High Availability Configuration Guide, 17.2.0

November 2017

Supporting AT&T Vyatta Network Operating System

Contents

Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

About This Guide

This guide describes how to configure AT&T products that run on the AT&T Vyatta Network Operating System (referred to as a virtual router, vRouter, or router in the guide) to provide high availability.

VRRP

VRRP overview

This section presents the following topics with regard to using Virtual Router Redundancy Protocol (VRRP) on the AT&T Vyatta vRouter:

- VRRP protocol
- VRRP group
- Virtual IP address
- VIP Address owner
- Virtual MAC address
- VRRP interface
- VRRP advertisements
- Election of the master router
- Preemption
- VRRP authentication
- VRRP synchronization group
- State-specific filtering
- SNMP support for VRRP
- VRRP version 2 IPv6 support
- VRRP version 3 IPv6 support
- High Availability VPN with VRRP
- Tracking interfaces

VRRP protocol

VRRP is a protocol that allows a cluster of routers to act as one virtual router. VRRP, as specified by RFC 2338, RFC 3768 and RFC 5798, provides router failover services during an interface failure.

On the AT&T Vyatta vRouter, VRRP is supported on a physical Ethernet interface and a VLAN interface (vif). On the AT&T Vyatta vRouter, VRRP version 2 supports IPv4 addresses along with a proprietary mode of VRRP version 2 that supports IPv6 addresses. VRRP version 3 supports IPv4 and IPv6 addresses.

VRRP group

A VRRP group consists of a cluster of interfaces or virtual interfaces that provide redundancy for a primary, or “master,” interface in the group. Each interface in the group is typically on a separate router. Redundancy is managed by the VRRP process on each system.

The VRRP group has a unique numeric identifier and can be assigned up to 20 virtual IP addresses. All interfaces in the group must be assigned the same VRRP group identifier and virtual address; otherwise, they cannot provide redundancy for one another. Interfaces that are being mapped to a virtual address cannot have the same address as the virtual address. VRRP version 3 provides owner mode, which allows interfaces to have the same address as the virtual address.

The addresses of the interfaces that participate in the VRRP group must be on the same subnet for the protocol to work. While it is common for the virtual IP address to be on this subnet, it does not have to be on the subnet. An interface supports multiple VRRP groups.

Virtual IP address

Routers in a VRRP cluster share a virtual IP, or VIP. This sharing provides alternate paths through the network for hosts without explicitly configuring them, and creates redundancy that eliminates any individual router as a

single point of failure in the network. This redundancy is particularly important for default routes, the failure of which could otherwise be a catastrophic event on a network.

A virtual router is an abstract object, managed by the VRRP process, that is defined by its virtual router ID (the group identifier of the set of routers that forms the virtual router). The virtual router presents its VIP to the network. Hosts on the network are configured with a route to direct packets to the VIP rather than to the IP addresses of the real interfaces.

The virtual router uses the group identifier to construct a virtual MAC address from a standard MAC prefix (specified in the VRRP standard) plus the group identifier. ARP or Neighbor Discovery requests for the VIP are resolved to the virtual MAC address, which “floats” from real router to real router, depending on which router is acting as the master router for the virtual router. If the master router fails, the backup router is brought into service by using the virtual MAC address and VIP of the virtual router. In this way, service can transparently continue around a failed gateway to hosts on the LAN, which means that the host default route does not change.

The master router forwards packets for local hosts and responds to ARP requests, neighbor solicitations, ICMP pings, and IP datagrams that are directed to the VIP. Backup routers remain idle, even if healthy. ARP requests, pings, and datagrams that are made to the real IP addresses of interfaces are responded to by the interface in the normal way.

VIP address owner

A router considered the VIP address owner when the primary address that is configured for the interface on which VRRP is running is the VIP. VRRP automatically detects when the router owns the VIP address and advertises the VIP with priority 255.

Versions before release 3.5 IPAO required configuration. From release 3.5 onwards no configuration is required.

Virtual MAC address

RFC 3768 and RFC 5798 define a specific 48-bit MAC address that is to be associated with each VRRP virtual router. The ARP translation for the virtual router IPv4 or IPv6 address points to this MAC address.

The master router uses this well-defined MAC address as the source MAC address of VRRP packets that it sends, in this way teaching switches to send packets for that MAC address to itself. If one master fails and another router takes over as master, it acts in the same way.

Using the well-defined MAC address ensures quick failover of traffic for that MAC address. In addition, the ARP translations of the other hosts and routers on the network do not need to change when a new router takes over as master. This configuration is recommended.

In legacy versions of the AT&T Vyatta vRouter, the VIP was directly linked with the real MAC address of the master router rather than the well-defined MAC address specified by the RFCs. In that implementation, if the master router failed and a new master was elected, the VIP assumed the MAC address of the physical interface of the new master router and the new master notified the network of its MAC address by issuing a gratuitous ARP. This behavior is still the default for VRRP.

Configure RFC-compliant MAC address behavior by setting the **`rfc-compatibility`** option for the VRRP group with interfaces <interface> vrrp vrrp-group <group-id> rfc-compatibility *(page 34)* when you specify VRRP group information for the interface that you are adding to the VRRP group. Note that an AT&T Vyatta vRouter in RFC-compliant mode does not interoperate with an AT&T Vyatta vRouter in noncompliant mode.

Note:

Currently, no VMware settings allow RFC-compliant MAC address behavior for VRRP interfaces. To prevent RFC compatibility issues when running a vRouter on a VMware platform, by default, RFC-compliant MAC address behavior is not enabled on the vRouter. To enable MAC address RFC-compliance, you must run the `set interfaces <interface> vrrp vrrp-group <group-id> rfc-compatibility` command.

The default non-RFC-compliant MAC address behavior is required for any deployment in which VMware provides Layer 2 services because vSwitch does not support true MAC learning and blocks traffic when MAC addresses move between the ports of an L2 switch. In addition, a key component of RFC-compliant

VRRP is that MAC addresses must be able to freely move between routers participating in a group. For more information about MAC support on VMware, refer to VMware documentation.

VRRP interface

When the **rfc-compatibility** option is set, the VRRP process creates a special VRRP interface. The system automatically assigns the VRRP virtual MAC address to this interface. When a new master interface is elected, the system uses the procedure that is described in RFC 3768 to have the new master take over the virtual MAC address.

The VRRP interface that is created by the VRRP process operates in a special pass-through mode. The pass-through mode allows the router to receive packets that are addressed to the well-known VRRP MAC address for a given VRID on the parent interface.

The system automatically generates a name for the VRRP interface by appending vrrp*n* (where *n* is an arbitrary number starting from 1) to the physical interface prefix, for example, dp0vrrp1.

The interface name is based on the highest existing interface; for a scenario in which you configure three groups (dp0vrrp1, dp0vrrp2, and dp0vrrp3), then delete dp0vrrp2 and create another group, the interface name for the new group is dp0vrrp4.

The VRRP interface remains on the system as long as the **rfc-compatibility** option is set, and remains on the system independent of the state of the VRRP instance (backup or master).

VRRP advertisements

To signal that it is still in service, the master interface or vif sends “heartbeat” packets called “advertisements” to the LAN segment, using the IANA assigned multicast addresses for VRRP (224.0.0.18 for IPv4 and FF02:0:0:0:0:0:0:12 for IPv6). These advertisements confirm the health of the master router to backup routers and contain other VRRP information, such as the priority of the master.

If the backup routers do not receive advertisements from the VRRP master router for three advertisement intervals, the master is declared out of service and the VRRP protocol triggers the failover process.

Election of the master router

VRRP dynamically elects the router that is to be the master. In most cases, the master router is simply the router with the interface that has the highest configured priority. If two interfaces have identical priorities, the router with the interface that has the higher IP address is elected master.

If the master interface fails, the interface with the next-highest priority is elected master and assumes the virtual address of the group. If the system is configured to comply with RFC 3768 and RFC 5798, the network continues to use the well-defined MAC address to locate the device that is using the VIP. The new master also ensures that all network devices are notified of the change by sending a gratuitous ARP message.

The priority of the master interface is typically set to 50 greater than the other routers in the set. The backup interface can be left with the default priority; however, if more than one interface is acting as a backup, they could be configured with different priorities.

Preemption

If preemption is enabled, a backup router with a priority that is higher than the current master “preempts” the master and becomes the master itself. The backup router preempts the master by beginning to send its own VRRP advertisements. The master router examines these advertisements and discovers that the backup router has a higher priority than itself. The master then stops sending advertisements, while the backup continues to send, thus, making itself the new master.

Preemption is useful in situations in which a lower-performance backup router becomes the master when a higher-performance router fails. In this case, a new higher-performance router can be brought online, and it automatically preempts the lower-performance backup.

VRRP authentication

If a password is set for VRRP authentication, the authentication type must also be defined. If the password is set and the authentication type is not defined, the system generates an error when you try to commit the configuration.

Similarly, you cannot delete the VRRP password without also deleting the VRRP authentication type. If you do, the system generates an error when you try to commit the configuration.

If you delete both the VRRP authentication password and authentication type, VRRP authentication is disabled.

The IETF decided that authentication is not to be used for VRRP version 3. For more information, refer to RFC 5798.

VRRP synchronization group

Interfaces in a VRRP synchronization group ("sync group") are synchronized such that, if one of the interfaces in the group fails over to backup, all interfaces in the group fail over to backup. For example, in many cases, if one interface on a master router fails, the whole router fails over to a backup router. By assigning all the interfaces in a set of VRRP routers to a sync group, the failure of one interface triggers a failover of all the interfaces in the sync group to the backups that are configured for each interface in the sync group.

If you manually disable an interface that belongs in a VRRP sync group, whether the group is acting as the master or backup router, the state of the VRRP Finite State Machine (FSM) for each of the interfaces within the group changes to the fault state. A transition to the fault state is triggered by one of the following reasons:

- An interface has been administratively shut down.
- A media link failure has been detected.
- An interface hardware failure has been detected.

If the interfaces in the fault state remain in this state because they are not enabled, these interfaces remain in the fault state, thereby, terminating the router failover services that are available through the VRRP feature. For VRRP to function, you must have both master and backup VRRP routers.

> **Note:** If you have to disable an interface that belongs to a VRRP sync group, first create a new sync group that consists of the given interface and the other interface that is paired with it (master or backup) by using the `interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`sync-group`** *group* command. With this step, you are essentially moving the set of interfaces to a new sync group when you create a new sync group because interfaces can belong only to one sync group at any given time. After creating the new sync group, you can proceed with the disabling of the given interface without terminating the router failover services on the sync group in which the interface initially resided.
>
> **Note:** The configuration of a sync group does not take effect or is not reflected in the active state of VRRP until there is more than one group in the sync group.

State-specific filtering

The VRRP specifications require that all packets with a destination MAC address of the virtual MAC address be dropped when the VRRP process is in the backup state. The AT&T Vyatta vRouter implementation of VRRP complies with the specification in this regard.

SNMP support for VRRP

For remote management of VRRP, the AT&T Vyatta vRouter supports the vrrpTrapNewMaster object of RFC 2787 VRRP-MIB and supports the KEEPALIVED-MIB, authored by Vincent Bernat. The KEEPALIVED-MIB extends the keepalived daemon to support the Net-SNMP agentx protocol and provides additional information that is specific to the AT&T Vyatta vRouter implementation, such as state information, sync group state information, and so on.

For a full description of AT&T Vyatta vRouter support for SNMP, refer to the "SNMP" chapter of AT&T Vyatta Network Operating System Remote Management Configuration Guide.

VRRP version 2 IPv6 support

The AT&T Vyatta vRouter supports VRRPv2 with IPV6 addresses. VRRP IPv6 uses link local addresses and supports neighbor-discovery messages between master and backup routers.

Note: Do not configure any IPv4 address on the interface when configuring VRRP IPv6.

When using VRRP IPv6, the primary addresses of the interfaces, virtual IP address, and hosts and server addresses must be configured to be IPv6 addresses. Other interfaces can still have IPv4 VRRP configuration to handle IPv4 traffic.

Note: For VRRP IPv6, the minimum advertise interval is 1 second.

Note: For master and backup router setup, you must configure the priority of the master router to be higher than that of the backup router.

Note: This functionality is not based on standards, and this feature is approaching end of life. We recommend that you move to standards-based VRRP version 3, which has full IPv6 address support.

VRRP version 3 IPv6 support

The AT&T Vyatta vRouter supports full standards-based VRRP version 3, which supports IPv6 addresses and has none of the restrictions of the VRRP version 2 implementation.

High availability VPN with VRRP

The AT&T Vyatta vRouter provides the ability to maintain connectivity through one IPsec tunnel by using a pair of AT&T Vyatta vRouter with VRRP. When one router fails or is brought down for maintenance, the new VRRP master router restores IPsec connectivity between the local and remote networks.

When configuring High Availability VPN with VRRP whenever a VRRP virtual address is added to an AT&T Vyatta vRouter interface, you must reinitialize the IPsec daemon because the IPsec service listens only for connections to the addresses that are present on the AT&T Vyatta vRouter when the IKE service daemon is initialized.

For a pair of AT&T Vyatta vRouter routers with VRRP, the standby router does not have the VRRP virtual address that is present on the device during initialization because the master router may not have that address present. Therefore, to reinitialize the IPsec daemon when a VRRP state transition occurs, run the following command on the master and backup routers:

```
interfaces dataplane interface-name vrrp vrrp-group group-id notify
```

Tracking interfaces

The AT&T Vyatta vRouter provides the ability to track whether a logical interface is up, down, or not present, a VRRP group can only detect the status change of the interface on which the VRRP group is configured. VRRP cannot detect faults on the uplink interface or direct uplink of the master router, so services are interrupted if this uplink interface fails.

The Tracking interface functionality is used to mitigate this, by associating a VRRP group with an interface status. When the uplink interface or direct uplink of the master fails, the priority of the VRRP Master router is adjusted. This adjustment can be used to trigger a switchover to a VRRP backup router, ensuring traffic forwarding can continue.

VRRP configuration examples

This section presents the following topics:

- Basic VRRP version 3 IPv4 configuration
- VRRP Configuration with a synchronization group
- Configuring High Availability VPN with VRRP

Basic VRRP version 3 IPv4 configuration

This section presents the following topics:

- Configuring the master system
- Configuring the backup system

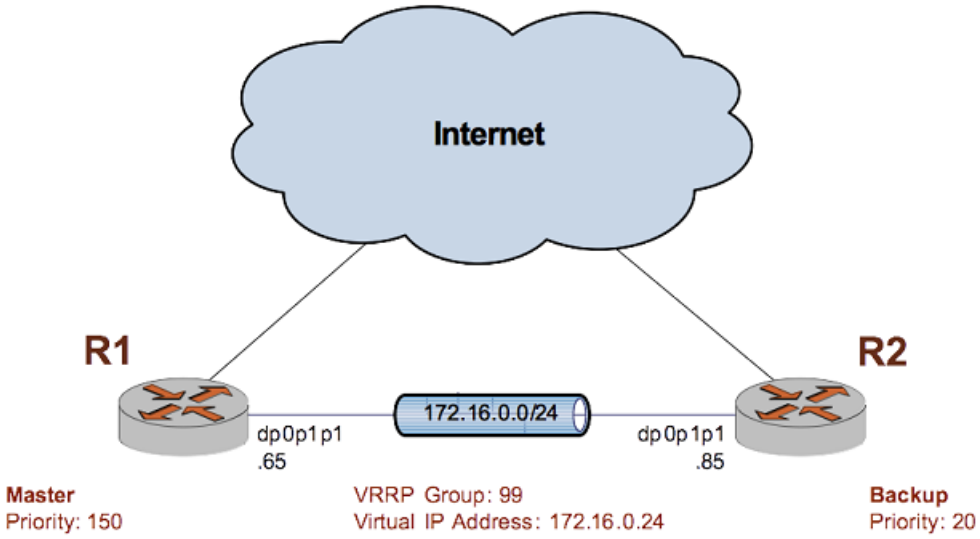
The following sequence sets up a basic VRRP version 3 IPv4 configuration between two AT&T Vyatta vRouter.

In VRRP, consider the following behavior.

- The system that is configured with the highest priority is initially elected the master router. If more than one system has the highest priority, then the system with the highest IP address is elected the master router.
- Enabling preemption allows a higher-priority neighbor to preempt the current master and become master itself.

In this section, a sample configuration is presented for VRRP. When you finish the steps, the system is configured as shown in the following figure.

Figure 1: VRRP sample configuration



Configuring the master system

The following table shows how to enable VRRP version 3 on the dp0p1p1 interface of the master system (R1) and assign it to the 99 VRRP group. The virtual address is 172.16.0.24/24. Preemption is enabled, and R1 is assigned a priority of 150. The VRRP interface is defined to enable RFC-compliant MAC address handling.

To configure the master system for VRRP, perform the following steps in configuration mode.

Table 1: Configuring the master system for VRRP version 3

Step	Command
Create the VRRP configuration node for dp0p1p1 on R1. This configuration enables VRRP on that interface. Assign the VRRP group and protocol version.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99` `vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 version 3`

Step	Command
Specify the virtual address of the VRRP group.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 virtual-address 172.16.0.24`
Enable RFC-compliant MAC address handling and create the VRRP interface.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 rfc-compatibility`
Enable preemption.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 preempt true`
Set the priority of this system to 150.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 priority 150`
Commit the configuration.	`vyatta@R1# commit`

Display the configuration.	`vyatta@R1# show interfaces dataplane dp0p1p1 vrrp`
	
	`vrrp {`
	`  vrrp-group 99 {`
	`  priority 150`
	`  rfc-compatibility`
	`  version 3`
	`  virtual-address 172.16.0.24`
	`  }`
	`}`

Configuring the backup system

The following table shows how to enable VRRP version 3 on the dp0p1p1 interface of the backup system (R2), and assign it to the 99 VRRP group. The virtual address is the same as that for R1: 172.16.0.24/24. Preemption is enabled, and R2 is assigned a priority of 20. This priority is lower than the priority of R1, so R1 is the master and R2 is the backup under ordinary circumstances.

To configure the backup system for VRRP, perform the following steps in configuration mode.

Table 2: Configuring the backup system for VRRP version 3

Step	Command
Create the VRRP configuration node for dp0p1p1 of R2. This configuration enables VRRP on that interface. Assign the VRRP group and protocol version.	`vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99` `vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 version 3`
Specify the virtual address of the VRRP group.	`vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 virtual-address 172.16.0.24`
Enable preemption.	`vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 preempt true`

Step	Command
Enable RFC-compliant MAC address handling and create the VRRP interface.	`vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 rfc-compatibility`
Set the priority of this system to 20. This priority is lower than that set for R1, so R1 becomes the master.	`vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 priority 20`
Commit the configuration.	`vyatta@R2# commit`

Display the configuration.	`vyatta@R2# show interfaces dataplane dp0p1p1 vrrp`
	
	`vrrp {`
	`  vrrp-group 99 {`
	`  priority 20`
	`  rfc-compatibility`
	`  version 3`
	`  virtual-address 172.16.0.24`
	`  }`
	`}`

Basic VRRP version 3 IPv6 configuration

This section presents the following topics:

- Configuring the master system
- Configuring the backup system

Note: We strongly recommend the use of VRRP version 3 for IPv6 functionality.

The following is assumed for the examples in these topics:

- IPv6 address of master system (R1): 3003::300:1/64
- IPv6 address of backup system (R2): 3003::300:3/64
- Virtual IP address: 3003::300:2/64

Configuring the master system

The following table shows how to enable VRRP version 3 on the dp0p1p1 interface of R1 and assign the interface to VRRP group 41. R1 is assigned a priority of 150. The VRRP interface is defined to enable RFC 3768-compliant MAC address handling.

To configure the master system for VRRP version 3, perform the following steps in configuration mode.

Table 3: Configuring the master system for VRRP version 3

Step	Command
Set the IPv6 address of the dp0p1p1 interface for R1.	`vyatta@R1# set interfaces dataplane dp0p1p1 address 3003::300:1/64`
Create the VRRP configuration node for dp0p1p1 on R1. Enable VRRP on dp0p1p1. Assign the VRRP group and protocol version.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41` `vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41 version 3`

Step	Command
Specify the virtual address of the VRRP group.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41 virtual-address 3003::300:2`
Specify the IPv6 link-local virtual address.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41 virtual-address fe80::2`
Enable RFC 3768-compliant MAC address handling and create the VRRP interface.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41 rfc-compatibility`
Set the priority of this system to 150.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41 priority 150`
Commit the configuration.	`vyatta@R1# commit`

Display the configuration.	`vyatta@R1# show interfaces dataplane dp0p1p1 vrrp`
	
	`vrrp {`
	`  vrrp-group 41 {`
	`  priority 150`
	`  rfc-compatibility`
	`  version 3`
	`  virtual-address 3003::300:2`
	`  }`
	`}`

Configuring the backup system

The following table shows how to enable VRRP version 3 on the dp0p1p1 interface of R2 and assigns dp0p1p1 to VRRP group 41. The virtual address is the same as that for R1: 3003::300:2/64. R2 has the default priority (100), which is lower than the priority of R1 (150), so R1 is the master and R2 is the backup under ordinary circumstances.

To configure the backup system for VRRP version 3, perform the following steps in configuration mode.

Table 4: Configuring the backup system for VRRP version 3

Step	Command
Set the IPv6 address of the dp0p1p1 interface for R2.	`vyatta@R2# set interfaces dataplane dp0p1p1 address 3003::300:3/64`
Create the VRRP configuration node for dp0p1p1 of R2. This command enables VRRP on that interface. Assign the VRRP group and protocol version.	`vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41` `vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41 version 3`
Specify the virtual address of the VRRP group.	`vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41 virtual-address 3003::300:2`

Step	Command
Enable RFC 3768-compliant MAC address handling and create the VRRP interface.	`vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 41 rfc-compatibility`
Commit the configuration.	`vyatta@R2# commit`

Display the configuration.	`vyatta@R2# show interfaces dataplane dp0p1p1 vrrp`
	
	`vrrp {`
	`  vrrp-group 41 {`
	`  rfc-compatibility`
	`  version 3`
	`  virtual-address 3003::300:2`
	`  }`
	`}`

VRRP configuration with a synchronization group

This section presents the following topics:

- Configuring the master system
- Configuring the backup system

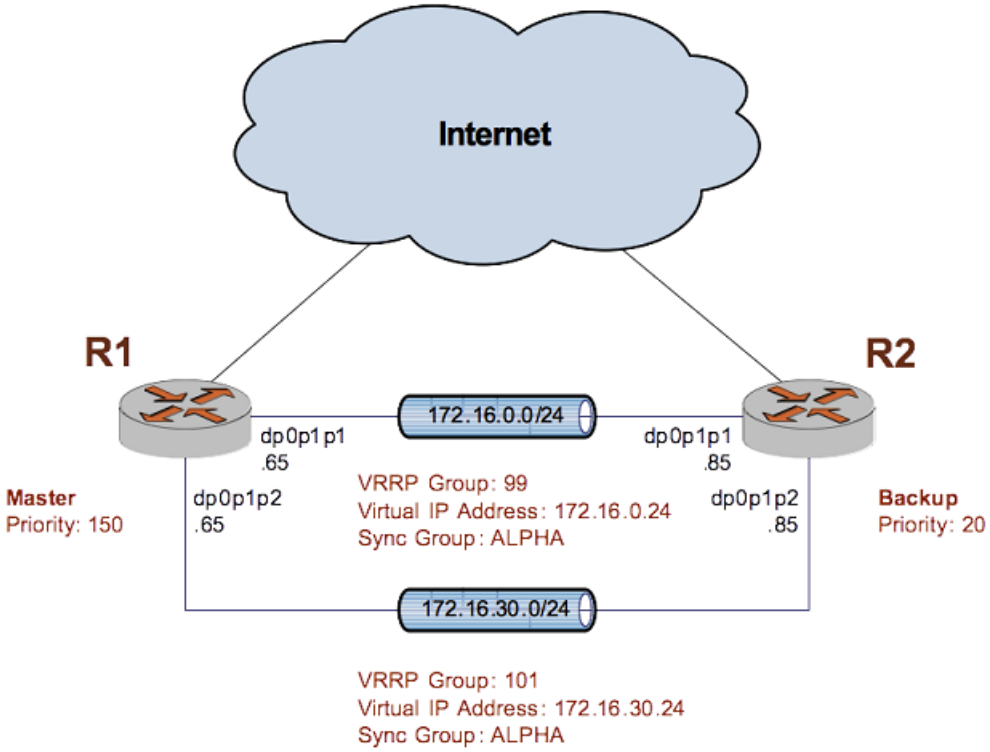
This example builds on the previous example by adding an interface to each system, specifying a VRRP group and virtual IP address for these interfaces, and then including all interfaces in a sync group so that if one of the interfaces on the master fails, all interfaces on the master pass control to interfaces on the backup system.

Note: The outcome of the disabling of an interface that belongs to a VRRP sync group is that router failover services that are available through the VRRP feature are terminated for the given group. For more information on a work-around, refer to VRRP Synchronization Group *(page 12)*.

Note: To avoid state-failover oscillation between peers, AT&T recommends not enabling preemption for sync groups.

When you finish the steps, the system is configured as shown in the following figure.

Figure 2: VRRP with a sync group



Configuring the master system

The following table shows how to configure the master system for VRRP with a sync group. To configure the system in this way, perform the following steps in configuration mode.

Table 5: Configuring the master system for VRRP with a sync group

Step	Command
Add the sync group configuration to the existing configuration for the 99 VRRP group on dp0p1p1.	`vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 sync-group ALPHA` `vyatta@R1# set interfaces dataplane dp0p1p1 vrrp vrrp-group 99 sync-group ALPHA version 3`

Display the VRRP configuration on dp0p1p1.	`vyatta@R1# show interfaces dataplane dp0p1p1 vrrp`
	
	`vrrp {`
	`  vrrp-group 99 {`
	`  }`
	`  priority 150`
	`  sync-group ALPHA`
	`  version 3`
	`  virtual-address 172.16.0.24`
	`  }`
	`}`

Step	Command
Create the VRRP configuration node for dp0p1p2 on R1. This configuration enables VRRP on that interface. Assign the VRRP group.	`vyatta@R1# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101` `vyatta@R1# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 version 3`
Specify the virtual address of the VRRP group.	`vyatta@R1# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 virtual-address 172.16.30.24`
Enable RFC-compliant MAC address handling and create the VRRP interface.	`vyatta@R1# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 rfc-compatibility`
Set the priority of this system to 150.	`vyatta@R1# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 priority 150`
Add the VRRP group on dp0p1p2 to the sync group.	`vyatta@R1# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 sync-group ALPHA`
Commit the configuration.	`vyatta@R1# commit`

Display the configuration.	`vyatta@R1# show interfaces dataplane dp0p1p2 vrrp`
	
	`vrrp {`
	`  vrrp-group 101 {`
	`  priority 150`
	`  rfc-compatibility`
	`  sync-group ALPHA`
	`  version 3`
	`  virtual-address 172.16.30.24`
	`  }`
	`}`

Configuring the backup system

The following table shows how to configure the backup system for VRRP with a sync group. To configure the system in this way, perform the following steps in configuration mode.

Table 6: Configuring the backup system for VRRP with a sync group

Step	Command
Add the sync group configuration to the existing configuration for the 99 VRRP group on dp0p1p1.	`vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 101 sync-group ALPHA` `vyatta@R2# set interfaces dataplane dp0p1p1 vrrp vrrp-group 101 sync-group ALPHA version 3`

Step	Command

Display the VRRP configuration on dp0p1p1.	`vyatta@R2# show interfaces dataplane dp0p1p1 vrrp`
	
	`vrrp {`
	`  vrrp-group 101 {`
	`  interface {`
	`  }`
	`  priority 20`
	`  sync-group ALPHA`
	`  version 3`
	`  virtual-address 172.16.0.24`
	`  }`
	`}`

Create the VRRP configuration node for dp0p1p2 on R2. This configuration enables VRRP on that interface. Assign the VRRP group.	`vyatta@R2# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101` `vyatta@R2# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 version 3`
Specify the virtual address of the VRRP group.	`vyatta@R2# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 virtual-address 172.16.30.24`
Enable RFC 3768-compliant MAC address handling and create the VRRP interface.	`vyatta@R2# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 rfc-compatibility`
Set the priority of this system to 20.	`vyatta@R2# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 priority 20`
Add the VRRP group on dp0p1p2 to the sync group.	`vyatta@R2# set interfaces dataplane dp0p1p2 vrrp vrrp-group 101 sync-group ALPHA`
Commit the configuration.	`vyatta@R2# commit`

Display the configuration.	`vyatta@R2# show interfaces dataplane dp0p1p2 vrrp`
	
	`vrrp {`
	`  vrrp-group 101 {`
	`  priority 20`
	`  rfc-compatibility`
	`  sync-group ALPHA`
	`  version 3`
	`  virtual-address 172.16.30.24`
	`  }`
	`}`

Configuring high availability VPN with VRRP

To configure high availability VPN with VRRP on an AT&T Vyatta vRouter, perform the following steps in configuration mode.

Table 7: Configuring high availability VPN with VRRP

Step	Command
Add an IPsec notify option to the keepalive configuration file.	`vyatta@R1# set interfaces dataplane dp02p2 vrrp vrrp-group 10  notify ipsecvyatta@R1# set interfaces dataplane dp02p2 vrrp vrrp-group 10  notify ipsec`

Display the configuration.	`vyatta@R1# show interfaces dataplane dp02p2 vrrp`
	
	`vrrp {`
	`  vrrp-group 10 {`
	`  notify {`
	`  ipsec`
	`  }`
	`  priority 100`
	`  version 3`
	`  virtual-address 192.168.10.20`
	`  }`
	`  }`

Configuring interface tracking

To configure interface tracking with VRRP on AT&T Vyatta vRouter, perform the following steps in configuration mode.

Table 8: Configuring interface tracking with VRRP

Step	Command
Configure to decrement the priority of the group 10 VRRP on this router when the dp0s10 interface goes down.	`vyatta@R1# set int dataplane dp02p2 vrrp vrrp-group 10 track dp0s10 weight type decrement` `vyatta@R1# set int dataplane dp02p2 vrrp vrrp-group 10 track dp0s10 weight value 100`

Display the configuration.	`vyatta@R1# show interfaces dataplane dp02p2 vrrp`
	
	`vrrp {`
	`  vrrp-group 10 {`
	`  track-interface dp0s10 {`
	`  weight {`
	`  type decrement`
	`  value 100`
	`  }`
	`  }`
	`  version 3`
	`  virtual-address 91.0.0.100`
	`  }`
	`  }`

Configuring symmetric routing with VRRP and BGP

In this topology, VRRP (Virtual Router Redundancy Protocol) runs on the local internal LAN network. BGP runs between the ISP and the VRRP routers. The northbound traffic from the local internal LAN network routers does not use the same path as does the southbound (incoming) traffic from the ISP network routers. This mode of routing is referred to as asymmetric routing.

Note:

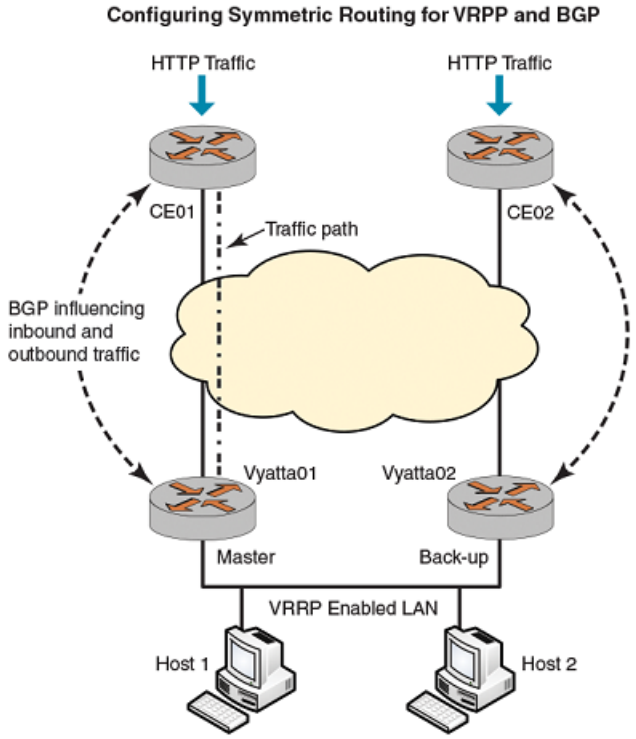
If you are upgrading the AT&T Vyatta vRouter to the 4.0 release from any previous release, ensure that unique VRRP group IDs are used across interfaces to avoid conflicting configurations among those interfaces. Furthermore, by employing only the following command, use unique VRRP group IDs for the VRRP groups that are tracked by BGP.

```
protocols bgp asn neighbor id interface interface-name vrrp-failover vrrp-group vrrp-group-id
```

. VRRP groups that are not tracked do not require unique IDs within the vRouter.

This example explains how to set up symmetric routing using BGP and VRRP.

Figure 3: Configuring symmetric routing for VRPP and BGP



The following components of the topology are shown in the preceding figure.

- Vyatta01 and Vyatta02 are the two routers in the local internal LAN network that are running VRRP. Vyatta01 is the master router and Vyatta02 is the backup router.
- CE01 and CE02 are the ISP gateway routers that are running BGP.

Both the VRRP master and backup routers, Vyatta01 and Vyatta02, are connected to an ISP gateway router, by using BGP. This topology provides two possible outgoing and incoming paths for the traffic.

For example, the HTTP traffic passes from CE01 to the master router, Vyatta01. To make routing symmetric, the data sent from Vyatta01 must pass through CE01.

When VRRP failover occurs, the flow of northbound traffic switches to the backup router, Vyatta02 which becomes the new master router.

After the switchover, the southbound traffic coming from the ISP network to the LAN network should pass traffic through Vyatta02 to achieve symmetric routing.

BGP recognizes the multiple paths to send traffic. BGP determines the best path to send traffic by using the path selection algorithm. Modify the BGP path selection process in ISP routers by influencing the following parameters:

- Multi-exit discriminator (MED)
- AS PATH Length
- BGP route map for VRRP failover

To enable BGP updates to ISP routers for influencing the BGP attributes perform following steps in configuration mode.

Table 9: Configuring VRRP settings

Router	Step	Commands
Vyatta01 and Vyatta02	Configure BGP as a client to notify VRRP state changes.	`vyatta@R1#set interfaces dataplane interface-name vrrp vrrp-group vrrp group-id notify bgp`
Vyatta01 and Vyatta02	Set MED configuration for the neighbor.	`vyatta@R1# set protocols bgp asn neighbor id interface interface-name vrrp-failover vrrp-group vrrp-group-id med med-value`
Vyatta01 and Vyatta02	Set prepend-as configuration for the neighbor.	`vyatta@R1# set protocols bgp asn neighbor id interface interface-name vrrp-failover vrrp-group vrrp-group-id prepend-as as-path-string`
Vyatta01 and Vyatta02	Set route-map configuration for the neighbor.	`vyatta@R1#set protocols bgp asn neighbor id interface interface-name vrrp-failover vrrp-group vrrp-group-id route-map route-map-name`

VRRP Commands

interfaces <interface> vrrp vrrp-group <group-id>

Assigns an interface to a VRRP group.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id*

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id*

interface
The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The identifier of a VRRP group to which the interface is to belong. The identifier ranges from 1 through 255.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
        }
    }
}
```

Use this command to assign an interface to a VRRP group. An interface or virtual interface (vif) can belong to more than one VRRP group.

Use the `set` form of the command to assign an interface to a VRRP group.

Use the `delete` form of the command to remove an interface from a VRRP group.

Use the `show` form of the command to view VRRP group configuration settings for an interface.

interfaces <interface> vrrp vrrp-group <group-id> advertise-interval <interval>

Sets the advertisement interval for a VRRP version 2 group on an interface.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`advertise-interval`** *interval*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`advertise-interval`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`advertise-interval`**

The master router sends VRRP advertisements at one-second intervals.

interface

The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id

The identifier of a VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

interval

The interval, in seconds, between VRRP advertisement packets. All interfaces in a given VRRP group must use the same advertisement interval. The interval ranges from 1 through 255. The default interval is 1.

Note: The default interval value is 1.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            advertise-interval interval
        }
    }
}
```

Use this command to set the advertisement interval for a VRRP version 2 group on an interface.

Use the `set` form of the command to set the advertisement interval for a VRRP version 2 group on an interface.

Use the `delete` form of the command to restore the default advertisement interval of 1000 ms (1 second) for a VRRP version 2 group on an interface, which means that the master router sends advertisement packets at one-second intervals.

Use the `show` form of the command to display the advertisement interval for a VRRP version 2 group on an interface.

interfaces dataplane <interface> vrrp vrrp-group <group-id> fast-advertise-interval <interval>

Sets the advertisement interval for a VRRP version 3 group on an interface.

Syntax:

`set interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`fast-advertise-interval`** *interval*

Syntax:

`delete interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`fast-advertise-interval`**

Syntax:

`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`advertise-interval`**

The master router sends advertisement packets in VRRP version 3 at intervals of 1000 ms (1 second).

interface

The type keyword and identifier of an interface data plane. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id

The identifier of a VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

interval

The interval between advertisement packets in VRRP version 3. All interfaces in a given VRRP group must use the same advertisement interval. The interval ranges from 10 through 40950 milliseconds. The default interval is 1000 milliseconds.

The VRRPv3 protocol operates in centiseconds and the millisecond value entered with the command is rounded up to the nearest 10 centiseconds for the protocol.

Configuration mode

```
interfaces dataplane interface {
    vrrp {
        vrrp-group group-id {
            fast-advertise-interval interval
        }
    }
}
```

Use this command to set the advertisement interval for a VRRP version 3 group on an interface.

Use the `set` form of the command to set the advertisement interval for a VRRP version 3 group on an interface.

Use the `delete` form of the command to restore the default advertisement interval of 1000 ms (1 second) for a VRRP version 3 group on an interface, which means that the master router sends advertisement packets ranging from 10 to 40950 milliseconds.

Use the `show` form of the command to display the advertisement interval for a VRRP version 3 group on an interface.

While it is possible to configure a value as low as 10ms, pay attention to the capability of the network and hardware the hello packets are running on. If values are too low, then the network may become unstable. Values of hundreds of milliseconds tend to work in most environments.

interfaces <interface> vrrp vrrp-group <group-id> authentication password <pwd>

Sets the VRRP version 2 authentication password for a VRRP group on an interface.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`authentication password`** *pwd*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`authentication password`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`authentication password`**

If a password is not set, an interface is not required to authenticate itself to the VRRP group.

interface
The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The identifier of a VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

pwd
The password the interface is to use to authenticate itself as a member of the VRRP group.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            authentication {
                password pwd
            }
        }
    }
}
```

Use this command to set a password for a VRRP group on an interface.

If a password is set for VRRP authentication, the authentication type (AH or plain text-password) must also be defined. If the password is set and the authentication type is not defined, the system generates an error when you try to commit the configuration.

Use the `set` form of the command to specify the VRRP authentication password for a VRRP group on an interface.

Use the `delete` form of the command to delete the VRRP authentication password for a VRRP group on an interface.

- You cannot delete the VRRP password without also deleting the VRRP authentication type. If you attempt to delete the password, the system generates an error when you try to commit the configuration.
- If you delete both the VRRP authentication password and authentication type, VRRP authentication is disabled on the interface.

Use the `show` form of the command to view the VRRP authentication password for a VRRP group on an interface.

interfaces <interface> vrrp vrrp-group <group-id> authentication type <type>

Specifies the VRRP version 2 authentication type for a VRRP group on an interface.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`authentication type`** *type*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`authentication type`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`authentication type`**

An interface is not required to authenticate itself to the VRRP group.

interface
The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The identifier of a VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

type
The type of authentication to be used. The type is either of the following:

- ah: IP Authentication Header (AH) protocol
- plaintext-password: Plain-text password authentication

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            authentication {
                type type
            }
        }
    }
}
```

Use this command to set the VRRP version 2 authentication type for a VRRP group on an interface.

If the authentication type is set for VRRP authentication, a password must also be specified. If the authentication type is defined and a password is not set, the system generates an error when you try to commit the configuration.

Use the `set` form of the command to specify the VRRP version 2 authentication type for a VRRP group on an interface.

Use the `delete` form of the command to delete the VRRP version 2 authentication type for a VRRP group on an interface.

- You cannot delete the VRRP authentication type without also deleting the VRRP password. If you do delete the authentication type, the system generates an error when you try to commit the configuration.
- If you delete both the VRRP authentication password and authentication type, VRRP authentication is disabled on the interface.

Use the `show` form of the command to view the VRRP version 2 authentication type for a VRRP group on an interface.

interfaces <interface> vrrp vrrp-group <group-id> description <desc>

Records a brief description of a VRRP group.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`description`** *desc*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`description`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`description`**

interface
The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The identifier of a VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

desc
A description of the VRRP group on a virtual interface (vif).

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            description desc
        }
    }
}
```

Use this command to record a brief description of a VRRP group.

Use the `set` form of the command to record a description of a VRRP group.

Use the `delete` form of the command to delete the description of a VRRP group.

Use the `show` form of the command to display the description of a VRRP group.

interfaces <interface> vrrp vrrp-group <group-id> disable

Disables a VRRP group without discarding VRRP configuration for the group.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`disable`**

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`disable`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id*

A VRRP group is enabled.

interface
The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The identifier of a VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            disable
        }
    }
}
```

Use this command to disable a VRRP group on a data plane interface without deleting VRRP configuration for the group. Later, you can re-enable the VRRP group by deleting the command.

Use the `set` form of the command to disable a VRRP group.

Use the `delete` form of the command to re-enable a VRRP group.

Use the `show` form of the command to view VRRP group configuration.

interfaces <interface> vrrp vrrp-group <group-id> hello-source-address <addr>

Specifies the source address for VRRP hello packets.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`hello-source-address`** *addr*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`hello-source-address`** *addr*

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`hello-source-address`**

The IP address of the interface is used as the source of VRRP hello packets.

interface
The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The identifier of a VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

addr
The IP address to use as the VRRP source address when sending VRRP hello packets. The format of the address is *ipv4-addr* or *IPv6-addr* . The address must already be defined on an interface.

Configuration mode

```
interfaces interface {
```

```
    vrrp {
        vrrp-group group-id {
            hello-source-address addr
        }
    }
}
```

Use this command to specify the source address for VRRP hello packets. This address is typically used when an address other than the default address for the interface is required. Note that the address must already be defined on an interface.

Use the `set` form of the command to specify the source address of VRRP hello packets.

Use the `delete` form of the command to restore the default source address, which is the IP address of the interface.

Use the `show` form of the command to view the configuration.

interfaces <interface> vrrp vrrp-group <group-id> interface description <descr>

Specifies a description for a VRRP interface.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`interface description`** *descr*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`interface description`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`interface description`**

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

descr
A mnemonic name or description of the VRRP interface.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            interface {
                description desc
            }
        }
    }
}
```

Use this command to specify a description of a VRRP interface.

Use the `set` form of this command to specify a description of a VRRP interface.

Use the `delete` form of this command to remove the description of a VRRP interface.

Use the `show` form of this command to display the description of a VRRP interface.

interfaces <interface> vrrp vrrp-group <group-id> preempt <preempt>

Enables or disables preemption for a VRRP group on an interface.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`preempt`** *preempt*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`preempt`**

Syntax:
`show interfaces` *interface* **`vif vrrp vrrp-group`** *group-id* **`preempt`**

Preemption is enabled.

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

preempt
Allows a higher-priority VRRP backup router to assert itself as master over a lower-priority router. Supported values are as follows:

- **`true`**: Allow the master router to be preempted by a backup router with higher priority.
- **`false`**: Do not allow the master router to be preempted by a backup router with higher priority.

The default is **`true`**; that is, the master router can be preempted by a backup router with higher priority.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            preempt preempt
        }
    }
}
```

Use this command to enable or disable preemption for a VRRP group on an interface.

If preemption is enabled, a backup router with a higher priority than the current master “preempts” the master and becomes the master itself.

A backup router preempts the master by beginning to send its own VRRP advertisements. The master router examines these advertisements and discovers that the backup router has a higher priority than itself. The master then stops sending advertisements, while the backup continues to send, thus, making itself the new master.

Preemption is useful when a lower-performance backup router becomes the master because a higher-performance router fails. In this case, a new higher-performance router can be brought online, and it automatically preempts the lower-performance backup.

Use the `set` form of the command to enable or disable preemption delay for a VRRP group on an interface.

Use the `delete` form of the command to restore preemption for a VRRP group on an interface.

Use the `show` form of the command to view VRRP preemption configuration on an interface.

interfaces <interface> vrrp vrrp-group <group-id> preempt-delay <delay>

Sets the preemption delay for a VRRP group on an interface.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`preempt-delay`** *delay*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`preempt-delay`**

Syntax:
`show interfaces` *interface* **`vif vrrp vrrp-group`** *group-id* **`preempt-delay`**

A router that is preempting another router does not wait.

interface
The type of interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group that is being configured. The identifier ranges from 1 through 255.

delay
The amount of time, in seconds, to postpone preemption. The delay ranges from 0 through 1000 (16.67 minutes.), where 0 means no delay. The default delay is 0.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            preempt-delay delay
        }
    }
}
```

Use this command to set the preemption delay for a VRRP group on an interface. The preemption delay is the amount of time a router must wait before preempting a lower-priority VRRP router and becoming the master.

Use the `set` form of the command to set the preemption for a VRRP group on an interface.

Use the `delete` form of the command to restore the default preemption delay, which is 0 seconds, for a VRRP group on an interface.

Use the `show` form of the command to display the preemption delay for a VRRP group on an interface.

interfaces <interface> vrrp vrrp-group <group-id> priority <priority>

Sets the priority of an interface within a VRRP group.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`priority`** *priority*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`priority`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`priority`**

The default priority is 100.

interface
The type of interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group that is being configured. The identifier ranges from 1 through 255.

priority
The priority with which this interface is considered for election as the master within the VRRP group. The higher the configured number, the higher the priority.

The priority for a VRRP backup router ranges from 1 through 254. The VRRP master router must have the highest priority, which is typically set to 50 greater than any backup router. The address owner is typically set to 255. The default priority is 100.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            priority priority
        }
    }
}
```

Use this command to set the priority of an interface within a VRRP group. This priority determines the likelihood that it is elected the master router in a cluster of VRRP routers.

The master interface in the VRRP group is elected the master based on its priority in which the higher the configured number, the higher the priority. If the master interface fails, the interface with the next highest priority is elected master and assumes the virtual address of the group. The new master notifies the network of its MAC address by sending a gratuitous ARP message.

The priority of the master interface is typically set to 255. The backup interface can be left with the default priority; however, if more than one interface is acting as backup, the interfaces should be configured with different priorities.

Use the `set` form of the command to specify the priority of an interface within a VRRP group.

Use the `delete` form of the command to remove the priority of an interface within a VRRP group.

Use the `show` form of the command to view the priority of an interface within a VRRP group.

interfaces <interface> vrrp vrrp-group <group-id> rfc-compatibility

Creates a VRRP interface, which enables RFC-compliant MAC address behavior.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`rfc-compatibility`**

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`rfc-compatibility`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`rfc-compatibility`**

If this option is not configured, the system uses legacy (non-RFC 3768-compliant) MAC address behavior when a new master is elected. For details, refer to Virtual MAC Address *(page 10)*.

interface
The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id

The identifier of a VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

In addition to the parameters shown here, a VRRP interface can be configured in the same way as the parent interface.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            rfc-compatibility
        }
    }
}
```

Use this command to enable RFC 3768-compliant MAC address behavior when a new master router is elected. Setting this option defines a VRRP interface for the VRRP group that you are configuring.

RFC 3768 defines a specific 48-bit MAC address that is to be associated with each VRRP virtual router. The ARP translation of the IPv4 or IPv6 address for the virtual router points to this MAC address.

The master router uses this well-defined MAC address as the source MAC address of VRRP packets that it sends, in this way teaching switches to send packets for that MAC address to itself. If one master router fails and another router takes over as the master, it acts in the same way.

When a VRRP interface is configured for a VRRP group, the system assigns the well-defined MAC address to the VRRP interface according to RFC 3768. Using the well-defined MAC address ensures quick failover of traffic for that MAC address. In addition, the ARP translations of the other hosts and routers on the network do not need to change when a new router takes over as master router. This configuration is recommended.

When configured, the system automatically assigns the VRRP virtual MAC address to the VRRP interface. When a new master router is elected, the system uses the procedure described in RFC 3768 to have the new master take over the virtual MAC address. The VRRP interface remains on the system as long as the configuration does, independent of whether the VRRP instance is in a backup or master state.

The system automatically generates a name for the VRRP interface by appending vrrp*n* (where *n* is an arbitrary number starting from 1) to the physical interface prefix, for example, dp0vrrp1.

The interface name is based on the highest existing interface; for a scenario in which you configure three groups (dp0vrrp1, dp0vrrp2, and dp0vrrp3), then delete dp0vrrp2 and create another group, the interface name for the new group is dp0vrrp4.

The VRRP interface remains on the system as long as the **`rfc3768-compatibility`** option is set, and remains on the system independent of the state of the VRRP instance (backup or master).

The VRRP interface that is created by the VRRP process operates in a special “pass-through” mode. The pass-through mode allows the router to receive packets addressed to the well-known VRRP MAC address for a given VRID on the parent interface. The VRRP interface is used only to send VRRP advertisement packets when its associated VRRP group is acting as the master router for the group.

Use the `set` form of this command to direct the system to use RFC 3768-compliant MAC address handling when a new master is elected and create a VRRP interface.

Use the `delete` form of this command to remove the VRRP interface and restore the legacy (non-compliant) VRRP MAC address behavior.

Use the `show` form of the command to view VRRP interface configuration.

> **Note:** The default behavior of nonRFC#compliant MACs is necessary for any environment in which VMware provides Layer 2 services because the vSwitch product does not support true MAC learning and blocks traffic when virtual MAC addresses move ports unless the vSwitch is in promiscuous mode. For the latest product support information, refer to VMware documentation.

interfaces <interface> vrrp vrrp-group <group-id> run-transition-scripts

Specifies a script to run when the VRRP on an interface changes state.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`run-transition-scripts`** [**`backup`** | **`fault`** | **`master`**] *script*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`run-transition-scripts`** [**`backup`** | **`fault`** | **`master`**]

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`run-transition-scripts`** [**`backup`** | **`fault`** | **`master`**]

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

`backup` ***script***
Specifies the name of the executable script to run during VRRP state transition to the backup state.

`fault` ***script***
Specifies the name of the executable script to run during VRRP state transition to the fault state.

`master` ***script***
Specifies the name of the executable script to run during VRRP state transition to the master state.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            run-transition-scripts {
                backup script
                fault script
                master script
            }
        }
    }
}
```

Use this command to specify a script to run when the VRRP group on an interface changes state. The state is backup, fault, or master. A script file is assumed to be in the `/config/scripts` directory unless an absolute path is specified.

Use the `set` form of the command to specify a script to run when the VRRP group on an interface changes state.

Use the `delete` form of the command to stop the script from being run when an interface changes states.

Use the `show` form of the command to view the configuration.

interfaces <interface> vrrp vrrp-group <group-id> sync-group <group>

Assigns an interface to a VRRP sync group on a router.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`sync-group`** *group*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`sync-group`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`sync-group`**

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

group
A text string defining the name of a sync group.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            sync-group group
        }
    }
}
```

Use this command to assign an interface to a VRRP sync group on a router.

Interfaces in a sync group are synchronized such that, if one of the interfaces in the group fails over to backup, all interfaces in the group fail over to backup.

For example, in many cases, if one interface on a master router fails, the whole router fails over to a backup router. By assigning all the interfaces on the master to a sync group, the failure of one interface triggers a failover of all the interfaces in the sync group to the backup that is configured for the interface.

Use the `set` form of the command to assign an interface to a VRRP sync group on a router.

Use the `delete` form of the command to remove an interface from a VRRP sync group on a router.

Use the `show` form of the command to display the VRRP sync group on a router for an interface.

interfaces dataplane <interface> vrrp vrrp-group <group-id> notify ipsec

Notifies the IPSec daemon when VRRP on an interface changes state.

Syntax:
`set interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`notify ipsec`**

Syntax:
`delete interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`notify ipsec`**

Syntax:
`show interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id*

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

Configuration mode

```
interfaces {
    dataplane interface {
        vrrp {
            vrrp-group group-id {
                notify {
                    ipsec
                }
            }
        }
    }
}
```

Use the `set` form of the command to notify the IPSec daemon when VRRP on an interface changes state.

Use the `delete` form of the command to remove notification of the IPSec daemon when VRRP on an interface changes state.

Use the `show` form of the command to display whether the IPSec daemon is notified when VRRP on an interface changes state, which lets you determine if High Availability VPN is enabled.

interfaces dataplane <interface> vrrp vrrp-group <group-id> notify bgp

Configures BGP as a client to notify VRRP state changes.

Syntax:
`set interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`notify`** *bgp*

Syntax:
`delete interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`notify`** *bgp*

Disabled

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

Configuration mode

```
interfaces {
    dataplane interface {
        vrrp {
            vrrp-group group-id {
                notify {
                    bgp
                }
            }
        }
    }
}
```

Use the **set** form of the command to notify BGP when VRRP on an interface changes stat

Use the `delete` form of the command to remove notification of BGP when VRRP on an interface changes state.

Use the `show` form of the command to display whether BGP is notified when VRRP on an interface changes state, which lets you determine if High Availability VPN is enabled.

interfaces <interface> vrrp vrrp-group <group-id> virtual-address <addr>

Sets the virtual IP address or network address of a VRRP group on an interface.

Syntax:
`set interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`virtual-address`** *addr*

Syntax:
`delete interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`virtual-address`**

Syntax:
`show interfaces` *interface* **`vrrp vrrp-group`** *group-id* **`virtual-address`**

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

addr
The virtual IP address or network address of the VRRP group. The format of the address is *ipv4-addr* for IPv4 addresses, or *ipv6-addr* for IPv6 addresses.

Configuration mode

```
interfaces interface {
    vrrp {
        vrrp-group group-id {
            virtual-address addr
         }
   }
}
```

Use this command to set the virtual IP address or network address of a VRRP group on an interface. Every VRRP group must have a virtual address, and all interfaces and vifs in the VRRP group must be configured with the same virtual address.

The virtual address is “shared” by the VRRP group and is dynamically assigned to the master interface in the group. The master links the virtual address to its own MAC address in the network by issuing a gratuitous ARP to the LAN segment. If the master interface fails, the group elects a new master to whom the virtual address is then assigned. The new master notifies the network of the changed MAC address by issuing another gratuitous ARP.

In general, a real interface or vif should not be configured with the virtual address of the VRRP group. In practice, if a real interface is configured with the virtual address, the interface is said to “own” the virtual address. The VRRP standard (RFC 3768) prescribes that a router owning the virtual address should be assigned a priority of 255, which automatically elects the router owning the VIP as the master. If you do assign a virtual address to a real interface, set the priority of the interface to 255.

Use the `set` form of the command to specify the virtual address or network address of a VRRP group on an interface.

Use the `delete` form of the command to remove the virtual address or network address of a VRRP group on an interface. However, note that the virtual address is mandatory in VRRP configuration.

Use the `show` form of the command to display the virtual address or network address of a VRRP group on an interface.

interfaces dataplane <interface> vrrp vrrp-group <group-id> accept <accept>

Enables the master router to accept all packets destined for a virtual IP address.

Syntax:
`set interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`accept`** *accept*

Syntax:
`delete interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`accept`**

Syntax:
`show interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id*

The default accept value is false and is disabled on VRRP.

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

`accept`
Enable the virtual master router to accept packets destined for the address owner's IP address.

The options available are:

- True
- False

Deployments that rely on pinging the address owner's IP address must configure the accept mode to True.

Configuration mode

```
interfaces dataplane interface {
        vrrp {
            vrrp-group group-id {
                accept accept
            }
        }
}
```

Use the `set` form of the command to set the master router to accept all packets destined for the virtual IP address.

Use the `delete` form of the command to remove the master router from accepting all packets destined for the virtual IP address.

Use the `show` form of the command to display whether the master router accepts all packets destined for the virtual IP address.

interfaces dataplane <interface> vrrp vrrp-group <group-id> version <version>

Displays the version of VRRP: 2 for RFC3768 or 3 for RFC5798.

Syntax:
`set interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`version`** *version*

Syntax:

```
delete interfaces dataplane interface vrrp vrrp-group group-id version
```

Syntax:

```
show interfaces dataplane interface vrrp vrrp-group group-id
```

The default version of VRRP is 2.

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

version
The version of VRRP:

- 2 for RFC3768
- 3 for RFC5798

Configuration mode

```
interfaces dataplane interface {
        vrrp {
            vrrp-group group-id {
                version version
            }
        }
}
```

Use the `set` form of the command to set the VRRP version to 2 for RFC3768 and to 3 for RFC5798.

Use the `delete` form of the command to remove the VRRP version.

Use the `show` form of the command to display the VRRP version.

interfaces dataplane <interface> vrrp vrrp-group <group-id> track-interface <interface>

Tracks an interface by monitoring whether the interface is up or down and sets the state of the VRRP instance accordingly.

Syntax:

```
set interfaces dataplane interface vrrp vrrp-group group-id track-interface interface
```

Syntax:

```
delete interfaces dataplane interface vrrp vrrp-group group-id track-interface
```

Syntax:

```
show interfaces dataplane interface vrrp vrrp-group group-id
```

If the interface goes down, the VRRP instance is set to the FAULT state.

interface
The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id
The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

interface
The interface to be tracked.

After the interface is up, the VRRP instance reverts to the previous state.

Configuration mode

```
interfaces dataplane interface {
        vrrp {
            vrrp-group group-id {
                track-interface interface
            }
        }
}
```

Use the `set` form of the command to set the interface tracking and check whether the interface is up or down and set the state of the VRRP instance accordingly.

Use the `delete` form of the command to remove the interface tracking.

Use the `show` form of the command to display whether the interface tracking is enabled.

interfaces dataplane <interface> vrrp vrrp-group <group-id> track-interface <interface> weight value <num>

Sets the weight for a VRRP group, which determines by how much the priority of a group is decremented or incremented when an interface goes down.

Syntax:

`set interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`track-interface`** *interface* **`weight value`** *num*

Syntax:

`delete interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`track-interface`** *interface* **`weight value`**

Syntax:

`show interfaces dataplane` *interface* **`vrrp vrrp-group`** *group-id* **`track-interface`**

interface

The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

`vrrp-group` ***group-id***

The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

`track-interface` ***interface***

The interface to be tracked.

After the interface is up, the VRRP instance reverts to the previous state.

`value` ***num***

The value to increment or decrement the priority. Number can range from 1 through 254. If weight 'value' is provided, 'type' must be provided and vice versa.

Configuration mode

```
interfaces dataplane interfaces {
        vrrp {
            vrrp-group group-id {
                track-interface interface {
                        weight {
                                value num

                        }
                }
            }
        }
}
```

Use the `set` form of the command to change the priority of the VRRP instance.

Use the `delete` form of the command to remove the priority of the VRRP instance.

Use the `show` form of the command to display the priority of the VRRP instance.

interfaces dataplane <interface> vrrp vrrp-group <group-id> track-interface <interface> weight type <type>

Sets the tracking priority of a VRRP interface.

Syntax:

```
set interfaces dataplane interface vrrp vrrp-group group-id track-interface interface weight type type
```

Syntax:

```
delete interfaces dataplane interface vrrp vrrp-group group-id track-interface interface weight type
```

Syntax:

```
show interfaces dataplane interface vrrp vrrp-group group-id track-interface interface weight type
```

interface

The interface type and identifier. For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

`vrrp-group` *group-id*

The VRRP group to which the interface belongs. The identifier ranges from 1 through 255.

`track-interface` *interface*

The interface to be tracked.

After the interface is up, the VRRP instance reverts to the previous state.

`type` *type*

A keyword of either **`increment`** or **`decrement`** to specify how the value is to be applied to the group when the tracked interface goes down.

Configuration mode

```
interfaces dataplane interfaces {
        vrrp {
            vrrp-group group-id {
                track-interface interface {
                        weight {
                                type type

                        }
                }
            }
        }
}
```

Use the `set` form of the command to set the interface tracking priority.

Use the `delete` form of the command to remove the interface tracking priority.

Use the `show` form of the command to display whether the interface tracking priority is enabled.

monitor interfaces vrrp <interface> flow

Displays flow statistics for a VRRP interface.

Syntax:

```
monitor interfaces vrrp interface flow
```

interface

The identifier of a VRRP interface. For the format of a VRRP interface name, refer to the usage guidelines.

Operational mode

Use this command to display flow statistics for a VRRP interface. Type CTRL+C to stop the output.

monitor interfaces vrrp <interface> traffic

Monitors traffic on a VRRP interface.

Syntax:

```
monitor interfaces vrrp interface traffic [ detail [ filter filter-name | unlimited [ filter filter-name ] ] | filter filter-name | save filename | unlimited [ filter filter-name ] ]
```

interface
The identifier of a VRRP interface. For the format of a VRRP interface name, refer to the usage guidelines.

detail
Provides detailed information about the monitored VRRP traffic.

unlimited
Monitors an unlimited amount of traffic.

save *filename*
Saves the monitored traffic to the specified file.

filter *filter-name*
Applies the specified PCAP (packet capture) filter to traffic.

Operational mode

Use this command to monitor traffic on a VRRP interface. Type CTRL+C to stop the output.

The following example shows how to monitor traffic on the dp0p192p1 interface.

```
vyatta@vyatta:~$ monitor interfaces vrrp dp0p192p1 traffic
Capturing on /var/run/dataplane/capture/p192p1
  0.000000      1.1.1.1 -> 224.0.0.18   VRRP 54 Announcement (v2)
  1.000298      1.1.1.1 -> 224.0.0.18   VRRP 54 Announcement (v2)
  2.001358      1.1.1.1 -> 224.0.0.18   VRRP 54 Announcement (v2)
  3.002449      1.1.1.1 -> 224.0.0.18   VRRP 54 Announcement (v2)
  4.002937      1.1.1.1 -> 224.0.0.18   VRRP 54 Announcement (v2)
  5.003872      1.1.1.1 -> 224.0.0.18   VRRP 54 Announcement (v2)
```

monitor vrrp

Generates debug information for the VRRP process.

Syntax:

```
monitor vrrp
```

When used with no option, monitoring is performed in the foreground.

Operational mode

Use this command to monitor VRRP operation.

This is common to all the `monitor` commands.

As of now we do not have the option to save the logs to a file using the `monitor` command.

The following example shows how to monitor VRRP in the foreground and then stop the output by typing CTRL+C. The example omits output from the command.

```
vyatta@vyatta:~$ monitor vrrp
...
CTRL+C
```

reset vrrp master interface <interface> group <group-id>

Forces a transition of the VRRP state to the backup state for an interface.

Syntax:

`reset vrrp master interface` *interface* **`group`** *group-id*

interface

An interface to force to the backup state. Include the type keyword and the interface identifier (for example, dataplane dp0p1p2). For detailed keywords and arguments that can be specified as interfaces, refer to Supported Dataplane Interfaces *(page 77)*.

group-id

A VRRP group within the interface to force to the backup state.

Operational mode

Use this command to force the VRRP master interface to transition to the backup state.

The following example shows how to force the VRRP master interface to the backup state. Notice that before the `run` command is entered, the state is master, and after the command is entered, it is backup. Also, notice the change in “Master router”.

```
vyatta@vyatta:~$ reset vrrp master interface dp0p192p1 group 1
Forcing vyatta-dp0p192p1-1 to BACKUP...
vyatta@vyatta:~$ show vrrp
                              RFC         Addr   Last         Sync
Interface         Group  State   Compliant   Owner  Transition  Group
---------         -----  -----   ---------   -----  ----------  -----
dp0p192p1         1      MASTER  dp0vrrp1    no     5s          <none>
```

restart vrrp

Restarts the VRRP process.

Syntax:

`restart vrrp process`

Operational mode

Use this command to restart the VRRP process.

show interfaces vrrp

Displays information about the configured VRRP interfaces on the vRouter that are RFC-compliant.

Syntax:

`show interfaces vrrp [` **`detail`** | *vrrpx* `[` **`brief`** `] ]`

When used with no option, this command displays summary information about all configured RFC-compliant VRRP interfaces.

`detail`

Displays detailed information about all configured RFC-compliant VRRP interfaces.

vrrpx

The name of an RFC-compliant VRRP interface.

brief

Displays summary information about an RFC-compliant VRRP interface.

Operational mode

Use this command to display information about the configured RFC-compliant VRRP interfaces on the vRouter. For more information about RFC-compliant VRRP interfaces, refer to interfaces <interface> vrrp vrrp-group <group-id> rfc-compatibility *(page 34)*.

The following example shows how to display summary information about the configured RFC-compliant VRRP interfaces.

```
vyatta@vyatta:~$ show interfaces vrrp
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface IP Address S/L Description
--------- ---------- — -----------
dp0vrrp1 3.3.3.3/32 u/u
```

The following example shows how to display detailed information about the configured RFC-compliant VRRP interfaces.

```
vyatta@vyatta:~$ show interfaces vrrp detail
dp0vrrp1@dp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group
 default
link/ether 00:00:5e:00:01:01 brd ff:ff:ff:ff:ff:ff
inet 3.3.3.3/32 scope global dp0vrrp1
valid_lft forever preferred_lft forever
inet6 fe80::200:5eff:fe00:101/64 scope link
valid_lft forever preferred_lft forever

RX: bytes packets errors ignored overrun mcast
0 0 0 0 0 0
TX: bytes packets errors dropped carrier collisions
49098282 909221 0 0 0 0
```

The following example shows how to display detailed information about a specific RFC-compliant VRRP interface.

```
vyatta@vyatta:~$ show interfaces vrrp dp0vrrp1
dp0vrrp1@dp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group
 default
link/ether 00:00:5e:00:01:01 brd ff:ff:ff:ff:ff:ff
inet 3.3.3.3/32 scope global dp0vrrp1
valid_lft forever preferred_lft forever
inet6 fe80::200:5eff:fe00:101/64 scope link
valid_lft forever preferred_lft forever

RX: bytes packets errors ignored overrun mcast
0 0 0 0 0 0
TX: bytes packets errors dropped carrier collisions
39476724 731045 0 0 0 0
```

The following example shows how to display summary information about a specific RFC-compliant VRRP interface.

```
vyatta@vyatta:~$ show interfaces vrrp dp0vrrp1 brief
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface IP Address S/L Description
--------- ---------- — -----------
dp0vrrp1 3.3.3.3/32 u/u
```

show log vrrp

Displays log messages that are generated by the VRRP process.

Syntax:

```
show log vrrp
```

Operational mode

Use this command to display log messages that are generated by the VRRP process.

The following example shows VRRP log messages from a backup router.

```
vyatta@R1:~$ show log vrrp
Feb 14 21:31:27 vyatta Keepalived_vrrp: ------< Global definitions >------
Feb 14 21:31:27 vyatta Keepalived_vrrp:  Router ID = dut4
Feb 14 21:31:27 vyatta Keepalived_vrrp:  Smtp server connection timeout = 30
Feb 14 21:31:27 vyatta Keepalived_vrrp:  Email notification from = root@dut4
Feb 14 21:31:27 vyatta Keepalived_vrrp: ------< VRRP Topology >------
Feb 14 21:31:27 vyatta Keepalived_vrrp:  VRRP Instance = vyatta-dp0p1p2-2
Feb 14 21:31:27 vyatta Keepalived_vrrp:    Want State = BACKUP
Feb 14 21:31:27 vyatta Keepalived_vrrp:    Runing on device = dp0p1p2
Feb 14 21:31:27 vyatta Keepalived_vrrp:    Virtual Router ID = 2
Feb 14 21:31:27 vyatta Keepalived_vrrp:    Priority = 100
Feb 14 21:31:27 vyatta Keepalived_vrrp:    Advert interval = 3sec
Feb 14 21:31:27 vyatta Keepalived_vrrp:    Virtual IP = 1
Feb 14 21:31:27 vyatta Keepalived_vrrp:      172.16.117.100/32 dev dp0p1p2 scope global
Feb 14 21:31:27 vyatta Keepalived_vrrp:    Backup state transition script = /opt/vyatta/sbin/
vyatta-vrrp-state.pl backup dp0p1p2 2 dp0p1p2 null 172.16.117.100
Feb 14 21:31:27 vyatta Keepalived_vrrp:    Master state transition script = /opt/vyatta/sbin/
vyatta-vrrp-state.pl master dp0p1p2 2 dp0p1p2 null 172.16.117.100
Feb 14 21:31:27 vyatta Keepalived_vrrp:    Fault state transition script = /opt/vyatta/sbin/
vyatta-vrrp-state.pl fault dp0p1p2 2 dp0p1p2 null 172.16.117.100
Feb 14 21:31:27 vyatta Keepalived_vrrp: Using LinkWatch kernel netlink reflector...
Feb 14 21:31:27 vyatta Keepalived_vrrp: VRRP_Instance(vyatta-dp0p1p2-2) Entering BACKUP
 STATE
Feb 14 21:31:27 vyatta Keepalived_vrrp: VRRP_Instance(vyatta-dp0p1p2-2) removing protocol
 VIPs.
Feb 14 21:31:27 vyatta Keepalived_vrrp: Opening script file /opt/vyatta/sbin/vyatta-vrrp-
state.pl
```

show vrrp

Displays information about VRRP.

Syntax:

```
show vrrp [ detail | interface interface [ group group-name ] | statistics [ interface interface [ group
group-name ] ] | sync-group [ group group-name ] ]
```

When used with no option, this command displays VRRP state information for all VRRP-configured interfaces.

detail

Displays detailed VRRP information for all VRRP-configured interfaces.

interface *interface*

Displays VRRP information for the specified interface. Use the identifier for the physical Ethernet interface or vif. If the VRRP interface is configured, this command also displays the VRRP interface name and the name of the physical interface or vif. The information that is displayed includes whether the local system is currently running as the owner of the VRRP interface. Note that tab completion that is used after the **interface** keyword provides only parent interface names.

group-name
A VRRP group.

Operational mode

Use this command to display information about VRRP groups, including current VRRP elections and statistics.

> **Note:** When the configuration of the keepalived daemon is modified, the VRRP process is reloaded to read the new configuration and the transition timer is restarted. Therefore, `Last Transition` shows either the last state transition or last time the keepalived configuration was read.

The following example shows how to display information about VRRP.

```
vyatta@R1~$ show vrrp
                                RFC        Addr   Last        Sync
Interface         Group  State  Compliant  Owner  Transition  Group
---------         -----  -----  ---------  -----  ----------  -----
dp0s6                1   MASTER dp0vrrp1   no     2s          <none>
dp0s6                2   MASTER no         no     2s          <none>
```

show vrrp detail

Displays details about all the VRRP groups that are configured at the system level.

Syntax:

```
show vrrp detail
```

`detail`
Specifies the display of detailed VRRP information.

Operational mode

Use this command to display information about all the VRRP groups configured at the system level.

The system automatically generates a name for the VRRP interface by appending vrrp*n* (where *n* is an arbitrary number starting from 1) to the physical interface prefix, for example, dp0vrrp1.

The interface name is based on the highest existing interface; for a scenario in which you configure three groups (dp0vrrp1, dp0vrrp2, and dp0vrrp3), then delete dp0vrrp2 and create another group, the interface name for the new group is dp0vrrp4.

> **Note:** When the configuration of the keepalived daemon is modified, the VRRP process is reloaded to read the new configuration and the transition timer is restarted. Therefore, `Last transition` shows either the last state transition or last time the keepalived configuration was read.

The following example shows how to display information about all the VRRP groups configured at the system level.

```
vyatta@vyatta:~$ show vrrp detail
------------------------------------------------
Interface: dp0s6
--------------
  Group: 1
  ----------
  State:                        MASTER
  Last transition:              15m50s

  Version:                      3
  RFC Compliant
  Virtual MAC interface:        dp0vrrp1
  Address Owner:                yes
```

```
  Source Address:               172.16.0.11
  Configured Priority           200
  Effective Priority            200
  Advertisement interval:       1000 milli-sec
  Preempt:                      enabled
  Accept:                       enabled

  VIP count:                    1
    172.16.0.11/32
```

show vrrp interface

Displays information about the groups configured on the VRRP interface or all VRRP interfaces.

Syntax:

```
show vrrp interface [ vrrpx ]
```

When used with no option, this command displays information about all VRRP interfaces.

vrrpx

A VRRP interface. For the format of the VRRP interface name, refer to the usage guidelines.

Operational mode

Use this command to display information about one VRRP interface or all VRRP interfaces.

The system automatically generates a name for the VRRP interface by appending vrrp*n* (where *n* is an arbitrary number starting from 1) to the physical interface prefix, for example, dp0vrrp1.

The interface name is based on the highest existing interface; for a scenario in which you configure three groups (dp0vrrp1, dp0vrrp2, and dp0vrrp3), then delete dp0vrrp2 and create another group, the interface name for the new group is dp0vrrp4.

Note:

When the configuration of the keepalived daemon is modified, the VRRP process is reloaded to read the new configuration and the transition timer is restarted. Therefore, `Last transition` shows either the last state transition or last time the keepalived configuration was read.

The following example shows how to display summary information of the group configured on the VRRP interfaces.

```
vyatta@vyatta:~$ show vrrp interface dp0p5p1
------------------------------------------------
Interface: dp0p5p1
--------------
Group: 100
----------
State:                        BACKUP
Last transition:              33m50s

Master router:                2ffe:20::15
Master priority:              255
Version:                      3
RFC Compliant
Virtual MAC interface:        dp0vrrp2
Address Owner:                no

Source Address:               2ffe:20::20
Configured Priority           100
Effective Priority            100
Advertisement interval:       1000 milli-sec
Preempt:                      enabled
Accept                        disabled
```

```
Sync-group:                   ABC
Tracked interfaces count:                                    2
    dpop3p1    state    UP    weight    +20
    dp0p4p1    state    UP    weight    -10
VIP count:                    1
    2ffe:20::15/128
```

update vrrp garp interface <interface> vrrp-group <group>

Sends a gratuitous ARP (GARP) packet to update address tables in the event of a problem with MAC address tables in the network.

Syntax:

`update vrrp garp` **`interface`** *interface* **`vrrp-group`** *group*

NA

interface

A VRRP interface. For the format of the VRRP interface name, refer to the usage guidelines.

group

A VRRP group.

Operational mode

Use this command to update address tables in the event of a problem with MAC address tables in the network.

This command to request a GARP packet is sent for the specified group through the specified interface. The packet will update both Layer 2 switching tables and the ARP caches of all relevant hosts.

Configuration Synchronization

Configuration synchronization configuration

This section describes configuration synchronization and provides examples that show how to set it up and use it. This section presents the following topics:

- Configuration synchronization overview
- Master and standby systems
- Out-of-sync systems
- Configuration synchronization examples

Configuration synchronization overview

Many high-availability deployments involve the use of hot standby systems. In these scenarios, one priority reduces configuration differences between the master and standby systems. The ability to synchronize defined portions of configuration minimizes configuration differences at failover while posing the least-possible burden on the master system.

By default, all configuration is excluded from synchronization and you must explicitly include a configuration node for it to be synchronized. Only a single standby system is currently supported.

Every time the master system starts or a `commit`, `load`, or `merge` command is run on the master system, the standby system is synchronously updated and its configuration is saved. The system tries to make all parameter values on the standby system within the **`sync-map`** identical to the master. Partial commits, in the event of a commit failure, synchronize only the sections that were successfully committed on the master system.

Note: Be sure to save the configuration on the master system and be mindful that this operation does not trigger the standby system to save its configuration.

Note: Both the master and standby systems must be running AT&T Vyatta vRouter, properly configured for entitlement, and able to access the AT&T Vyatta vRouter entitlement server.

Master and standby systems

Configuration synchronization allows you to designate a master AT&T Vyatta vRouter that can synchronize defined portions of the configuration with a remote standby system. The master system dispatches locally generated configuration commands (`set`, `delete`, `commit`, and so on) to the remote standby system; the commands are dispatched from the master system when the configuration change is committed.

The master system uses the REST API of theAT&T Vyatta vRouter to propagate commands to the remote systems; for this reason, the remote system must have HTTPS enabled to participate.

When setting up the master and standby systems on the network, keep in mind that the AT&T Vyatta vRouter does not prevent you from setting up mis-synchronization scenarios that can result in network problems. Take care to avoid scenarios that cause network problems. For example:

- Configuration information that is unique to a system should not be synchronized between systems because synchronizing this information can cause problems on the network. Examples of configuration that should be excluded from synchronization include interface, IP addresses, and VRRP priorities.
- Avoid setting up two master systems to synchronize each other.
- Avoid setting up two master systems so that both synchronize the same configuration on a third system.

Note that some configuration items are not allowed to be synchronized because synchronizing them would damage your system configuration. The two items that cannot be synchronized are the `system config-sync` configuration node itself and the `service https` configuration node. Disallowed configuration items do not appear in the configuration tree when excluded items are listed.

Note also that only one **sync-map** can be defined for the standby system.

Finally, avoid direct modification of the configuration on the standby system in areas contained in the **sync-map** as this may result in conflicts and commit failures during the synchronization process.

Out-of-sync systems

The master system logs a warning if configuration elements become out of synchronization (out of sync), but does not attempt to correct configuration. You can view the warnings in the log file by issuing the `show config-sync status` operational command.

If configuration does become out of sync, you can reset the configuration on the standby system by issuing the `update config-sync` operational command. When this command has successfully completed, the systems are synchronized again.

If the systems become out of sync because of a configuration conflict, the conflict must be addressed before entering the `update config-sync` command.

The `update config-sync` command is also useful if the secondary is booted with an out-of-sync configuration. Entering this command on the master system synchronizes the configurations without having to commit a configuration change on the master.

To set up configuration synchronization on the AT&T Vyatta vRouter, use the following work flow.

1. Identify the master system.
2. Identify the remote standby system.
3. Identify the portion of the configuration tree to be synchronized ("include").

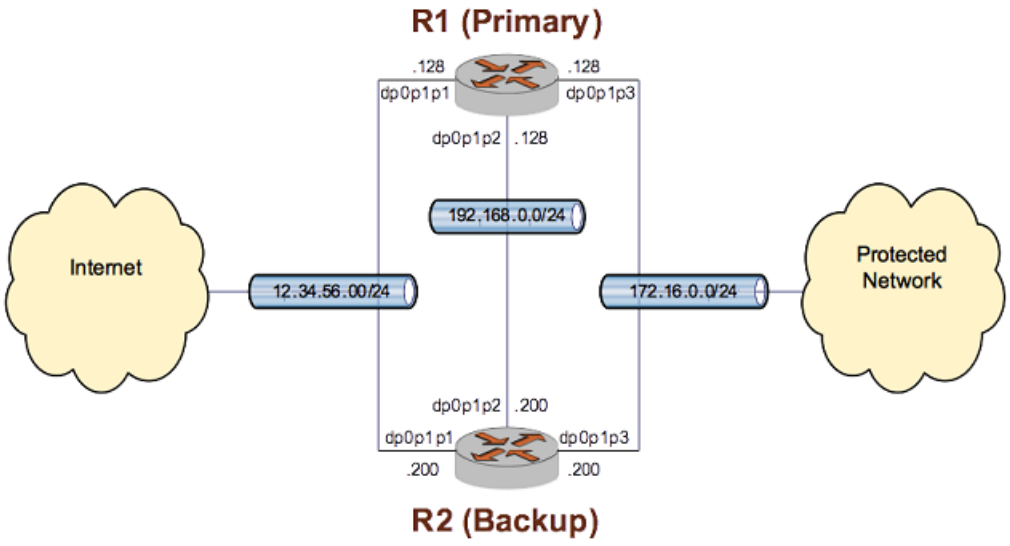
Configuration synchronization examples

This section contains examples of configuration synchronization on the AT&T Vyatta vRouter.

Basic configuration synchronization

This section sets up configuration synchronization with the scenario shown in the following figure.

Figure 4: Configuration synchronization



In this example:

- The master system is R1.
- The standby system is R2. The remote system is to be accessed by using the default username and password: vyatta.

- Firewall configuration is to be synchronized.

To configure R1 for configuration synchronization in this way, perform the following steps in configuration mode.

Table 10: Configuring R1 for configuration synchronization

Step	Command
Create the configuration synchronization service and a synchronization map (TEST) with a rule (2) within the synchronization map. Specify a node (firewall) as a node on which to take action.	`vyatta@R1# set system config-sync sync-map TEST rule 2 location security firewall`
Specify the synchronization action to take on the firewall configuration node during configuration synchronization. The firewall node is to be included in configuration synchronization.	`vyatta@R1# set system config-sync sync-map TEST rule 2 action include`
Define the remote system with which to synchronize and set the username (vyatta) to be used for connecting to the remote system.	`vyatta@R1# set system config-sync remote-router 192.168.1.22 username vyatta`
Set the password (vyatta) to be used for connecting to the remote system.	`vyatta@R1# set system config-sync remote-router 192.168.1.22 password vyatta`
Set TEST as the synchronization map to be used for configuration synchronization with the remote system.	`vyatta@R1# set system config-sync remote-router 192.168.1.22 sync-map TEST`
Commit the configuration.	`vyatta@R1# commit`

Display the configuration.	`vyatta@R1# show system config-sync`
	`remote-router 192.168.1.22 {`
	`password vyatta`
	`sync-map TEST`
	`username vyatta`
	`}`
	`sync-map TEST {`
	`rule 2 {`
	`action include`
	`location security firewall`
	`}`
	`}`

Configuration Synchronization Commands

system config-sync remote-router <addr>

Specifies the address of the standby system.

Syntax:
`set system config-sync remote-router` *addr*

Syntax:
`delete system config-sync remote-router` *addr*

Syntax:
`show system config-sync remote-router` *addr*

addr
Multi-node. The IPv4 address of the standby system.

Configuration mode

```
system {
    config-sync {
        remote-router addr {}
    }
}
```

Use this command to specify the address of the system whose configuration is to be automatically synchronized with a subset of that defined on the master system by the **`sync-map`**.

Use the `set` form of the command to specify the address of the standby system.

Use the `delete` form of the command to remove the address of the standby system.

Use the `show` form of the command to view the configuration syncronization configuration of the standby system.

system config-sync remote-router <addr> password <password>

Specifies the password to be used to access the standby system.

Syntax:
`set system config-sync remote-router` *addr* **`password`** *password*

Syntax:
`delete system config-sync remote-router` *addr* **`password`**

Syntax:
`show system config-sync remote-router` *addr* **`password`**

addr
Multi-node. The IPv4 address of the standby system.

password

The password to use (with a username) to access the standby system.

Configuration mode

```
system {
    config-sync {
        remote-router addr {
            password password
        }
    }
}
```

Use this command to specify the password to use to access the standby system for automated configuration synchronization.

Use the `set` form of the command to specify the password.

Use the `delete` form of the command to remove the password.

Use the `show` form of the command to display the password.

system config-sync remote-router <addr> sync-map <sync-map-name>

Specifies the synchronization map that is used to define the standby system configuration.

Syntax:
`set system config-sync remote-router` *addr* **`sync-map`** *sync-map-name*

Syntax:
`delete system config-sync remote-router` *addr* **`sync-map`**

Syntax:
`show system config-sync remote-router` *addr* **`sync-map`**

addr
Multi-node. The IPv4 address of the standby system.

sync-map-name
The name of a synchronization map.

Configuration mode

```
system {
    config-sync {
        remote-router addr {
            sync-map sync-map-name
        }
    }
}
```

Use this command to specify a synchronization map that is used to define the subset of the local configuration to be synchronized with the standby system. Only one **`sync-map`** can be defined per **`remote-router`**.

Use the `set` form of the command to specify a synchronization map.

Use the `delete` form of the command to remove a synchronization map.

Use the `show` form of the command to display a synchronization map.

system config-sync remote-router <addr> username <username>

Specifies the username to be used to access the standby system.

Syntax:
`set system config-sync remote-router` *addr* **`username`** *username*

Syntax:
`delete system config-sync remote-router` *addr* **`username`**

Syntax:
`show system config-sync remote-router` *addr* **`username`**

addr
Multi-node. The IPv4 address of the standby system.
username
The username to use (with a password) to access the standby system.

Configuration mode

```
system {
    config-sync {
        remote-router addr {
            username username
        }
    }
}
```

Use this command to specify the username to use to access the standby system for automated configuration synchronization. The user must have administrator (admin) rights on the standby system or synchronization fails. Synchronization also fails if an invalid username or password is specified.

Use the `set` form of the command to specify the username.

Use the `delete` form of the command to remove the username.

Use the `show` form of the command to display the username

system config-sync sync-map <sync-map-name>

Specifies the portions of the configuration to be synchronized with the standby system.

Syntax:
`set system config-sync sync-map` *sync-map-name*

Syntax:
`delete system config-sync sync-map` *sync-map-name*

Syntax:
`show system config-sync sync-map` *sync-map-name*

sync-map-name
Multi-node. The name of a synchronization map.

Configuration mode

```
system {
    config-sync {
```

```
        sync-map sync-map-name {}
    }
}
```

Use this command to specify a synchronization map that defines the portions of the configuration to be synchronized with the standby system. Multiple configuration nodes can be specified to identify multiple synchronization maps. Only one synchronization map can be assigned to the standby system at one time.

Use the `set` form of the command to specify a synchronization map.

Use the `delete` form of the command to remove a synchronization map.

Use the `show` form of the command to display a synchronization map.

system config-sync sync-map <sync-map-name> rule <rule-num>

Specifies a synchronization map rule.

Syntax:

`set system config-sync sync-map` *sync-map-name* **`rule`** *rule-num*

Syntax:

`delete system config-sync sync-map` *sync-map-name* **`rule`** *rule-num*

Syntax:

`show system config-sync sync-map` *sync-map-name* **`rule`** *rule-num*

sync-map-name
Multi-node. The name of a synchronization map.

rule-num
Multi-node. The numeric identifier for a rule. The identifier ranges from 1 through 1024.

Configuration mode

```
system {
    config-sync {
        sync-map sync-map-name {
            rule rule-num {}
        }
    }
}
```

Use this command to specify a synchronization map rule that defines the portions of the configuration to be synchronized with the standby system. Multiple configuration nodes can be specified to identify multiple synchronization map rules. The rule number specifies the order of evaluation with respect to other rules. The first match of a configuration element stops further comparisons.

Use the `set` form of the command to specify a synchronization map rule.

Use the `delete` form of the command to remove a synchronization map rule.

Use the `show` form of the command to display a synchronization map rule.

system config-sync sync-map <sync-map-name> rule <rule-num> action <action>

Specifies an action for a rule.

Syntax:

`set system config-sync sync-map` *sync-map-name* **`rule`** *rule-num* **`action`** *action*

Syntax:
`delete system config-sync sync-map` *sync-map-name* **`rule`** *rule-num* **`action`**

Syntax:
`show system config-sync sync-map` *sync-map-name* **`rule`** *rule-num* **`action`**

sync-map-name
Multi-node. The name of a synchronization map.
rule-num
Multi-node. The numeric identifier for a rule. The identifier ranges from 1 through 1024.
`action` ***action***
Specifies the action to be taken:

- **`include`**: Include the configuration node specified by the **`location`** parameter in the configuration synchronization.
- **`exclude`**: Exclude the configuration node specified by the **`location`** parameter from the configuration synchronization.

Configuration mode

```
system {
    config-sync {
        sync-map sync-map-name {
            rule rule-num {
                action action
           }
        }
    }
}
```

Use this command to specify an action for a rule.

Use the `set` form of the command to specify the action for a rule.

Use the `delete` form of the command to remove the action for a rule.

Use the `show` form of the command to display the action for a rule.

system config-sync sync-map <sync-map-name> rule <rule-num> location <config-path>

Specifies a configuration node to be acted on by a rule.

Syntax:
`set system config-sync sync-map` *sync-map-name* **`rule`** *rule-num* **`location`** *config-path*

Syntax:
`delete system config-sync sync-map` *sync-map-name* **`rule`** *rule-num* **`location`**

Syntax:
`show system config-sync sync-map` *sync-map-name* **`rule`** *rule-num* **`location`**

sync-map-name
Multi-node. The name of a synchronization map.
rule-num
Multi-node. The numeric identifier for a rule. The identifier ranges from 1 through 1024.
config-path
The path to the configuration node. If the configuration node contains more than one character string, the strings should be separated by spaces and the entire string should be enclosed in double quotation marks (for example, “system login user dave”).

Configuration mode

```
system {
    config-sync {
        sync-map sync-map-name {
            rule rule-num {
                location config-path
            }
        }
    }
}
```

Use this command to specify a configuration node to be acted on by a rule.

Use the `set` form of the command to specify the configuration node to be acted on by a rule.

Use the `delete` form of the command to remove the location configuration.

Use the `show` form of the command to view the location configuration.

system config-sync timeout

Specifies the timeout value to synchronize system configuration with remote routers.

Syntax:
`set system config-sync timeout` { *seconds* }

Syntax:
`delete system config-sync timeout` [*seconds*]

Syntax:
`show system config-sync timeout` [*seconds*]

The default synchronization timeout is 900 seconds.

seconds
The time required to synchronize system configuration with remote routers in seconds. The value ranges from 0 to 4294967295.

Configuration mode

```
system {
     config-sync {
          timeout seconds
     }
}
```

Use the `set` form of this command to set the timeout value to synchronize system configuration with remote routers.

Use the `delete` form of this command to remove the synchronization timeout.

Use the `show` form of this command to display the timeout value to synchronize system configuration with remote routers.

show config-sync difference

Displays configuration differences between the master system and the standby system.

Syntax:
`show config-sync difference` [*addr* [**`detail`**]]

addr
The IP address of the standby system.

detail
Provides detailed information.

Operational mode

Use this command to compare the configuration that is identified in the synchronization map on the standby system with the synchronization map of the master system to find any differences.

The following example shows how to display differences in configuration synchronization.

```
vyatta@R1> show config-sync difference

192.168.0.200 configuration is in sync
```

The following example shows how to display detailed differences in configuration synchronization.

```
vyatta@R1> show config-sync difference 192.168.74.200 detail

Configuration only on master (compared to 192.168.74.200):
  zone-policy zone lan default-action drop
  zone-policy zone lan from public firewall name public_to_lan
  zone-policy zone lan interface dp0p1p1
  zone-policy zone public default-action drop
  zone-policy zone public from lan firewall name lan_to_public
  zone-policy zone public interface dp0p1p4
192.168.74.200 configuration is out of sync
```

show config-sync status

Provides details of the last commit to the standby system.

Syntax:
`show config-sync status [ addr [ detail ] ]`

addr
The IP address of the standby system.

detail
Provides detailed information.

Operational mode

Use this command to display the commit status of the standby system. The status includes any errors and where they occurred in the synchronization process, the reason for the errors, and the software versions of the local and remote systems.

The following example shows how to display the status of configuration synchronization.

```
vyatta@R1> show config-sync status
remote-router:  192.168.0.200
  version:              999.larkspurse.08101304
  sync-map:             MAP1
  last sync:            good
  last sync time:       Thu Aug 12 18:18:14 2010
  in-sync?:             yes
  access-status:        connected
```

The following example shows how to display the detailed status of configuration synchronization.

```
vyatta@R1> show config-sync status 192.168.0.200 detail
remote-router:  192.168.0.200
  version:              999.larkspurse.08101304
  sync-map:             MAP1
  last sync:            good
  last sync time:       Thu Aug 12 18:18:14 2010
  in-sync?:             yes
  access-status:        connected

remote configuration:
   cluster {
     dead-inteval 2000
     group MAIN {
        auto-fallback true
        monitor 172.16.0.131
        monitor 12.34.56.133
        primary R1
        secondary R2
        service 12.34.56.100/24/dp0p1p1
        service 172.16.0.100/24/dp0p1p3
     }
     interface dp0p1p2
     keepalive-interval 500
     monitor-dead-interval 2000
     pre-shared-secret testing
  }
  firewall {
     name internet_to_protected {
        default_action drop
:
```

update config-sync

Synchronizes configuration on the standby system.

Syntax:

`update config-sync [ addr ]`

addr

The IP address of the standby system.

Operational mode

Use this command to synchronize the configuration of the standby system with the master system based on the configuration nodes specified in the **sync-map**.

Connection Synchronization

Connection synchronization overview

Connection synchronization is a feature that is used by the system to support high availability between two instances of AT&T Vyatta vRouter running VRRP.

To support high availability, the firewall states must be synchronized between the master and backup routers. The connection synchronization feature is used by the system to perform this synchronization. When a backup router with VRRP becomes the master router, this feature initializes the firewall states in the new master.

Note: State synchronization for NAT and ALG is not supported.

Connection synchronization helps keep existing stateful connections going through the master and backup routers alive even after failover.

Note: When you configure connection synchronization on an AT&T Vyatta vRouter, the maximum number of session entries that you can configure by using the `system session table-size` command is 200000 when the system memory is 4G, or 100000 entries when the system memory is 2G.

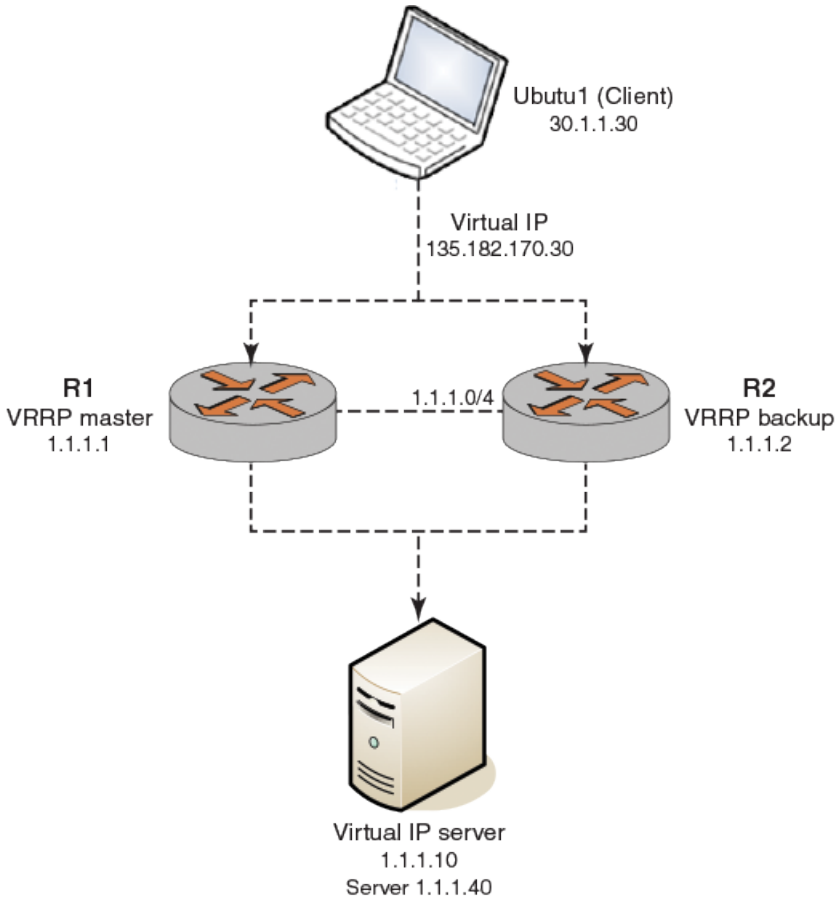
Note: Connection synchronization works only for active-passive configurations.

Configuring connection synchronization

This example shows how to configure connection synchronization between two vRouters that are configured by enabling the firewall and VRRP. The R1 vRouter is configured as the VRRP master router and the R2 vRouter is configured as the VRRP backup router.

When you complete the example, the interfaces are configured as shown in the following figure.

Figure 5: Configuring connection synchronization



To configure the connection synchronization, you must configure the failover mechanism, interface, and remote peer for each router.

Perform the following steps on the R1 vRouter.

Table 11: Configuring connection synchronization on the R1 vRouter

Step	Command
Assign an IP address directly to the dp0p192p1 untagged Ethernet interface.	`vyatta@R1# set interfaces dataplane dp0p192p1 address 192.168.1.10/24` `vyatta@R1# set interfaces dataplane dp0p192p1 description Uplink`
Assign an IP address directly to the dp0p224p1 untagged Ethernet interface.	`vyatta@R1# set interfaces dataplane dp0p224p1 address 135.182.170.2/24` `vyatta@R1# set interfaces dataplane dp0p224p1 description Downlink`

Step	Command
Configure the firewall for the dp0p224p1 interface.	`vyatta@R1# set interfaces dataplane dp0p224p1 firewall in Inbound` `vyatta@R1# set interfaces dataplane dp0p224p1 firewall outbound`
Configure the firewall for the dp0p192p1 interface.	`vyatta@R1# set interfaces dataplane dp0p192p1 firewall in DNS-HTTP`

Configure VRRP for the dp0p224p1 interface.	`vyatta@R1# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 priority 150`
	`vyatta@R1# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 rfc-compatibility`
	`vyatta@R1# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 sync-group Inside`
	`vyatta@R1# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 virtual-address 135.182.170.30`
	`vyatta@R1# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 notify bgp`

Assign an IP address to the dp0p256p1 interface.	`vyatta@R1# set interfaces dataplane dp0p256p1 address 1.1.1.1/24` `vyatta@R1# set interfaces dataplane dp0p256p1 description CONNSYNC`

Step	Command
Configure the firewall.	vyatta@R1# set security firewall name inbound rule 10 action accept vyatta@R1# set security firewall name inbound rule 10 log vyatta@R1# set security firewall name inbound rule 10 protocol tcp vyatta@R1# set security firewall name inbound rule 10 state enable vyatta@R1# set security firewall name DNS-HTTP rule 10 action accept vyatta@R1# set security firewall name DNS-HTTP rule 10 log vyatta@R1# set security firewall name DNS-HTTP rule 10 protocol udp vyatta@R1# set security firewall name DNS-HTTP rule 10 state enable vyatta@R1# set security firewall session-log tcp established vyatta@R1# set security firewall session-log tcp fin-sent vyatta@R1# set security firewall session-log udp closed vyatta@R1# set security firewall session-log udp established vyatta@R1# set security firewall session-log udp new vyatta@R1# set security firewall session-log udp timeout vyatta@R1# set security firewall session-log tcp fin-wait
Configure the connection synchronization failover.	vyatta@R1# set service connsync failover-mechanism vrrp sync-group Inside vyatta@R1# set service connsync interface dp0p256p1
Configure an IP address for the R2 vRouter as the remote peer.	vyatta@R1# set service connsync remote-peer 1.1.1.2
Configure SSH.	vyatta@R1# set service ssh

Step	Command

Verify the connection synchronization configuration.	`vyatta@R1:~$ show service`
	`service {`
	`  connsync {`
	`  failover-mechanism {`
	`  vrrp {`
	`  sync-group Inside`
	`  }`
	`  }`
	`  interface dp0p256p1`
	`  remote-peer 1.1.1.2`
	`  }  0`

Verify the configured interfaces.	`vyatta@R1:~$ show interfaces`
	`interfaces {`
	`  dataplane dp0p160p1 {`
	`  address 10.18.191.11/24`
	`  }`
	`  dataplane dp0p192p1 {`
	`  address 192.168.1.10/24`
	`  description UPlink`
	`  firewall {`
	`  in DNS-HTTP`
	`  }`
	`  }`
	`  dataplane dp0p224p1 {`
	`  address 135.182.170.2/24`
	`  description Downlink`
	`  firewall {`
	`  in Inbound`
	`  out Outbound`
	`  }`
	`  vrrp {`
	`  vrrp-group 99 {`
	`  notify {`
	`  bgp`
	`  }`
	`  priority 150`
	`  rfc-compatibility`
	`  sync-group Inside`
	`  virtual-address 135.182.170.30`
	`  }`
	`  }`
	`  }`
	`  dataplane dp0p256p1 {`
	`  address 1.1.1.1/30`
	`  description CONNSYNC`
	`  }`
	`  loopback lo {`
	`  address 7.7.7.1/32`
	`  }`
	`}`

Step	Command

Verify entries in the session table.	`vyatta@R1:~$ run show session-table`
	`TCP state codes: SS - SYN SENT, SR - SYN RECEIVED, ES - ESTABLISHED,`
	`FW - FIN WAIT, CW - CLOSE WAIT, LA - LAST ACK,`
	`TW - TIME WAIT, CL - CLOSE, LI - LISTEN`
	`CONN ID Source Destination Protocol TIMEOUT Intf Parent`
	`3121279 135.182.170.204:1052 155.1.34.13:53 udp [17] new 18 dp0p192p1 0`
	`3121307 135.182.170.218:1066 155.1.34.13:53 udp [17] new 18 dp0p192p1 0`
	`3121433 135.182.170.206:1129 155.1.34.13:53 udp [17] new 22 dp0p192p1 0`
	`3121438 135.182.170.134:1132 155.1.34.13:53 udp [17] new 22 dp0p192p1 0`
	`3121447 135.182.170.213:1136 155.1.34.13:53 udp [17] new 22 dp0p192p1 0`
	`3121509 135.182.170.244:1167 155.1.34.13:53 udp [17] new 22 dp0p192p1 0`
	`3121539 135.182.170.184:1182 155.1.34.13:53 udp [17] new 22 dp0p192p1 0`
	`3121544 135.182.170.112:1185 155.1.34.13:53 udp [17] new 22 dp0p192p1 0`
	`3121565 135.182.170.197:1195 155.1.34.13:53 udp [17] new 22 dp0p192p1 0`

Verify entries in the internal cache table.	`vyatta@R1:~$ run show connsync internal-cache`
	`Number of entries: 16385`
	`Source Destination Protocol`
	`135.182.170.130:8886 155.1.34.62:53 udp [17]`
	`135.182.170.142:10198 155.1.34.50:53 udp [17]`
	`135.182.170.109:23089 155.1.34.85:53 udp [17]`
	`135.182.170.179:30361 155.1.34.75:53 udp [17]`
	`135.182.170.225:23826 155.1.34.16:53 udp [17]`
	`135.182.170.177:12593 155.1.34.95:53 udp [17]`
	`135.182.170.143:24313 155.1.34.93:53 udp [17]`
	`135.182.170.180:36112 155.1.34.110:53 udp [17]`
	`135.182.170.171:26976 155.1.34.79:53 udp [17]`

Step	Command

Verify connection synchronization statistics.	`vyatta@R1:~$ run show connsync statistics`
	`local:`
	`msg: tx/rx 1/3997592 tx_err/rx_err 0/0`
	`tx_end/rx_end 1/2`
	`cache: size/max: 81401/1048576 insert:`
	`1618119 update: 2172451`
	`delete: 33786 expirted: 1124967`
	`evicted: 0`
	`err: update/delete/stale 0/0/0`
	`remote:`
	`msg: tx/rx 1/7075953 tx_err/rx_err 0/0`
	`tx_end/rx_end 1/0`
	`cache: size/max: 0/1048576 insert:`
	`3059007 update: 3718759`
	`delete: 99227 expirted: 2977595`
	`evicted: 0`
	`err: update/delete/stale 0/0/0`

Perform the following steps on the R2 vRouter.

Table 12: Configuring connection synchronization on the R2 vRouter

Step	Command
Assign an IP address directly to the dp0p192p1 untagged Ethernet interface.	`vyatta@R2# set interfaces dataplane dp0p192p1 address 192.168.1.11/24` `vyatta@R2# set interfaces dataplane dp0p192p1 description Uplink`
Assign an IP address directly to the dp0p224p1 untagged Ethernet interface.	`vyatta@R2# set interfaces dataplane dp0p224p1 address 135.182.170.2/24` `vyatta@R2# set interfaces dataplane dp0p224p1 description Downlink`
Configure the firewall for the dp0p224p1 interface.	`vyatta@R2# set interfaces dataplane dp0p224p1 firewall in Inbound` `vyatta@R2# set interfaces dataplane dp0p224p1 firewall out Outbound`
Configure the firewall for the dp0p192p1 interface.	`vyatta@R2# set interfaces dataplane dp0p192p1 firewall in DNS-HTTP`

Step	Command

Configure VRRP for the dp0p224p1 interface.	`vyatta@R2# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 priority 150`
	`vyatta@R2# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 rfc-compatibility`
	`vyatta@R2# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 sync-group Inside`
	`vyatta@R2# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 virtual-address 135.182.170.30`
	`vyatta@R2# set interfaces dataplane dp0p224p1 vrrp vrrp-group 99 notify bgp`

Assign an IP address to the dp0p256p1 interface.	`vyatta@R2# set interfaces dataplane dp0p256p1 address 1.1.1.1/24` `vyatta@R2# set interfaces dataplane dp0p256p1 description CONNSYNC`

Step	Command

Configure the firewall.	`vyatta@R2#  set security firewall name inbound rule 10 action accept`
	`vyatta@R2# set security firewall name inbound rule 10 log`
	`vyatta@R2# set security firewall name inbound rule 10 protocol tcp`
	`vyatta@R2# set security firewall name inbound rule 10 state enable`
	`vyatta@R2# set security firewall name DNS-HTTP rule 10 action accept`
	`vyatta@R1# set security firewall name DNS-HTTP rule 10 log`
	`vyatta@R2# set security firewall name DNS-HTTP rule 10 protocol udp`
	`vyatta@R2# set security firewall name DNS-HTTP rule 10 state enable`
	`vyatta@R2# set security firewall session-log tcp established`
	`vyatta@R2# set security firewall session-log tcp fin-sent`
	`vyatta@R2# set security firewall session-log udp closed`
	`vyatta@R2# set security firewall session-log udp established`
	`vyatta@R2# set security firewall session-log udp new`
	`vyatta@R2# set security firewall session-log udp timeout`
	`vyatta@R2# set security firewall session-log tcp fin-wait`

Configure the connection synchronization failover.	`vyatta@R2# set service connsync failover-mechanism vrrp sync-group Inside` `vyatta@R2# set service connsync interface dp0p256p1`
Configure an IP address for the R2 vRouter as the remote peer.	`vyatta@R2# set service connsync remote-peer 1.1.1.1`

Step	Command
Configure SSH.	`vyatta@R2# set service ssh`

Connection Synchronization Commands

service connsync failover-mechanism vrrp sync-group <sync-group>

Enables or disables the failover mechanism for connection synchronization.

Syntax:
`set service connsync failover-mechanism vrrp sync-group` *sync-group*

Syntax:
`delete service connsync failover-mechanism vrrp sync-group [` *sync-group* `]`

Syntax:
`show service connsync failover-mechanism vrrp sync-group`

sync-group
The name of a VRRP synchronization group.

Configuration mode

```
service {
    connsync {
        failover-mechanism {
            vrrp {
                sync-group sync-group
            }
        }
    }
}
```

Use the `set` form of this command to enable or disable the failover mechanism for connection synchronization.

Use the `delete` form of this command to delete the current configuration of the failover mechanism for connection synchronization.

Use the `show` form of this command to display the state of the failover mechanism for connection synchronization.

service connsync interface <interface>

Specifies the interface to use for connection synchronizing flow entries.

Syntax:
`set service connsync interface` *interface*

Syntax:
`delete service connsync interface [` *interface* `]`

Syntax:
`show service connsync interface`

interface

The name of an interface.

Configuration mode

```
service {
    connsync {
        interface interface {
        }
    }
}
```

Use the `set` form of this command to specify the interface to use for synchronizing flow entries.

Use the `delete` form of this command to remove the interface that is used for synchronizing flow entries.

Use the `show` form of this command to display the interface that is used for synchronizing flow entries.

service connsync remote-peer <address>

Specifies the remote peer for connection synchronization.

Syntax:
`set service connsync remote-peer` *address*

Syntax:
`delete service connsync interface [` *address* `]`

Syntax:
`show service connsync interface`

address
The IP address for a remote peer.

Configuration mode

```
service {
    connsync {
        remote-peer address {
        }
    }
}
```

Use the `set` form of this command to specify the address of a connection synchronization remote peer.

Use the `delete` form of this command to delete the address of the remote peer for connection synchronization.

Use the `show` form of this command to display the address of the remote peer for connection synchronization.

restart connsync

Restarts connection synchronization.

Syntax:
`restart connsync`

Operational mode.

Connection synchronization must be configured on the system to use this command.

Use this command to restart connection synchronization.

show connsync external-cache

Displays external cache entries for connection synchronization.

Syntax:

```
show connsync external-cache
```

Operational mode

Connection synchronization must be configured on the system to use this command.

Use this command to display external cache entries for connection synchronization.

The following example shows how to display cache information for connection synchronization.

```
vyatta@vyatta:~$ show connsync external-cache
Numner of entries: 16385
Source                          Destination                   Protocol
192.168.17.30:58337             192.168.100.40:5007            udp [17]
192.168.17.30:37468             192.168.100.40:5001            udp [17]
192.168.17.30:38719             192.168.100.40:5001            udp [17]
192.168.17.30:51328             192.168.100.40:5009            udp [17]
192.168.17.30:47237             192.168.100.40:5001            udp [17]
192.168.17.30:40794             192.168.100.40:5002            udp [17]
192.168.17.30:46375             192.168.100.40:5001            udp [17]
192.168.17.30:57484             192.168.100.40:5000            udp [17]
192.168.17.30:59403             192.168.100.40:5001            udp [17]
192.168.17.30:46396             192.168.100.40:5001            udp [17]
192.168.17.30:41066             192.168.100.40:5001            udp [17]
192.168.17.30:36309             192.168.100.40:5009            udp [17]
192.168.17.30:47408             192.168.100.40:5006            udp [17]
192.168.17.30:32963             192.168.100.40:5003            udp [17]
192.168.17.30:33957             192.168.100.40:5002            udp [17]
192.168.17.30:40226             192.168.100.40:5005            udp [17]
192.168.17.30:38901             192.168.100.40:5005            udp [17]
192.168.17.30:53885             192.168.100.40:5004            udp [17]
192.168.17.30:46619             192.168.100.40:5002            udp [17]
192.168.17.30:39099             192.168.100.40:5007            udp [17]
192.168.17.30:48855             192.168.100.40:5008            udp [17]
192.168.17.30:44828             192.168.100.40:5009            udp [17]
192.168.17.30:47916             192.168.100.40:5009            udp [17]
```

show connsync internal-cache

Displays internal cache entries for connection synchronization.

Syntax:

```
show connsync internal-cache
```

Operational mode

Connection synchronization must be configured on the system to use this command.

Use this command to display internal cache entries for connection synchronization.

The following example shows how to display cache information for connection synchronization.

```
vyatta@vyatta:~$ show connsync internal-cache
Numner of entries: 16385
Source                          Destionation                  Protocol
192.168.17.30:58337             192.168.100.40:5007            udp [17]
```

```
192.168.17.30:37468           192.168.100.40:5001           udp [17]
192.168.17.30:38719           192.168.100.40:5001           udp [17]
192.168.17.30:51328           192.168.100.40:5009           udp [17]
192.168.17.30:47237           192.168.100.40:5001           udp [17]
192.168.17.30:40794           192.168.100.40:5002           udp [17]
192.168.17.30:46375           192.168.100.40:5001           udp [17]
192.168.17.30:57484           192.168.100.40:5000           udp [17]
192.168.17.30:59403           192.168.100.40:5001           udp [17]
192.168.17.30:46396           192.168.100.40:5001           udp [17]
192.168.17.30:41066           192.168.100.40:5001           udp [17]
192.168.17.30:36309           192.168.100.40:5009           udp [17]
192.168.17.30:47408           192.168.100.40:5006           udp [17]
192.168.17.30:32963           192.168.100.40:5003           udp [17]
192.168.17.30:33957           192.168.100.40:5002           udp [17]
192.168.17.30:40226           192.168.100.40:5005           udp [17]
192.168.17.30:38901           192.168.100.40:5005           udp [17]
192.168.17.30:53885           192.168.100.40:5004           udp [17]
192.168.17.30:46619           192.168.100.40:5002           udp [17]
192.168.17.30:39099           192.168.100.40:5007           udp [17]
192.168.17.30:48855           192.168.100.40:5008           udp [17]
192.168.17.30:44828           192.168.100.40:5009           udp [17]
192.168.17.30:47916           192.168.100.40:5009           udp [17]
```

show connsync statistics

Displays statistics for connection synchronization.

Syntax:

```
show connsync statistics
```

Operational mode

Connection synchronization must be configured on the system to use this command.

Use this command display statistics for connection synchronization.

The errors in the statistics report indicate that the connection synchronization messages are not delivered to connection synchronization clients correctly to connection synchronization clients. The error counter increases when memory allocation issues, communication issues due to wrong messaging, or an installation error due to a version mismatch among peers occurs. Therefore, when the error counter increases, the administrator miust ensure that the same image version is running on both connection synchronization peers and then reboot the peers.

The following example shows how to display statistics for connection synchronization.

```
vyatta@vyatta:~$ show connsync statistics
local:
msg: tx/rx 0/456880 tx_err/rx_err 0/0 tx_end/rx_end 1/1
cache: size/max: 65292/1048576       insert: 65292 update: 391587
       delete: 0 expirted: 0 evicted: 0
       err: update/delete/stale 0/0/0
remote:
msg: tx/rx 0/1 tx_err/rx_err 0/0 tx_end/rx_end 1/1
cache: size/max: 0/1048576       insert: 0 update: 0
       delete: 0 expirted: 0 evicted: 0
       err: update/delete/stale 0/0/0
```

show connsync status

Displays status of connection synchronization.

Syntax:

```
show connsync status
```

Operational mode

Connection synchronization must be configured on the system to use this command.

Use this command to display the status of connection synchronization.

Supported Data Plane Interfaces

The following table shows the syntax and parameters of the supported types of data plane interfaces.

hack: broke the first row into several rows to make page break because table width doesn't work yet.

Interface Type	Syntax	Parameters
Data plane	`dataplane` *interface-name*	*interface-name*: The name of a data plane interface. Following are the supported formats of the interface name: • `dpxpypz`—The name of a data plane interface, where — `dpx` specifies the data plane identifier (ID). Currently, only dp0 is supported. — `py` specifies a physical or virtual PCI slot index (for example, p129). — `pz` specifies a port index (for example, p1). For example, dp0p1p2, dp0p160p1, and dp0p192p1.
		• `dpxemy` —The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where `emy` specifies an embedded network interface number (typically, a small number). For example, dp0em3.
		• `dpxsy`—The name of a data plane interface in a system in which the BIOS identifies the network interface card to reside in a particular physical or virtual slot *y*, where *y* is typically a small number. For example, for the dp0s2 interface, the BIOS identifies slot 2 in the system to contain this interface.

Interface Type	Syntax	Parameters
		• `dpxPnpypz` —The name of a data plane interface on a device that is installed on a secondary PCI bus, where `Pn` specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.
		• `dp0vrrp`*N* — The name of a VRRP data plane interface, where *N* is the nth VRRP group that is created. This name is generated by the router when rfc-compatibility mode is used. The first created interface is dp0vrrp1, the second is dp0vrrp2, and so on.
Data plane vif	`dataplane` *interface-name* `vif` *vif-id* [`vlan` *vlan-id*]	*interface-name*: Refer to the preceding description. *vif-id*: A virtual interface ID. The ID ranges from 1 through 4094. *vlan-id*: The VLAN ID of a virtual interface. The ID ranges from 1 through 4094.

VRF Support

VRF support for VRRP

VRF allows an AT&T Vyatta vRouter to support multiple routing tables, one for each VRF routing instance. Virtual Router Redundancy Protocol (VRRP) operates within the context of a single Layer 3 IP subnet, so its operation is not affected by VRF routing instances. Note that VRRP operates appropriately for directly connected interfaces in different routing instances with overlapping address spaces.

The following example shows that the dp0s11 interface is bound to the BLUE routing instance and the dp0s4 interface is bound to the RED routing instance.

```
routing {
 routing-instance BLUE {
  instance-type vrf
  interface dp0s11
 }
 routing-instance RED {
  instance-type vrf
  interface dp0s4
 }
}
```

The following example shows that VRRP for IPv4 is configured with overlapping addresses for the interfaces that were created previously in the RED and BLUE routing instances.

```
dataplane dp0s4 {
 address 11.0.0.2/24
 vrrp{
  vrrp-group 1 {
   priority 254
   version 3
   virtual-address 11.0.0.10
  }
}
dataplane dp0s11 {
 address 11.0.0.2/24
 vrrp{
  vrrp-group 1 {
  priority 254
  version 3
  virtual-address 11.0.0.10
 }
}
```

The following example command sequence shows that VRRP for IPv6 is configured with overlapping addresses for the interfaces that were created previously in the RED and BLUE routing instances.

```
dataplane dp0s4 {
 address 2001::2/64
vrrp{
 vrrp-group 3 {
   priority 254
   version 3
   virtual-address 2001::10
   virtual-address fe80::10
  }
 }
}
dataplane dp0s11 {
 address 2001::2/64
```

```
vrrp{
 vrrp-group 3 {
  priority 254
  version 3
  virtual-address 2001::10
  virtual-address fe80::10
 }
}
}
```

For more information about VRRP, see AT&T Vyatta Network Operating System Basic System Configuration Guide.

Command support for VRF routing instances

VRF allows an AT&T Vyatta vRouter to support multiple routing tables, one for each VRF routing instance. Some commands in this guide support VRF and can be applied to particular routing instances.

Use the guidelines in this section to determine correct syntax when adding VRF routing instances to commands. For more information about VRF, refer to AT&T Vyatta Network Operating System Basic Routing Configuration Guide. This guide includes an overview of VRF, VRF configuration examples, information about VRF-specific features, and a list of commands that support VRF routing instances.

Adding a VRF routing instance to a Configuration mode command

For most Configuration mode commands, specify the VRF routing instance at the beginning of a command. Add the appropriate VRF keywords and variable to follow the initial action (**`set`**, **`show`**, or **`delete`**) and before the other keywords and variables in the command.

Example: Configuration mode example: syslog

The following command configures the syslog logging level for the specified syslog host. The command does not include a VRF routing instance, so the command applies to the default routing instance.

```
vyatta@R1# set system syslog host 10.10.10.1 facility all level debug
vyatta@R1# show system syslog
syslog {
    host 10.10.10.1 {
            facility all {
                    level debug
            }
    }
}
```

The following example shows the same command with the VRF routing instance (GREEN) added. Notice that **`routing routing-instance GREEN`** has been inserted between the basic action (**set** in the example) and the rest of the command. Most Configuration mode commands follow this convention.

```
vyatta@R1# set routing routing-instance GREEN system syslog host 10.10.10.1 facility all
 level debug
vyatta@R1# show routing
routing {
     routing-instance GREEN {
             system {
                     syslog {
                             host 11.12.13.2:514 {
                                     facility all {
                                             level debug
                                     }
                             }
                     }
             }
     }
```

```
}
```

Example: Configuration mode example: SNMP

Some features, such as SNMP, are not available on a per-routing instance basis but can be bound to a specific routing instance. For these features, the command syntax is an exception to the convention of specifying the routing instance at the beginning of Configuration mode commands.

The following example shows how to configure the SNMPv1 or SNMPv2c community and context for the RED and BLUE routing instances. The first two commands specify the RED routing instance as the context for community A and BLUE routing instance as the context for community B. The subsequent commands complete the configuration.

For more information about configuring SNMP, refer to AT&T Vyatta Network Operating System Remote Management Configuration Guide.

```
vyatta@R1# set service snmp community commA context RED
vyatta@R1# set service snmp community commB context BLUE
vyatta@R1# set service snmp view all oid 1
vyatta@R1# set service snmp community commA view all
vyatta@R1# set service snmp community commB view all
vyatta@R1# show service snmp community
 community commA {
        context RED
        view all
 }
 community commB {
        context BLUE
        view all
 }
[edit]
vyatta@vyatta#
```

Adding a VRF routing instance to an Operational mode command

The syntax for adding a VRF routing instance to an Operational mode command varies according to the type of command parameters:

- If the command does not have optional parameters, specify the routing instance at the end of the command.
- If the command has optional parameters, specify the routing instance after the required parameters and before the optional parameters.

Example: Operational mode examples without optional parameters

The following command displays dynamic DNS information for the default routing instance.

```
vyatta@vyatta:~$ show dns dynamic status
```

The following command displays the same information for the specified routing instance (GREEN). The command does not have any optional parameters, so the routing instance is specified at the end of the command.

```
vyatta@vyatta:~$ show dns dynamic status routing-instance GREEN
```

Example: Operational mode example with optional parameters

The following command obtains multicast path information for the specified host (10.33.2.5). A routing instance is not specified, so the command applies to the default routing instance.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 detail
```

The following command obtains multicast path information for the specified host (10.33.2.5) and routing instance (GREEN). Notice that the routing instance is specified before the optional **detail** keyword.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 routing-instance GREEN detail
```

Example: Operational mode example output: SNMP

The following SNMP **show** commands display output for routing instances.

```
vyatta@vyatta:~$ show snmp routing-instance
Routing Instance SNMP Agent is Listening on for Incoming Requests:
Routing-Instance            RDID
-----------------           ----
RED                         5

vyatta@vyatta:~$ show snmp community-mapping
SNMPv1/v2c Community/Context Mapping:
Community                   Context
---------                   -------
commA                       'RED'
commB                       'BLUE'
deva                        'default'


vyatta@vyatta:~$ show snmp trap-target
SNMPv1/v2c Trap-targets:
Trap-target                   Port   Routing-Instance Community
-----------                   ----   ---------------- ---------
1.1.1.1                              'RED'           'test'


vyatta@vyatta:~$ show snmp v3 trap-target
SNMPv3 Trap-targets:
Trap-target                   Port   Protocol Auth Priv Type   EngineID              Routing-
Instance User
-----------                   ----   -------- ---- ---- ----   --------
 ---------------- ----
2.2.2.2                       '162'  'udp'    'md5'      'infor                        'BLUE'
         'test'
```

List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers

Acronym	Description
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM

Acronym	Description
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access