Virtual Router Redundancy Service

Last Updated: November 16, 2011

Virtual Router Redundancy Service (VRRS) provides a multiclient information abstraction and management service between a First Hop Redundancy Protocol (FHRP) and a registered client. The VRRS multiclient service provides a consistent interface with FHRP protocols by abstracting over several FHRPs and providing an idealized view of their state. VRRS manages data updates, allowing interested clients to register in one place and receive updates for named FHRP groups or all registered FHRP groups.

Virtual Router Redundancy Protocol (VRRP) is an FHRP that acts as a server that pushes FHRP status information out to all registered VRRS clients. Clients obtain status on essential information provided by the FHRP, including current and previous redundancy states, active and inactive L3 and L2 addresses, and, in some cases, information about other redundant gateways in the network. Clients can use this information to provide stateless and stateful redundancy information to clients and protocols.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for VRRS

- VRRS plug-ins must be configured on subinterfaces that are not configured with an FHRP, but which share a physical interface with an FHRP it is following.
- VRRPv2 is configurable only on Gigabit Ethernet interfaces.

Information About VRRS

VRRS Overview

VRRS improves the scalability of FHRP. VRRS provides a stateless redundancy service to applications (VRRS clients) by monitoring VRRP. VRRS provides a database of the current VRRP state and operates without maintaining sessions or keeping track of previous states of the clients and servers with which it communicates. VRRP acts as a VRRS server. VRRS clients are other Cisco IOS processes or applications that use VRRP to provide or withhold a service or resource dependent upon the state of the group.

VRRS by itself is limited to maintaining its own state. Linking a VRRS client to a VRRP group provides a mechanism that allows VRRS to provide a service to client applications so they can implement stateless or stateful failover. Stateless failover is failover without syncing of state. Stateful failover requires communication with a nominated backup before failure so that operational data is not lost when failover occurs.

Using VRRS with VRRP

VRRP provides server support for VRRS. The VRRP server pushes state and status information to VRRS when an internal update occurs. VRRS updates its internal database upon receiving a server update, and then sends push notifications to each of the VRRS clients associated with the shared name. Clients are interested in the protocol state, virtual MAC address, and virtual IP address information associated with a group. The association name between a client and a VRRP group is a character name string. The information provided by VRRS allows clients to perform various activities that are dependent on the state of the associated VRRP group.

VRRP notifies VRRS of its current state (master, backup, or nonoperational INIT). The VRRP state is then passed on to clients or acted on by a plug-in. A VRRP group should be configured with a name to activate VRRS. Clients should be configured with the same name to bind them with VRRS.

The VRRP group name associates the VRRP group with any clients that are configured as part of VRRS with the same name.

VRRS Servers and Clients

VRRP acts as the VRRS server. Clients act on the VRRP server state. When a VRRP group changes state, VRRS clients act by altering their behavior (performing tasks such as shutting down interfaces or appending accounting logs) depending on the state received from VRRS.

The following can be VRRS clients:

- PPP over Ethernet (PPPoE) subinterfaces
- Access Node Control Protocol (ANCP) subinterfaces
- VRRS Interface-state plug-in
- VRRS MAC-Address plug-in
- VRRS Accounting plug-in

VRRS plug-ins extend the failover of VRRP without the need for configuring VRRP groups on all subinterfaces. Configuring a VRRS plug-in on subinterfaces is a substitute for having to configure multiple VRRP groups on many subinterfaces. Plug-ins provide a light-weight version of VRRP and scale better than a fully configured VRRP group. The state of the plug-ins follows the VRRP server state. Client plug-ins are configured on other subinterfaces that share the same physical interface as VRRP.

VRRS MAC-Address Plug-in

The VRRS MAC-Address plug-in provides a mechanism for controlling a virtual MAC address associated with the primary interface IP address. If the VRRS MAC-Address plug-in is configured on an interface, and a VRRP group shares a name association with the plug-in, then a VRRS active state associates a virtual MAC address with the configured primary IP address.

The VRRS MAC-Address plug-in is only interested in the VRRS active state, which is interpreted as up. All other states are interpreted as down. When the state is up and the additional interface criteria listed below have been met, then the VRRS MAC-Address plug-in provides the following services:

- Overwrites the interface IP address ARP table with a virtual MAC address provided by VRRS
- Inserts the virtual MAC address provided by VRRS into the MAC address filter of the interface
- Controls the ARP reply mechanism by substituting a VRRS-provided virtual MAC address
- Broadcasts unsolicited ARP messages that include the VRRS virtual MAC address

When VRRS is in a nonactive state, the virtual MAC address is unassociated from the primary IP address.

When you use the VRRS MAC-Address plug-in, the VRRS Interface-State plug-in must also be used in order to prevent address conflicts with other redundant members.

Additional interface criteria:

- Interfaces must be configured with an interface IP address.
- Interfaces must be in the line-protocol up state.
- Other FHRP protocols cannot be configured on the interface; these include HSRP, VRRP, and GLBP.

The VRRS MAC-Address plug-in is associated with a VRRS group name by configuring the **vrrs follow** *name* command.

VRRS Interface-State Plug-in

The VRRS Interface-State plug-in provides a mechanism for controlling the line-protocol state of a subinterface based on the state of VRRP. The VRRS Interface-State plug-in is an extension of the VRRS, and is directly controlled by the push events associated with the VRRS. If the plug-in is configured on an

interface, and a VRRP group shares a name association with the VRRS plug-in, then a VRRS active state allows the line- protocol state of the interface to be up. A VRRS nonactive state will cause the line protocol of the interface to be down.

Note　When first configured, the interface line protocol may immediately change to the down state until the VRRS state is confirmed as up.

The VRRS Interface-State plug-in is associated with a VRRS group name by configuring the **vrrs follow** *name* command.

The Interface-State plug-in restricts the operation of the **no shutdown** command. When an interface is line-protocol down, the interface state will not go up.

VRRS Accounting Plug-in

The VRRS Accounting plug-in provides a configurable AAA method list mechanism that provides updates to a RADIUS server when a VRRS group transitions its state.

The VRRS accounting plug-in is an extension of existing AAA system accounting messages. The VRRS Accounting plug-in provides accounting-on and accounting-off messages and an additional Vendor-Specific Attribute (VSA) that sends the configured VRRS name in RADIUS accounting messages. The VRRS name is configured using the **vrrp name** command in interface configuration mode.

The VRRS Accounting plug-in sends an accounting-on message to RADIUS when a VRRS group transitions to the master state, and it sends an accounting-off message when a VRRS group transitions from the master state.

The following RADIUS attributes are included in VRRS accounting messages by default:

- Attribute 4, NAS-IP-Address
- Attribute 26, Cisco VSA Type 1, VRRS
- Attribute 40, Acct-Status-Type
- Attribute 41, Acct-Delay-Time
- Attribute 44, Acct-Session-Id

Accounting messages for a VRRS transitioning out of master state are sent after all PPPoE accounting stop messages for sessions that are part of that VRRS.

The VRRS accounting type is implemented by AAA to support VRRS accounting.

How to Configure VRRS

Configuring a VRRS Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
5. **vrrp** *group-number* **name** [*vrrp-group-name*]
6. **vrrp** *group* **ip** *ip-address* [**secondary**]
7. **vrrp delay** {**minimum** *seconds* [**reload** *seconds*] | **reload** *seconds*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** Router(config)# **interface gigabitethernet 0/0/0**	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** Router(config-if)# **ip address 10.0.0.1 255.255.255.0**	Sets a primary or secondary IP address for an interface.
Step 5	**vrrp** *group-number* **name** [*vrrp-group-name*] **Example:** Router(config-if)# **vrrp 1 name name1**	Links a VRRS client to a VRRP group.

	Command or Action	Purpose
Step 6	**vrrp** *group* **ip** *ip-address* [**secondary**] **Example:** Router(config-if)# **vrrp 1 ip 10.0.1.20**	Enables VRRP on an interface and identifies the IP address of the virtual router.
Step 7	**vrrp delay** {**minimum** *seconds* [**reload** *seconds*]	**reload** *seconds*} **Example:** Router(config-if)# **vrrp delay minimum 30 reload 60**	Configures the delay period before the initialization of all VRRP groups on an interface. The recommended **minimum** *seconds* value is 30 seconds, and the recommended **reload** *seconds* value is 60 seconds.

Configuring the Clients That Use VRRS

Perform this task to configure the clients, including VRRS plug-ins, that use VRRS. This task is configured on multiple subinterfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number***.***subinterface*
4. **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]]
5. **vrrs follow** *name*
6. **vrrs interface-state**
7. **vrrs mac-address** [**arp** [**interval** *seconds*] [**duration** *seconds*]]
8. Repeat Steps 3 through 7 to configure additional subinterfaces.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.

Command or Action	Purpose
Step 3 **interface** *type number.subinterface* **Example:** Router(config)# **interface gigabitethernet0/0/0.1**	Configures a subinterface type and enters subinterface configuration mode.
Step 4 **ip address** *ip-address mask* [**secondary** [**vrf** *vrf-name*]] **Example:** Router(config-subif)# **ip address 10.0.0.1 255.255.255.0**	Sets a primary or secondary IP address for an interface. This interface should be a subinterface.
Step 5 **vrrs follow** *name* **Example:** Router(config-subif)# **vrrs follow name1**	Configures a name association between VRRS plug-ins and the VRRS server.
Step 6 **vrrs interface-state** **Example:** Router(config-subif)# **vrrs interface-state**	(Optional) Configures the VRRP shutdown plug-in on an interface.
Step 7 **vrrs mac-address** [**arp** [**interval** *seconds*] [**duration** *seconds*]] **Example:** Router(config-subif)# **vrrs mac-address**	(Optional) Configures the VRRS MAC-Address plug-in on an interface.
Step 8 Repeat Steps 3 through 7 to configure additional subinterfaces.	

Configuring VRRS Accounting

Perform this task to configure VRRS to send AAA accounting messages to the AAA server when there is a state-change in VRRS from active to standby or from standby to active.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa accounting vrrs** { **default** | *list-name* **start-stop** [*method1* [*method2...*]]
4. **aaa attribute list** *list-name*
5. **attribute type** *name value* [**service** *service*] [**protocol** *protocol*] [**mandatory**] [**tag** *tag-value*]
6. **exit**
7. **vrrs** *vrrs-group-name*
8. **accounting delay** *delay*
9. **accounting method** {**default** | *accounting-method-list*}
10. **attribute list** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.
Step 3	**aaa accounting vrrs** { **default**	*list-name* **start-stop** [*method1* [*method2...*]] **Example:** Router(config)# **aaa accounting vrrs vrrp-mlist-1 start-stop group radius**	Enables AAA accounting of requested services for billing or security purposes when you use VRRS.
Step 4	**aaa attribute list** *list-name* **Example:** Router(config)# **aaa attribute list vrrp-1-attr**	Defines a AAA attribute list locally on a router.

	Command or Action	Purpose
Step 5	**attribute type** *name value* [**service** *service*] [**protocol** *protocol*] [**mandatory**] [**tag** *tag-value*]	Defines an attribute type that is to be added to an attribute list locally on a router.
	Example:	
	Router(config-attr-list)# attribute type account-delay "10"	
Step 6	**exit**	Exits attribute list configuration mode and returns to global configuration mode.
	Example:	
	Router(config-attr-list)# **exit**	
Step 7	**vrrs** *vrrs-group-name*	(Optional) Specifies a distinct AAA accounting method list to use, a nonzero delay time for accounting-off messages, and additional attributes other than the default for a VRRP group.
	Example:	
	Router(config)# **vrrs vrrp-name-1**	
Step 8	**accounting delay** *delay*	(Optional) Specifies a delay time for sending accounting-off messages for VRRS.
	Example:	
	Router(config-vrrs)# **accounting delay 10**	
Step 9	**accounting method** {**default**	*accounting-method-list*}	(Optional) Enables VRRS accounting for a VRRP group.
	Example:	
	Router(config-vrrs)# **accounting method METHOD1**	
Step 10	**attribute list** *list-name*	(Optional) Specifies additional attributes to include in VRRS accounting-on and accounting-off messages.
	Example:	
	Router(config-vrrs)# **attribute list vrrp-1-attr**	

Monitoring and Maintaining VRRS

SUMMARY STEPS

1. **debug vrrp vrrs**
2. **debug vrrs accounting** {**all** | **errors** | **events**}
3. **debug vrrs infra** {**all** | **client** | **events** | **server**}
4. **debug vrrs plugin** {**all** | **arp-packet** | **client** | **database** | **if-state** | **mac** | **process** | **sublock** | **test**}
5. **show vrrs clients**
6. **show vrrs group** [*group-name*]
7. **show vrrs plugin database**
8. **show vrrs summary**

DETAILED STEPS

Step 1   **debug vrrp vrrs**
This command enables VRRP debugging statements for VRRS interactions.

Example:

```
Router# debug vrrp vrrs

VRRP VRRS debugging is on
*Feb  5 09:29:47.005: VRRP: Registered VRRS group "name1"
*Feb  5 09:29:53.237: VRRP: Updated info for VRRS group name1
*Feb  5 09:30:14.153: VRRP: Unregistered VRRS group "name1"
*Feb  5 09:30:14.153: VRRP: Registered VRRS group "name2"
*Feb  5 09:30:22.689: VRRP: Unregistered VRRS group "name2"
```

Step 2   **debug vrrs accounting** {**all** | **errors** | **events**}
This command enables debug messages for VRRS accounting.

Example:

```
Router# debug vrrs accounting

00:16:13: VRRS/ACCT/EV: entry create for abc(0x4E8C1F0)
00:16:13: VRRS/ACCT/EV: abc(0x4E8C1F0 12000006) client add ok2(No group)
```

Step 3   **debug vrrs infra** {**all** | **client** | **events** | **server**}
This command enables VRRS infrastructure debug messages.

Example:

```
Router# debug vrrs infra

*Sep 9 16:09:53.848: VRRS: Client 21 is not registered
*Sep 9 16:09:53.848: VRRS: Client 21 unregister failed
*Sep 9 16:09:53.848: VRRS: Client VRRS TEST CLIENT registered, id 21
*Sep 9 16:09:53.848: VRRS: Client 21 add, group VRRP-TEST-1 does not exist, allocating...
*Sep 9 16:09:53.848: VRRS: Client 21 add to VRRP-TEST-1. Vrrs handle F7000001, client
handle FE720
```

```
*Sep 9 16:09:53.848: VRRS: Server VRRP add, group VRRP-TEST-1, state INIT, vrrs handle
F7000001
```

Step 4   **debug vrrs plugin** {**all** | **arp-packet** | **client** | **database** | **if-state** | **mac** | **process** | **sublock** | **test**}

This command enables VRRS plug-in debug messages.

Example:

```
Router# debug vrrs plugin

Feb 17 19:15:38.052: VRRS-P(mac): GigEth0/0/0.1 Add 0000.12ad.0001 to MAC filter, using
(afilter_add)
Feb 17 19:15:38.053: VRRS-P(mac): Active count increase to (2) for MAC : 0000.12ad.0001
```

Step 5   **show vrrs clients**

This command displays a list of VRRS clients.

Example:

```
Router# show vrrs clients

ID  Priority  All-groups  Name
-----------------------------
1   High      No          VRRS-Plugins
2   Low       Yes         VRRS-Accounting
3   Normal    No          PPPOE-VRRS-CLIENT
```

Step 6   **show vrrs group** [*group-name*]

This command displays information about VRRS groups.

Example:

```
Router# show vrrs group DT-CLUSTER-3

DT-CLUSTER-3
Server Not configured, state INIT, old state INIT, reason Protocol
  Address family IPv4, Virtual address 0.0.0.0, Virtual mac 0000.0000.0000
  Active interface address 0.0.0.0, standby interface address 0.0.0.0
Client 5 VRRS TEST CLIENT, priority Low
```

Step 7   **show vrrs plugin database**

This command displays details about the internal VRRS plug-in database.

Example:

```
Router# show vrrs plugin database

VRRS Plugin Database
---------------------------------------------
Name = VRRS_NAME_1
Server connection = Live
State = Disabled
MAC addr = 0000.5e00.0101
Test Control = False
Client Handle = 3741319170
Interface list =
                gige0/0/0.2
                gige0/0/0.3
```

Step 8   **show vrrs summary**

This command displays a summary of all VRRS groups.

Example:

```
Router# show vrrs summary

Group                                         Server State Virtual-address
-------------------------------------------------------------------------
DT-CLUSTER-3                                  UNKNOW INIT    0.0.0.0
DT-CLUSTER-2                                  VRRP   BACKUP 11.1.1.1
DT-CLUSTER-1                                  VRRP   ACTIVE  1.1.1.1
```

Configuration Examples for VRRS

Example: Configuring a VRRS Server

```
Router# configure terminal
Router(config)# interface gigabitethernet0/0/0
Router(config-if)# ip address 10.0.0.1 255.255.255.0
Router(config-if)# vrrp 1 name name1
Router(config-if)# vrrp 1 ip 10.0.1.20
Router(config-if)# vrrp delay minimum 30 reload 60
```

Example: Configuring the Clients that use VRRS

The following example shows how to configure the clients, including VRRS plug-ins, that use VRRS.

```
Router# configure terminal
Router(config)# interface gigabitethernet0/0/0.1
Router(config-subif)# ip address 10.0.0.1 255.255.255.0
Router(config-subif)# vrrs follow name1
Router(config-subif)# vrrs interface-state
Router(config-subif)# vrrs mac-address
```

Example: Configuring VRRS Accounting

The following example shows how to configure VRRS to send AAA accounting messages to the AAA server when there is a state change in VRRS from active to standby or from standby to active.

```
Router# configure terminal
Router(config)# aaa accounting vrrs vrrp-mlist-1 start-stop group radius
Router(config)# aaa attribute list vrrp-1-attr
Router(config-attr-list)# attribute type account-delay "10"
Router(config-attr-list)# exit
Router(config)# vrrs vrrp-name-1
Router(config-vrrs)# accounting delay 10
Router(config-vrrs)# accounting method METHOD1
Router(config-vrrs)# attribute list vrrp-1-attr
```

Example: Confirming Operation of the VRRS Interface-State Plug-in

The following example shows how to confirm the VRRS Interface-State plug-in.

```
Router(config)# interface GigabitEthernet0/0/0
Router(config-if)# no ip address
Router(config-if)# exit
Router(config)# interface GigabitEthernet0/0/0.1
Router(config-if)# encapsulation dot1Q 1 native
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# vrrp 1 name VRRS_NAME_1
Router(config-if)# vrrp 1 ip 172.16.1.254
Router(config-if)# exit
Router(config)# interface GigabitEthernet0/0/0.2
Router(config-if)# encapsulation dot1Q 2
Router(config-if)# ip address 192.168.42.1 255.255.255.0
Router(config-if)# vrrs follow VRRS_NAME_1
Router(config-if)# vrrs interface-state
Router(config-if)# exit
Router(config)# interface GigabitEthernet0/0/0.3
Router(config-if)# encapsulation dot1Q 3
Router(config-if)# ip address 192.168.43.1 255.255.255.0
Router(config-if)# vrrs follow VRRS_NAME_1
Router(config-if)# vrrs interface-state
Router(config-if)# exit
Router(config)# interface GigabitEthernet0/0/0.4
Router(config-if)# encapsulation dot1Q 4
Router(config-if)# ip address 192.168.44.1 255.255.255.0
Router(config-if)# vrrs follow VRRS_NAME_2
Router(config-if)# vrrs interface-state
Router(config-if)# exit
Router# show ip interface brief

Interface                   IP-Address      OK? Method Status            Protocol
GigabitEthernet0/0/0        unassigned      YES NVRAM  up                up
GigabitEthernet0/0/0.1      172.24.1.1      YES manual up                up
GigabitEthernet0/0/0.2      192.168.42.1    YES manual up                up
GigabitEthernet0/0/0.3      192.168.43.1    YES manual up                up
GigabitEthernet0/0/0.4      192.168.44.1    YES manual up                down
! "interface-state" DOWN due to no VRRS server

Router# show vrrs plugin database

VRRS Plugin Database
-----------------------------------------------
Name = VRRS_NAME_1
Server connection = Live
State = Active
MAC addr = 0000.5e00.0101
Test Control = False
Client Handle = 3741319170
Interface list =
                GigE0/0/0.2
                GigE0/0/0.3
-----------------------------------------------
Name = VRRS_NAME_2
Server connection = Disconnected
State = Disabled
MAC addr = 0000.0000.0000
Test Control = False
Client Handle = 603979779
Interface list =
                GigE0/0/0.4
```

Example: Confirming Operation of the VRRS MAC-Address plug-in

The following example shows the confirm operation for the VRRS MAC address plug-in.

```
Router(config)# interface GigabitEthernet0/0/0
```

```
Router(config-if)# no ip address
Router(config-if)# exit
Router(config)# interface GigabitEthernet0/0/0.1
Router(config-if)# encapsulation dot1Q 1 native
Router(config-if)# ip address 172.24.1.1 255.255.255.0
Router(config-if)# vrrp 1 name VRRS_NAME_1
Router(config-if)# vrrp 1 ip 172.24.1.254
Router(config-if)# exit
Router(config)# interface GigabitEthernet0/0/0.2
Router(config-if)# encapsulation dot1Q 2
Router(config-if)# ip address 192.168.42.1 255.255.255.0
Router(config-if)# vrrs follow VRRS_NAME_1
Router(config-if)# vrrs mac-address arp interval 5 duration 360
Router(config-if)# exit
Router(config)# interface GigabitEthernet0/0/0.3
Router(config-if)# encapsulation dot1Q 3
Router(config-if)# ip address 192.168.43.1 255.255.255.0
Router(config-if)# vrrs follow VRRS_NAME_1
Router(config-if)# vrrs mac-address arp interval 5 duration 360
Router(config-if)# exit
Router(config)# interface GigabitEthernet0/0/0.4
Router(config-if)# encapsulation dot1Q 4
Router(config-if)# ip address 192.168.44.1 255.255.255.0
Router(config-if)# vrrs follow VRRS_NAME_2
Router(config-if)# vrrs mac-address arp interval 5 duration 360
Router(config-if)# exit
Router# show ip arp

Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.24.1.1             -    aabb.cc00.fb00  ARPA   GigabitEthernet0/0/0.1
Internet  172.24.1.254          -    0000.5e00.0101  ARPA   GigabitEthernet0/0/0.1
Internet  192.168.42.1          -    0000.5e00.0101  ARPA   GigabitEthernet0/0/0.2
!"mac-address" enabled interfaces using VRRP MAC via VRRS
Internet  192.168.43.1          -    0000.5e00.0101  ARPA   GigabitEthernet0/0/0.3
!"mac-address" enabled interfaces using VRRP MAC via VRRS
Internet  192.168.44.1          -    aabb.cc00.fb00  ARPA   GigabitEthernet0/0/0.4
!"mac-address" disabled interface using BIA
Router# debug arp

ARP packet debugging is on
*Sep 10 20:02:14.971: IP ARP: sent rep src 192.168.42.1 0000.5e00.0101,
              dst 192.168.42.1 ffff.ffff.ffff Ethernet0/0.2
*Sep 10 20:02:14.971: IP ARP: sent rep src 192.168.43.1 0000.5e00.0101,
              dst 192.168.43.1 ffff.ffff.ffff Ethernet0/0.3
*Sep 10 20:02:19.991: IP ARP: sent rep src 192.168.42.1 0000.5e00.0101,
              dst 192.168.42.1 ffff.ffff.ffff Ethernet0/0.2
*Sep 10 20:02:19.991: IP ARP: sent rep src 192.168.43.1 0000.5e00.0101,
              dst 192.168.43.1 ffff.ffff.ffff Ethernet0/0.3
Router# show controller gigabitethernet0/0/0

Interface GigabitEthernet0/0/0
Hardware is AMD Unknown
ADDR: 1EC55D8, FASTSEND: FC286088, MCI_INDEX: 0
DIST ROUTE ENABLED: 0
Route Cache Flag: 11
 amdp2_instance=0x1EC6798, registers=0x1EC5580, ib=0x1EC6D98
 rx ring entries=32, tx ring entries=64
 rxring=0x1EC6DE8, rxr shadow=0x1EC7020, rx_head=0, rx_tail=0
 txring=0x1EC70D8, txr shadow=0x1EC7510, tx_head=0, tx_tail=57, tx_count=57
 running=0, port id=0x5DCF8
 Software MAC address filter(hash:length/addr/mask/hits):
  0x00:  0  ffff.ffff.ffff  0000.0000.0000         0
  0x4C:  0  0100.5e00.0012  0000.0000.0000         0
  0x5F:  0  0000.5e00.0101  0000.0000.0000         0
! Virtual MAC, note for this interface, it may be VRRP that added this MAC.
  0xC0:  0  0100.0ccc.cccc  0000.0000.0000         0
  0xC0:  1  0180.c200.0002  0000.0000.0000         0
  0xC5:  0  0180.c200.0007  0000.0000.0000         0
  0xCC:  0  aabb.cc00.fb00  0000.0000.0000         0
Router# show vrrs plugin database

VRRS Plugin Database
------------------------------------------------
```

```
Name = VRRS_NAME_1
Server connection = Live
State = Active
MAC addr = 0000.5e00.0101
Test Control = False
Client Handle = 3741319170
Interface list =
                GigE0/0/0.2
                GigE0/0/0.3
-------------------------------------------------
Name = VRRS_NAME_2
Server connection = Diconnected
State = Disabled
MAC addr = 0000.0000.0000
Test Control = False
Client Handle = 603979779
Interface list =
                GigE0/0/0.4
```

Where to Go Next

If you want to configure additional VRRP features, see the Configuring VRRP document.

Additional References

Related Documents

Related Topic	Document Title
ANCP	*Access Node Control Protocol Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
VRRP	Configuring VRRP
VRRP and VRRS commands	*Cisco IOS IP Application Services Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
VRRP MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:
	http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2338	Virtual Router Redundancy Protocol
RFC 2787	Definitions of Managed Objects for the Virtual Router Redundancy Protocol
RFC 3768	Virtual Router Redundancy Protocol (VRRP)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.	http://www.cisco.com/cisco/web/support/ index.html
To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.	
Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	

Feature Information for VRRS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 **Feature Information for VRRS**

Feature Name	Releases	Feature Information
Virtual Router Redundancy Service (VRRS)	Cisco IOS XE Release 2.6	Virtual Router Redundancy Service (VRRS) provides a multiclient information abstraction and management service between a First Hop Redundancy Protocol (FHRP), and a registered client.
		The following commands were introduced or modified:
		aaa accounting vrrs , **accounting delay** (VRRS), **accounting method**(VRRS), **attribute list** (VRRS), **debug vrrp all**, **debug vrrp vrrs**, **debug vrrs accounting**, **debug vrrs infra**, **debug vrrs plugin**, **show vrrp**, **show vrrs clients**, **show vrrs clients**, **show vrrs group**, **show vrrs plugin database**, **show vrrs summary**, **vrrp delay**, **vrrs follow**, **vrrp ip**, **vrrs mac-address**, **vrrp name**, **vrrs**.

Glossary

AAA --authentication, authorization, and accounting.

RADIUS --Remote Authentication Dial-In User Service.

virtual router --One or more VRRP routers that form a group. The virtual router acts as the default gateway router for LAN clients. Also known as a VRRP group.

VRRP --Virtual Router Redundancy Protocol. An election protocol that dynamically assigns responsibility for one or more virtual routers to the VRRP routers on a LAN, allowing several routers on a multiaccess link to utilize the same virtual IP address.

VSA --vendor-specific attribute. An attribute that has been implemented by a particular vendor.

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.