF) state.

The 'vrrpOperAdminState' object must be set to 'down' prior to modifying the other read-create objects in the conceptual row. The value of the 'vrrpOperRowStatus' object (below) must be 'active', signifying that the conceptual row is valid (i.e., the objects are correctly set), in order for this object to be set to 'up'."

DEFVAL { down }

::= { vrrpOperEntry 4 }

vrrpOperPriority OBJECT-TYPE

SYNTAX Integer32 (0..255)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object specifies the priority to be used for the virtual router master election process. Higher values imply higher priority.

A priority of '0', although not settable, is sent by the master router to indicate that this router has ceased to participate in VRRP and a backup virtual router should transition to become a new master.

A priority of 255 is used for the router that owns the associated IP address(es)."

DEFVAL { 100 }

::= { vrrpOperEntry 5 }

vrrpOperIpAddrCount OBJECT-TYPE

SYNTAX Integer32 (0..255)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IP addresses that are associated with this virtual router. This number is equal to the number of rows in the vrrpAssoIpAddrTable that correspond to a given IF index/VRID pair."

::= { vrrpOperEntry 6 }

vrrpOperMasterIpAddr OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The master router's real (primary) IP address. This is the IP address listed as the source in VRRP advertisement last received by this virtual router."

::= { vrrpOperEntry 7 }

vrrpOperPrimaryIpAddr OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"In the case where there is more than one IP address for a given 'ifIndex', this object is used to specify the IP address that will become the 'vrrpOperMasterIpAddr', should the virtual router transition from backup to master. If this object is set to 0.0.0.0, the IP address which is numerically lowest will be selected."

DEFVAL { '00000000'H } -- 0.0.0.0

::= { vrrpOperEntry 8 }

vrrpOperAuthType OBJECT-TYPE

SYNTAX INTEGER {

noAuthentication(1),

-- VRRP protocol exchanges are not
-- authenticated.

simpleTextPassword(2),

-- Exchanges are authenticated by a
-- clear text password.

ipAuthenticationHeader(3)

-- Exchanges are authenticated using
-- the IP authentication header.

}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Authentication type used for VRRP protocol exchanges between virtual routers. This value of this object is the same for a given ifIndex.

New enumerations to this list can only be added via a new RFC on the standards track."

```
DEFVAL      { noAuthentication }
 ::= { vrrpOperEntry 9 }
```

vrrpOperAuthKey OBJECT-TYPE

```
SYNTAX      OCTET STRING (SIZE (0..16))
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"The Authentication Key. This object is set according to the value of the 'vrrpOperAuthType' object ('simpleTextPassword' or 'ipAuthenticationHeader'). If the length of the value is less than 16 octets, the agent will left adjust and zero fill to 16 octets. The value of this object is the same for a given ifIndex.

When read, vrrpOperAuthKey always returns an Octet String of length zero."

```
::= { vrrpOperEntry 10 }
```

vrrpOperAdvertisementInterval OBJECT-TYPE

```
SYNTAX      Integer32 (1..255)
UNITS       "seconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"The time interval, in seconds, between sending advertisement messages. Only the master router sends VRRP advertisements."

```
DEFVAL      { 1 }
 ::= { vrrpOperEntry 11 }
```

vrrpOperPreemptMode OBJECT-TYPE

```
SYNTAX      TruthValue
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"Controls whether a higher priority virtual router will preempt a lower priority master."

```
DEFVAL      { true }
 ::= { vrrpOperEntry 12 }
```

vrrpOperVirtualRouterUpTime OBJECT-TYPE

```

SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This is the value of the 'sysUpTime' object when this
    virtual router (i.e., the 'vrrpOperState') transitioned
    out of 'initialized'."
 ::= { vrrpOperEntry 13 }

```

vrrpOperProtocol OBJECT-TYPE

```

SYNTAX      INTEGER {
    ip (1),
    bridge (2),
    decnet (3),
    other (4)
}
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The particular protocol being controlled by this Virtual
    Router.

    New enumerations to this list can only be added via a new
    RFC on the standards track."
DEFVAL { ip }
 ::= { vrrpOperEntry 14 }

```

vrrpOperRowStatus OBJECT-TYPE

```

SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The row status variable, used in accordance to installation
    and removal conventions for conceptual rows. The rowstatus of
    a currently active row in the vrrpOperTable is constrained
    by the operational state of the corresponding virtual router.
    When 'vrrpOperRowStatus' is set to active(1), no other
    objects in the conceptual row, with the exception of
    'vrrpOperAdminState', can be modified. Prior to setting the
    'vrrpOperRowStatus' object from 'active' to a different value,
    the 'vrrpOperAdminState' object must be set to 'down' and the
    'vrrpOperState' object be transitioned to 'initialize'.

    To create a row in this table, a manager sets this object
    to either createAndGo(4) or createAndWait(5). Until instances
    of all corresponding columns are appropriately configured,
    the value of the corresponding instance of the 'vrrpOperRowStatus'
    column will be read as notReady(3).

```

In particular, a newly created row cannot be made active(1) until (minimally) the corresponding instance of 'vrrpOperVrId' has been set and there is at least one active row in the 'vrrpAssoIpAddrTable' defining an associated IP address for the virtual router."

```
::= { vrrpOperEntry 15 }
```

```
-- *****
-- VRRP Associated IP Address Table
-- *****
```

vrrpAssoIpAddrTable OBJECT-TYPE

SYNTAX SEQUENCE OF VrrpAssoIpAddrEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The table of addresses associated with this virtual router."

```
::= { vrrpOperations 4 }
```

vrrpAssoIpAddrEntry OBJECT-TYPE

SYNTAX VrrpAssoIpAddrEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table contains an IP address that is associated with a virtual router. The number of rows for a given ifIndex and VrId will equal the number of IP addresses associated (e.g., backed up) by the virtual router (equivalent to 'vrrpOperIpAddrCount')."

Rows in the table cannot be modified unless the value of 'vrrpOperAdminState' is 'disabled' and the 'vrrpOperState' has transitioned to 'initialize'."

INDEX { ifIndex, vrrpOperVrId, vrrpAssoIpAddr }

```
::= { vrrpAssoIpAddrTable 1 }
```

VrrpAssoIpAddrEntry ::=

SEQUENCE {

vrrpAssoIpAddr

IpAddress,

vrrpAssoIpAddrRowStatus

RowStatus

}

vrrpAssoIpAddr OBJECT-TYPE

SYNTAX IpAddress

```

MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "The assigned IP addresses that a virtual router is
    responsible for backing up."
 ::= { vrrpAssoIpAddrEntry 1 }

```

```

vrrpAssoIpAddrRowStatus OBJECT-TYPE
SYNTAX        RowStatus
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION
    "The row status variable, used according to installation
    and removal conventions for conceptual rows. Setting this
    object to active(1) or createAndGo(4) results in the
    addition of an associated address for a virtual router.
    Destroying the entry or setting it to notInService(2)
    removes the associated address from the virtual router.
    The use of other values is implementation-dependent."
 ::= { vrrpAssoIpAddrEntry 2 }

```

```

-- *****
-- VRRP Router Statistics
-- *****

```

```

vrrpRouterChecksumErrors OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The total number of VRRP packets received with an invalid
    VRRP checksum value."
 ::= { vrrpStatistics 1 }

```

```

vrrpRouterVersionErrors OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The total number of VRRP packets received with an unknown
    or unsupported version number."
 ::= { vrrpStatistics 2 }

```

```

vrrpRouterVrIdErrors OBJECT-TYPE
SYNTAX        Counter32
MAX-ACCESS    read-only
STATUS        current

```

DESCRIPTION

"The total number of VRRP packets received with an invalid VRID for this virtual router."

::= { vrrpStatistics 3 }

```
-- *****
-- VRRP Router Statistics Table
-- *****
```

vrrpRouterStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF VrrpRouterStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of virtual router statistics."

::= { vrrpStatistics 4 }

vrrpRouterStatsEntry OBJECT-TYPE

SYNTAX VrrpRouterStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing statistics information about a given virtual router."

AUGMENTS { vrrpOperEntry }

::= { vrrpRouterStatsTable 1 }

VrrpRouterStatsEntry ::=

```
SEQUENCE {
    vrrpStatsBecomeMaster
        Counter32,
    vrrpStatsAdvertiseRcvd
        Counter32,
    vrrpStatsAdvertiseIntervalErrors
        Counter32,
    vrrpStatsAuthFailures
        Counter32,
    vrrpStatsIpTtlErrors
        Counter32,
    vrrpStatsPriorityZeroPktsRcvd
        Counter32,
    vrrpStatsPriorityZeroPktsSent
        Counter32,
    vrrpStatsInvalidTypePktsRcvd
        Counter32,
    vrrpStatsAddressListErrors
        Counter32,
    vrrpStatsInvalidAuthType
```

```
        Counter32,
    vrrpStatsAuthTypeMismatch
        Counter32,
    vrrpStatsPacketLengthErrors
        Counter32
    }

vrrpStatsBecomeMaster OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of times that this virtual router's state
         has transitioned to MASTER."
    ::= { vrrpRouterStatsEntry 1 }

vrrpStatsAdvertiserCvcd OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of VRRP advertisements received by this
         virtual router."
    ::= { vrrpRouterStatsEntry 2 }

vrrpStatsAdvertiseIntervalErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    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."
    ::= { vrrpRouterStatsEntry 3 }

vrrpStatsAuthFailures OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of VRRP packets received that do not pass
         the authentication check."
    ::= { vrrpRouterStatsEntry 4 }

vrrpStatsIpTtlErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
```

DESCRIPTION

"The total number of VRRP packets received by the virtual router with IP TTL (Time-To-Live) not equal to 255."

::= { vrrpRouterStatsEntry 5 }

vrrpStatsPriorityZeroPktsRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of VRRP packets received by the virtual router with a priority of '0'."

::= { vrrpRouterStatsEntry 6 }

vrrpStatsPriorityZeroPktsSent OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of VRRP packets sent by the virtual router with a priority of '0'."

::= { vrrpRouterStatsEntry 7 }

vrrpStatsInvalidTypePktsRcvd OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of VRRP packets received by the virtual router with an invalid value in the 'type' field."

::= { vrrpRouterStatsEntry 8 }

vrrpStatsAddressListErrors OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of packets received for which the address list does not match the locally configured list for the virtual router."

::= { vrrpRouterStatsEntry 9 }

vrrpStatsInvalidAuthType OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of packets received with an unknown

```

        authentication type."
 ::= { vrrpRouterStatsEntry 10 }

vrrpStatsAuthTypeMismatch OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The total number of packets received with 'Auth Type' not
    equal to the locally configured authentication method
    ('vrrpOperAuthType')."
 ::= { vrrpRouterStatsEntry 11 }

vrrpStatsPacketLengthErrors OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The total number of packets received with a packet length
    less than the length of the VRRP header."
 ::= { vrrpRouterStatsEntry 12 }

-- *****
-- Trap Definitions
-- *****

vrrpNotifications OBJECT IDENTIFIER ::= { vrrpMIB 0 }

vrrpTrapPacketSrc OBJECT-TYPE
SYNTAX      IpAddress
MAX-ACCESS  accessible-for-notify
STATUS      current
DESCRIPTION
    "The IP address of an inbound VRRP packet. Used by
    vrrpTrapAuthFailure trap."
 ::= { vrrpOperations 5 }

vrrpTrapAuthErrorType OBJECT-TYPE
SYNTAX      INTEGER {
    invalidAuthType (1),
    authTypeMismatch (2),
    authFailure (3)
}
MAX-ACCESS  accessible-for-notify
STATUS      current
DESCRIPTION
    "Potential types of configuration conflicts.
    Used by vrrpAuthFailure trap."

```

::= { vrrpOperations 6 }

vrrpTrapNewMaster NOTIFICATION-TYPE
OBJECTS { vrrpOperMasterIpAddr }

STATUS current

DESCRIPTION
"The newMaster trap indicates that the sending agent
has transitioned to 'Master' state."

::= { vrrpNotifications 1 }

vrrpTrapAuthFailure NOTIFICATION-TYPE
OBJECTS { vrrpTrapPacketSrc, vrrpTrapAuthErrorType }

STATUS current

DESCRIPTION
"A vrrpAuthFailure trap signifies that a packet has
been received from a router whose authentication key
or authentication type conflicts with this router's
authentication key or authentication type. Implementation
of this trap is optional."

::= { vrrpNotifications 2 }

-- *****
-- Conformance Information
-- *****

vrrpMIBCompliances OBJECT IDENTIFIER ::= { vrrpConformance 1 }
vrrpMIBGroups OBJECT IDENTIFIER ::= { vrrpConformance 2 }

--
-- Compliance Statements
--

vrrpMIBCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
"The core compliance statement for all VRRP implementations."
MODULE -- this module
MANDATORY-GROUPS {
vrrpOperGroup,
vrrpStatsGroup
}
OBJECT vrrpOperPriority
WRITE-SYNTAX Integer32 (1..255)
DESCRIPTION "SETable values are from 1 to 255."

```

 ::= { vrrpMIBCompliances 1 }

-- .....
-- Conformance Groups
-- .....

vrrpOperGroup OBJECT-GROUP
  OBJECTS {
    vrrpNodeVersion,
    vrrpNotificationCntl,
    vrrpOperVirtualMacAddr,
    vrrpOperState,
    vrrpOperAdminState,
    vrrpOperPriority,
    vrrpOperIpAddrCount,
    vrrpOperMasterIpAddr,
    vrrpOperPrimaryIpAddr,
    vrrpOperAuthType,
    vrrpOperAuthKey,
    vrrpOperAdvertisementInterval,
    vrrpOperPreemptMode,
    vrrpOperVirtualRouterUpTime,
    vrrpOperProtocol,
    vrrpOperRowStatus,
    vrrpAssoIpAddrRowStatus
  }
  STATUS current
  DESCRIPTION
    "Conformance group for VRRP operations."
  ::= { vrrpMIBGroups 1 }

vrrpStatsGroup OBJECT-GROUP
  OBJECTS {
    vrrpRouterChecksumErrors,
    vrrpRouterVersionErrors,
    vrrpRouterVrIdErrors,
    vrrpStatsBecomeMaster,
    vrrpStatsAdvertiseRcvd,
    vrrpStatsAdvertiseIntervalErrors,
    vrrpStatsAuthFailures,
    vrrpStatsIpTtlErrors,
    vrrpStatsPriorityZeroPktsRcvd,
    vrrpStatsPriorityZeroPktsSent,
    vrrpStatsInvalidTypePktsRcvd,
    vrrpStatsAddressListErrors,
    vrrpStatsInvalidAuthType,
    vrrpStatsAuthTypeMismatch,
    vrrpStatsPacketLengthErrors
  }

```

```
    }
    STATUS current
    DESCRIPTION
        "Conformance group for VRRP statistics."
    ::= { vrrpMIBGroups 2 }

vrrpTrapGroup OBJECT-GROUP
    OBJECTS {
        vrrpTrapPacketSrc,
        vrrpTrapAuthErrorType
    }
    STATUS current
    DESCRIPTION
        "Conformance group for objects contained in VRRP notifications."
    ::= { vrrpMIBGroups 3 }

vrrpNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        vrrpTrapNewMaster,
        vrrpTrapAuthFailure
    }
    STATUS current
    DESCRIPTION
        "The VRRP MIB Notification Group."
    ::= { vrrpMIBGroups 4 }

END
```

4. Security Considerations

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write or read-create. Such objects may be considered sensitive or vulnerable to security attacks in some networking environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on VRRP router operations.

A number of objects in the vrrpOperTable possess the read-create attribute. Manipulation of these objects is capable of affecting the operation of a virtual router.

Specific examples of this include, but are not limited to:

- o The vrrpOperAdminState object which could be used to disable a virtual router.
- o The vrrpOperPrimaryIpAddr object which, if compromised, could allow assignment of an invalid IP address to a master router.

- o The authentication type/key related objects which could potentially render the VRRP security mechanisms ineffective.

Of additional concern is the ability to disable the transmission of traps. This would nullify the capability of a virtual router to provide notification in the event of an authentication failure.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [RFC2574] and the View-based Access Control Model RFC 2575 [RFC2575] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

5. Acknowledgements

The authors would like to thank Danny Mitzel, Venkat Prasad, Al Pham, Robert Hinden, Venkat Prasad, Barbera Denny, Fred Baker, Jeff Case, Flavio Fernandes, Acee Lindem, Scott Barvick, and Bert Wijnen for their comments and suggestions.

6. References

- [1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.
- [5] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

- [6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIV2", STD 58, RFC 2579, April 1999.
- [7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIV2", STD 58, RFC 2580, April 1999.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.
- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.
- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.
- [13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMPv3 Applications", RFC 2573, April 1999.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999
- [16] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999
- [17] Knight, S., Weaver, D., Whipple, D., Hinden, R., Mitzel, D., Hunt, P., Higginson, P., Shand, M. and Lindem, A., "Virtual Router Redundancy Protocol", RFC 2338, November 1997.
- [18] McCloghrie, K. and F. Kastholz, "The Interfaces Group MIB using SMIV2", RFC 2233, November 1997.

7. Authors' Addresses

Brian R. Jewell
Copper Mountain Networks, Inc.
2470 Embarcadero Way
Palo Alto, California 94303
US

Phone: +1 650 687 3367
EMail: bjewell@coppermountain.com

David Chuang
CoSine Communications
1200 Bridge Parkway
Redwood City, CA 94065
US

Phone: +1 650 628 4850
EMail: david_chuang@cosinecom.com

8. Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

9. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

