

Configuring Firewall Stateful Interchassis Redundancy

The Firewall Stateful Interchassis Redundancy feature enables you to configure pairs of routers to act as backup for each other. This feature can be configured to determine the active router based on a number of failover conditions. When a failover occurs, the standby router seamlessly takes over and starts performing traffic forwarding services and maintaining a dynamic routing table.

- Prerequisites for Firewall Stateful Interchassis Redundancy, on page 1
- Restrictions for Firewall Stateful Interchassis Redundancy, on page 1
- Information About Firewall Stateful Interchassis Redundancy, on page 2
- How to Configure Firewall Stateful Interchassis Redundancy, on page 6
- Configuration Examples for Firewall Stateful Interchassis Redundancy, on page 14
- Additional References for Firewall Stateful Interchassis Redundancy, on page 18
- Feature Information for Firewall Stateful Interchassis Redundancy, on page 18

Prerequisites for Firewall Stateful Interchassis Redundancy

- The interfaces attached to the firewall must have the same redundant interface identifier (RII).
- The active device and the standby device must have the same Cisco IOS XE Zone-Based Firewall configuration.
- The active device and the standby device must run on an identical version of the Cisco IOS XE software. The active device and the standby device must be connected through a switch.
- Embedded Service Processor (ESP) must match on both active and standby devices.

Restrictions for Firewall Stateful Interchassis Redundancy

- LAN and MESH scenarios are not supported.
- Cisco ASR 1006 and Cisco ASR 1013 platforms with dual Embedded Services Processors (ESPs) or dual Route Processors (RPs) in the chassis are not supported, because coexistence of interbox high availability (HA) and intrabox HA is not supported.

Cisco ASR 1006 and Cisco ASR 1013 platforms with single ESP and single RP in the chassis supports interchassis redundancy.

- If the dual IOS daemon (IOSd) is configured, the device will not support the firewall Stateful Interchassis Redundancy configuration.

Information About Firewall Stateful Interchassis Redundancy

How Firewall Stateful Inter-Chassis Redundancy Works

You can configure pairs of routers to act as hot standbys for each other. This redundancy is configured on an interface basis. Pairs of redundant interfaces are known as redundancy groups. The figure below depicts the active-standby device scenario. It shows how the redundancy group is configured for a pair of routers that has one outgoing interface. The *Redundancy Group Configuration--Two Outgoing Interfaces* figure depicts the active-active device scenario shows how two redundancy groups are configured for a pair of routers that have two outgoing interfaces.

Note that in both cases, the redundant routers are joined by a configurable control link and a data synchronization link. The control link is used to communicate the status of the routers. The data synchronization link is used to transfer stateful information from Network Address Translation (NAT) and the firewall and to synchronize the stateful database for these applications.

Also, in both cases, the pairs of redundant interfaces are configured with the same unique ID number known as the RII.

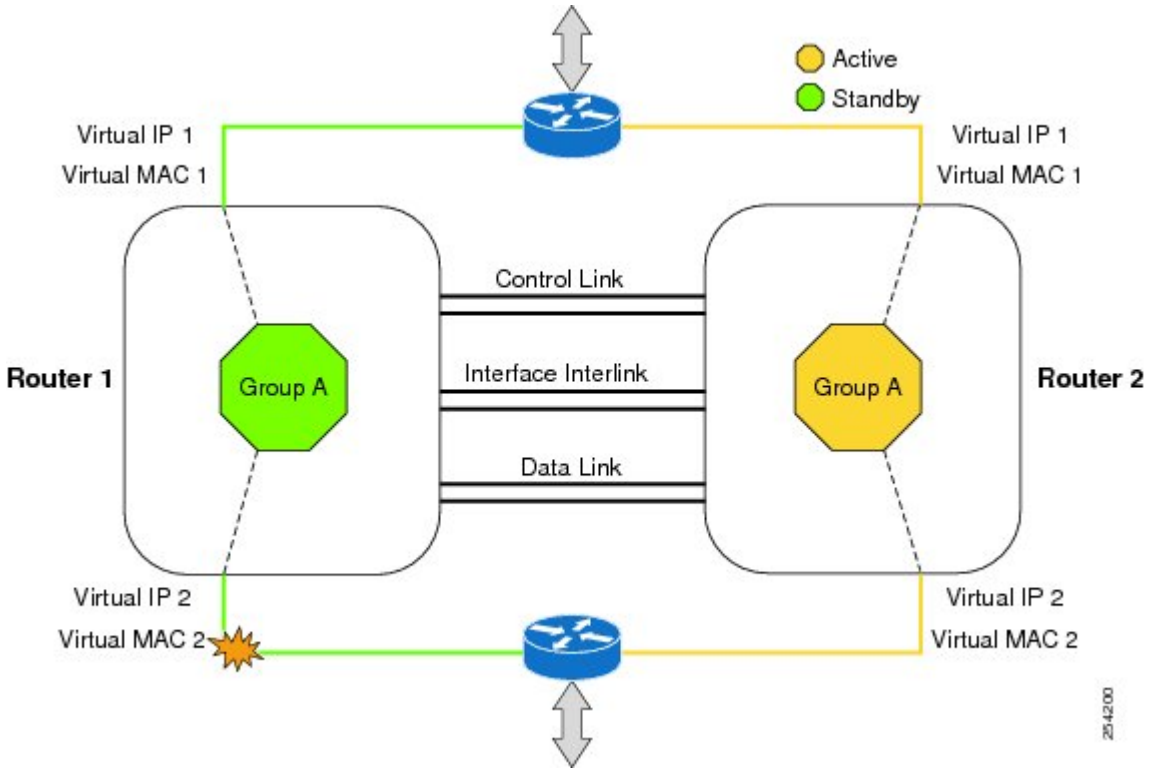
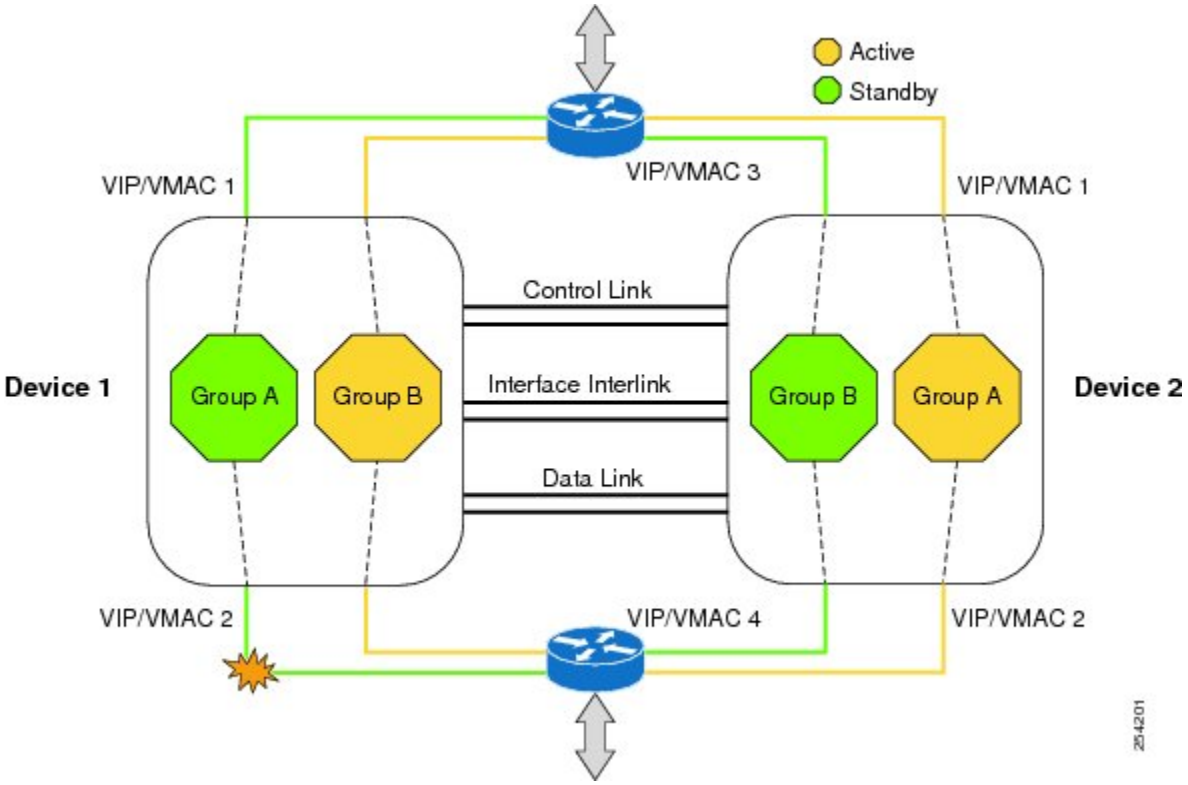


Figure 1: Redundancy Group Configuration--Two Outgoing Interfaces



The status of redundancy group members is determined through the use of hello messages sent over the control link. If either of the routers does not respond to a hello message within a configurable amount of time, it is considered that a failure has occurred, and a switchover is initiated. To detect a failure in milliseconds, the control links run the failover protocol integrated with the Bidirectional Forwarding Detection (BFD) protocol. You can configure the following parameters for the hello messages:

- Active timer
- Standby timer
- Hellotime--The interval at which hello messages are sent
- Holdtime--The amount of time before the active or the standby router is declared to be down

The hellotime defaults to 3 seconds to align with Hot Standby Router Protocol (HSRP), and the holdtime defaults to 10 seconds. You can also configure these timers in milliseconds by using the **timers hellotime msec** command.

To determine which pairs of interfaces are affected by the switchover, you must configure a unique ID number for each pair of redundant interfaces. This ID number is known as the RII associated with the interface.

A switchover to the standby router can also occur under other circumstances. Another factor that can cause a switchover is a priority setting that is configurable for each router. The router with the highest priority value will be the active router. If a fault occurs on either the active or the standby router, the priority of the router is decremented by a configurable amount known as the weight. If the priority of the active router falls below the priority of the standby router, a switchover occurs and the standby router becomes the active router. This default behavior can be overridden by disabling the preemption attribute for the redundancy group. You can

also configure each interface to decrease the priority when the L1 state of the interface goes down. This amount overrides the default amount configured for the redundancy group.

Each failure event that causes a modification of a redundancy group's priority generates a syslog entry that contains a time stamp, the redundancy group that was affected, previous priority, new priority, and a description of the failure event cause.

Another situation that will cause a switchover to occur is when the priority of a router or interface falls below a configurable threshold level.

In general, a switchover to the standby router occurs under the following circumstances:

- Power loss or reload occurs on the active router (this includes crashes).
- The run-time priority of the active router goes down below that of the standby router.
- The run-time priority of the active router goes down below the configured threshold value.
- The redundancy group on the active router is reloaded manually using the **redundancy application reload group** *rg-number* command.
- Two consecutive hello messages missed on any monitored interface forces the interface into testing mode. When this occurs, both units first verify the link status on the interface and then execute the following tests:
  - Network activity test
  - ARP test
  - Broadcast ping test

In the Firewall Stateful Inter-Chassis Redundancy feature, the redundancy group traffic is routed through the virtual IP address that is associated with the ingress interface of the redundancy group. The traffic sent to the virtual IP address is received by the router that has the redundancy group in the active state. During a redundancy group failover, the traffic to the virtual IP address is automatically routed to the newly active redundancy group.

The firewall drops the traffic that arrives on the standby redundancy group in case the redundancy group traffic is routed through the physical IP address of a standby router and the traffic reaches the standby redundancy group. However, when the traffic arrives on the active redundancy group, the established TCP or UDP sessions are synchronized to the standby redundancy group.

Exclusive Virtual IP Addresses and Exclusive Virtual MAC Addresses

Virtual IP (VIP) addresses and virtual MAC (VMAC) addresses are used by security applications to control interfaces that receive traffic. An interface is paired with another interface, and these interfaces are associated with the same redundancy group (RG). The interface that is associated with an active RG exclusively owns the VIP and VMAC. The Address Resolution Protocol (ARP) process on the active device sends ARP replies for any ARP request for the VIP, and the Ethernet controller for the interface is programmed to receive packets destined for the VMAC. When an RG failover occurs, the ownership of the VIP and VMAC changes. The interface that is associated with the newly active RG sends a gratuitous ARP and programs the interface's Ethernet controller to accept packets destined for the VMAC.

IPv6 Support

You can assign each redundancy group (RG) on a traffic interface for both IPv4 and IPv6 virtual IP (VIP) addresses under the same redundancy interface identifier (RII). Each RG uses a unique virtual MAC (VMAC) address per RII. For an RG, the IPv6 link-local VIP and global VIP coexist on an interface.

You can configure an IPv4 VIP, a link-local IPv6 VIP, and/or a global IPv6 VIP for each RG on a traffic interface. IPv6 link-local VIP is mainly used when configuring static or default routes, whereas IPv6 global VIP is widely used in both LAN and WAN topologies.

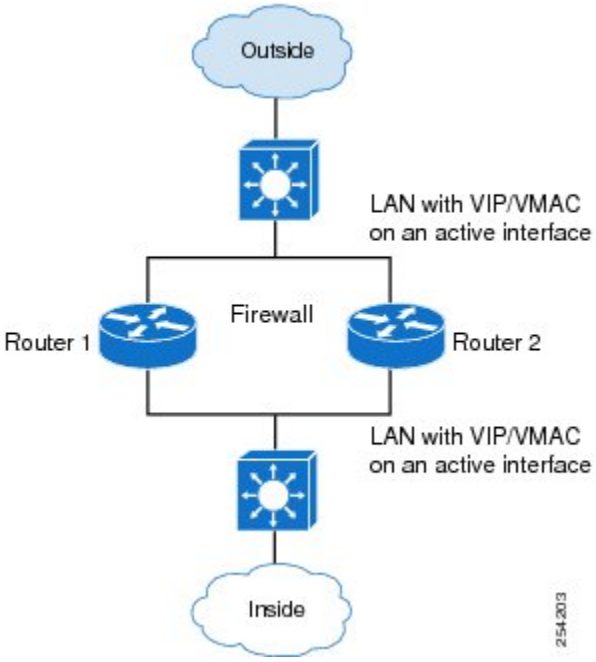
You must configure a physical IP address before configuring an IPv4 VIP.

Supported Topologies

The LAN-LAN topology is supported in the Firewall Stateful Inter-Chassis Redundancy architecture:

LAN-LAN

The figure below shows the LAN-LAN topology. When a dedicated appliance-based firewall solution is used, traffic is often directed to the correct firewall by configuring static routing in the upstream or downstream routers to an appropriate virtual IP address. In addition, the Aggregation Services Routers (ASRs) will participate in dynamic routing with upstream or downstream routers. The dynamic routing configuration supported on LAN facing interfaces must not introduce a dependency on routing protocol convergence; otherwise, fast failover requirements will not be met.



For more information about the LAN-LAN configuration, see the section, Example Configuring LAN-LAN.

VRF-Aware Interchassis Redundancy in Zone-Based Firewalls

In Cisco IOS XE Release 3.14S, zone-based firewalls support VRF-aware interchassis redundancy. The VPN routing and forwarding (VRF) name at the active and standby devices must the same. The same VRF configuration must be available on both active and standby devices.

The VRF-Aware Interchassis Redundancy in Zone-Based Firewalls feature uses a VRF mapping mechanism that sends the VRF hash key along with box-to-box high availability session sync messages across active and standby devices.

How to Configure Firewall Stateful Interchassis Redundancy

Configuring a Redundancy Application Group

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **name** *group-name*
7. **shutdown**
8. **priority** *value* [**failover threshold** *value*]
9. **preempt**
10. **track** *object-number* {**decrement** *value* | **shutdown**}
11. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.

	Command or Action	Purpose
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-grp)# name group1`	(Optional) Specifies an optional alias for the protocol instance.
Step 7	**shutdown** **Example:** `Device(config-red-app-grp)# shutdown`	(Optional) Shuts down a redundancy group manually.
Step 8	**priority** *value* [**failover threshold** *value*] **Example:** `Device(config-red-app-grp)# priority 100 failover threshold 50`	(Optional) Specifies the initial priority and failover threshold for a redundancy group.
Step 9	**preempt** **Example:** `Device(config-red-app-grp)# preempt`	Enables preemption on the group and enables the standby device to preempt the active device regardless of the priority.
Step 10	**track** *object-number* {**decrement** *value* | **shutdown**} **Example:** `Device(config-red-app-grp)# track 200 decrement 200`	Specifies the priority value of a redundancy group that will be decremented if an event occurs.
Step 11	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Configuring a Redundancy Group Protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **protocol** *id*
6. **name** *group-name*

7. **timers hellotime** {*seconds* | **msec** *milliseconds*} **holdtime** {*seconds* | **msec** *milliseconds*}
8. **authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key-string* **timeout** *seconds* | **key-chain** *key-chain-name*}
9. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**protocol** *id* **Example:** `Device(config-red-app)# protocol 1`	Specifies the protocol instance that will be attached to a control interface and enters redundancy application protocol configuration mode.
Step 6	**name** *group-name* **Example:** `Device(config-red-app-prtcl)# name prot1`	(Optional) Configures the redundancy group (RG) with a name.
Step 7	**timers hellotime** {*seconds* | **msec** *milliseconds*} **holdtime** {*seconds* | **msec** *milliseconds*} **Example:** `Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9`	Specifies the interval between when hello messages are sent and the time period before which a device is declared to be down.
Step 8	**authentication** {**text** *string* | **md5 key-string** [**0** | **7**] *key-string* **timeout** *seconds* | **key-chain** *key-chain-name*} **Example:** `Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100`	Specifies the authentication information.

	Command or Action	Purpose
Step 9	**end** **Example:** `Device(config-red-app-prtcl)# end`	Exits redundancy application protocol configuration mode and enters privileged EXEC mode.

Configuring a Virtual IP Address and a Redundant Interface Identifier

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **redundancy rii** *id*
5. **redundancy group** *id* **ip** *virtual-ip* **exclusive** [**decrement** *value*]
6. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 4	**redundancy rii** *id* **Example:** `Device(config-if)# redundancy rii 600`	Configures the redundancy interface identifier (RII) for a redundancy group. • The range is from 1 to 65535.
Step 5	**redundancy group** *id* **ip** *virtual-ip* **exclusive** [**decrement** *value*] **Example:** `Device(config-if)# redundancy group 1 ip 10.10.1.1 exclusive decrement 20`	Associates an interface with a redundancy group and enables a virtual IP address.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring a Control Interface and a Data Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **application redundancy**
5. **group** *id*
6. **data** *interface-type interface-number*
7. **control** *interface-type interface-number* **protocol** *id*
8. **timers delay** *seconds* [**reload** *seconds*]
9. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**redundancy** **Example:** `Device(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**application redundancy** **Example:** `Device(config-red)# application redundancy`	Enters redundancy application configuration mode.
Step 5	**group** *id* **Example:** `Device(config-red-app)# group 1`	Enters redundancy application group configuration mode.

	Command or Action	Purpose
Step 6	**data** *interface-type interface-number* **Example:** `Device(config-red-app-grp)# data GigabitEthernet 0/0/0`	Specifies the data interface that is used by the redundancy group.
Step 7	**control** *interface-type interface-number* **protocol** *id* **Example:** `Device(config-red-app-grp)# control gigabitethernet 0/0/2 protocol 1`	Specifies the control interface that is used by the redundancy group. • This interface is also associated with an instance of the control interface protocol.
Step 8	**timers delay** *seconds* [**reload** *seconds*] **Example:** `Device(config-red-app-grp)# timers delay 100 reload 400`	Specifies the time that a redundancy group will take to delay role negotiations that start after a fault occurs or the system is reloaded.
Step 9	**end** **Example:** `Device(config-red-app-grp)# end`	Exits redundancy application group configuration mode and enters privileged EXEC mode.

Managing and Monitoring Firewall Stateful Inter-Chassis Redundancy

Use the following commands to manage and monitor the Firewall Stateful Inter-Chassis Redundancy feature.

SUMMARY STEPS

1. **enable**
2. **debug redundancy application group config** {**all** | **error** | **event** | **func**}
3. **debug redundancy application group faults** {**all** | **error** | **event** | **fault** | **func**}
4. **debug redundancy application group media** {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**}
5. **debug redundancy application group protocol** {**all** | **detail** | **error** | **event** | **media** | **peer**}
6. **debug redundancy application group rii** {**error** | **event**}
7. **debug redundancy application group transport** {**db** | **error** | **event** | **packet** | **timer** | **trace**}
8. **debug redundancy application group vp** {**error** | **event**}
9. **show redundancy application group** [*group-id* | **all**]
10. **show redundancy application transport** {**client** | **group** [*group-id*]}
11. **show redundancy application control-interface group** [*group-id*]
12. **show redundancy application faults group** [*group-id*]
13. **show redundancy application protocol** {*protocol-id* | **group** [*group-id*]
14. **show redundancy application if-mgr group** [*group-id*]
15. **show redundancy application data-interface group** [*group-id*]
16. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug redundancy application group config** {**all** | **error** | **event** | **func**} **Example:** `Device# debug redundancy application group config all`	Displays the redundancy group application configuration.
Step 3	**debug redundancy application group faults** {**all** | **error** | **event** | **fault** | **func**} **Example:** `Device# debug redundancy application group faults error`	Displays the redundancy group application fault.
Step 4	**debug redundancy application group media** {**all** | **error** | **event** | **nbr** | **packet** {**rx** | **tx**} | **timer**} **Example:** `Device# debug redundancy application group media timer`	Displays the redundancy group application group media information.
Step 5	**debug redundancy application group protocol** {**all** | **detail** | **error** | **event** | **media** | **peer**} **Example:** `Device# debug redundancy application group protocol peer`	Displays the redundancy group application group protocol information.
Step 6	**debug redundancy application group rii** {**error** | **event**} **Example:** `Device# debug redundancy application group rii event`	Displays the redundancy group application group RII information.
Step 7	**debug redundancy application group transport** {**db** | **error** | **event** | **packet** | **timer** | **trace**} **Example:**	Displays the redundancy group application group transport information.

	Command or Action	Purpose
	`Device# debug redundancy application group transport trace`	
Step 8	**debug redundancy application group vp** {**error** | **event**} **Example:** `Device# debug redundancy application group vp event`	Displays the redundancy group application group VP information.
Step 9	**show redundancy application group** [*group-id* | **all**] **Example:** `Device# show redundancy application group all`	Displays the redundancy group information.
Step 10	**show redundancy application transport** {**client** | **group** [*group-id*]} **Example:** `Device# show redundancy application transport group 1`	Displays transport specific information for a redundancy group.
Step 11	**show redundancy application control-interface group** [*group-id*] **Example:** `Device# show redundancy application control-interface group 2`	Displays control interface information for a redundancy group.
Step 12	**show redundancy application faults group** [*group-id*] **Example:** `Device# show redundancy application faults group 2`	Displays fault-specific information for a redundancy group.
Step 13	**show redundancy application protocol** {*protocol-id* | **group** [*group-id*] **Example:** `Device# show redundancy application protocol 3`	Displays protocol specific information for a redundancy group.
Step 14	**show redundancy application if-mgr group** [*group-id*] **Example:** `Device# show redundancy application if-mgr group 2`	Displays interface manager information for a redundancy group.
Step 15	**show redundancy application data-interface group** [*group-id*]	Displays data interface specific information.

	Command or Action	Purpose
	Example: `Device# show redundancy application data-interface group 1`	
Step 16	**end** **Example:** `Device# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Configuration Examples for Firewall Stateful Interchassis Redundancy

Example: Configuring a Redundancy Application Group

The following example shows how to configure a redundancy group named group1 with priority and preempt attributes:

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# group 1
Device(config-red-app-grp)# name group1
Device(config-red-app-grp)# priority 100 failover-threshold 50
Device(config-red-app-grp)# preempt
Device(config-red-app-grp)# track 200 decrement 200
Device(config-red-app-grp)# end
```

Example: Configuring a Redundancy Group Protocol

The following example shows how to configure a redundancy group with timers set for hello time and hold time messages:

```
Device# configure terminal
Device(config)# redundancy
Device(config-red)# application redundancy
Device(config-red-app)# protocol 1
Device(config-red-app-prtcl)# timers hellotime 3 holdtime 9
Device(config-red-app-prtcl)# authentication md5 key-string 0 n1 timeout 100
Device(config-red-app-prtcl)# bfd
Device(config-red-app-prtcl)# end
```

Example: Configuring a Virtual IP Address and a Redundant Interface Identifier

The following example shows how to configure the redundancy group virtual IP address for Gigabit Ethernet interface 0/1/1:

```
Device# configure terminal
```

```
Device(config)# interface GigabitEthernet 0/1/1
Device(conf-if)# redundancy rii 600
Device(config-if)# redundancy group 2 ip 10.2.3.4 exclusive decrement 200
Device(config-if)# end
```

Example: Configuring a Control Interface and a Data Interface

```
Device# configure terminal
Device(config-red)# application redundancy
Device(config-red-app-grp)# group 1
Device(config-red-app-grp)# data GigabitEthernet 0/0/0
Device(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1
Device(config-red-app-grp)# timers delay 100 reload 400
Device(config-red-app-grp)# end
```

Example: Configuring a LAN-LAN Topology

The following is a sample LAN-LAN configuration that shows how a pair of routers that have two outgoing interfaces are configured for stateful redundancy. In this example, GigabitEthernet 0/1/1 is the ingress interface and GigabitEthernet 0/2/1 is the egress interface. Both interfaces are assigned to zones and a classmap is defined to describe the traffic between zones. Interfaces are also configured for redundancy. The “inspect” action invokes the application-level gateway (ALG) to open a pinhole to allow traffic on other ports. A pinhole is a port that is opened through an ALG to allow a particular application to gain controlled access to a protected network.

The following is the configuration on Device 1, the active device.

```
! Configures redundancy, control and data interfaces
redundancy
 mode none
 application redundancy
  group 2
   preempt
   priority 200 failover threshold 100
   control GigabitEthernet 0/0/4 protocol 2
   data GigabitEthernet 0/0/3
!
  protocol 2
   timers hellotime ms 250 holdtime ms 750
!
! Configures a VRF
ip vrf vrf1
!
! Configures parameter maps to add parameters that control the behavior of actions and match
 criteria.
parameter-map type inspect pmap-udp
 redundancy
  redundancy delay 10
!
parameter-map type inspect pmap-tcp
 redundancy
  redundancy delay 10
!
! Defines class-maps to describes traffic between zones
class-map type inspect match-any cmap-udp
 match protocol udp
!
class-map type inspect match-any cmap-ftp-tcp
 match protocol ftp
```

```
 match protocol tcp
!
! Associates class-maps with policy-maps to define actions to be applied
policy-map type inspect p1
 class type inspect cmap-udp
  inspect pmap-udp
!
 class type inspect cmap-ftp-tcp
  inspect pmap-tcp
!
! Identifies and defines network zones
zone security z-int
!
zone security z-hi
!
! Sets zone pairs for any policy other than deny all and assign policy-maps to zone-pairs
by defining a service-policy
zone-pair security hi2int source z-hi destination z-int
 service-policy type inspect p1
!
! Assigns interfaces to zones
interface GigabitEthernet 0/0/1
 ip vrf forwarding vrf1
 ip address 10.1.1.3 255.255.0.0
 ip virtual-reassembly
 zone-member security z-hi
 negotiation auto
 redundancy rii 20
 redundancy group 2 ip 10.1.1.10 exclusive decrement 50
!
interface GigabitEthernet 0/0/2
 ip vrf forwarding vrf1
 ip address 192.0.2.2 255.255.255.240
 ip virtual-reassembly
 zone-member security z-int
 negotiation auto
 redundancy rii 21
 redundancy group 2 ip 192.0.2.12 exclusive decrement 50
!
interface GigabitEthernet 0/0/4
 ip address 198.51.100.17 255.255.255.240
!
interface GigabitEthernet 0/0/4
 ip address 203.0.113.49 255.255.255.240
!
ip route vrf vrf1 192.0.2.0 255.255.255.240 GigabitEthernet0/0/2 10.1.1.4
ip route vrf vrf1 10.1.0.0 255.255.0.0 GigabitEthernet0/0/1 10.1.0.4
!
```

The following is the configuration on Device 2, the standby device:

```
! Configures redundancy, control and data interfaces
redundancy
 mode none
 application redundancy
  group 2
   preempt
   priority 200 failover threshold 100
   control GigabitEthernet 0/0/4 protocol 2
   data GigabitEthernet 0/0/3
!
  protocol 2
   timers hellotime ms 250 holdtime ms 750
!
! Configures a VRF
```

```
ip vrf vrf1
!
! Configures parameter maps to add parameters that control the behavior of actions and match
 criteria.
parameter-map type inspect pmap-udp
 redundancy
  redundancy delay 10
!
parameter-map type inspect pmap-tcp
 redundancy
  redundancy delay 10
!
! Defines class-maps to describes traffic between zones
class-map type inspect match-any cmap-udp
 match protocol udp
!
class-map type inspect match-any cmap-ftp-tcp
 match protocol ftp
 match protocol tcp
!
! Associates class-maps with policy-maps to define actions to be applied
policy-map type inspect p1
 class type inspect cmap-udp
  inspect pmap-udp
!
 class type inspect cmap-ftp-tcp
  inspect pmap-tcp
!
! Identifies and defines network zones
zone security z-int
!
zone security z-hi
!
! Sets zone pairs for any policy other than deny all and assign policy-maps to zone-pairs
by defining a service-policy
zone-pair security hi2int source z-hi destination z-int
 service-policy type inspect p1
!
! Assigns interfaces to zones
interface GigabitEthernet 0/0/1
 ip vrf forwarding vrf1
 ip address 10.1.1.6 255.255.0.0
 ip virtual-reassembly
 zone-member security z-hi
 negotiation auto
 redundancy rii 20
 redundancy group 2 ip 10.1.1.12 exclusive decrement 50
!
interface GigabitEthernet 0/0/2
 ip vrf forwarding vrf1
 ip address 192.0.2.5 255.255.255.240
 ip virtual-reassembly
 zone-member security z-int
 negotiation auto
 redundancy rii 21
 redundancy group 2 ip 192.0.2.10 exclusive decrement 50
!
interface GigabitEthernet 0/0/4
 ip address 198.51.100.21 255.255.255.240
!
interface GigabitEthernet 0/0/4
 ip address 203.0.113.53 255.255.255.240
!
ip route vrf vrf1 192.0.2.0 255.255.255.240 GigabitEthernet0/0/2 10.1.1.4
```

```
ip route vrf vrf1 10.1.0.0 255.255.0.0 GigabitEthernet0/0/1 10.1.0.4
!
```

Additional References for Firewall Stateful Interchassis Redundancy

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Security Command Reference: Commands A to C • Security Command Reference: Commands D to L • Security Command Reference: Commands M to R • Security Command Reference: Commands S to Z

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Firewall Stateful Interchassis Redundancy

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

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

Table 1: Feature Information for Firewall Stateful Interchassis Redundancy

Feature Name	Releases	Feature Information
Firewall Stateful Interchassis Redundancy	Cisco IOS XE Release 3.1(S)	The Firewall Stateful Interchassis Redundancy feature enables you to configure pairs of devices to act a backups for each other. The following commands were introduced or modified: **application redundancy**, **authentication**, **control**, **data**, **debug redundancy application group config**, **debug redundancy application group faults**, **debug redundancy application group media**, **debug redundancy application group protocol**, **debug redundancy application group rii**, **debug redundancy application group transport**, **debug redundancy application group vp**, **group**, **name**, **preempt**, **priority**, **protocol**, **redundancy rii**, **redundancy group**, **track**, **timers delay**, **timers hellotime**, **show redundancy application group**, **show redundancy application transport**, **show redundancy application control-interface**, **show redundancy application faults**, **show redundancy application protocol**, **show redundancy application if-mgr**, **show redundancy application data-interface**.
VRF-Aware Stateful Interchassis Redundancy in Zone-Based Firewalls	Cisco IOS XE Release 3.14S	In Cisco IOS XE Release 3.14S, zone-based firewalls support VRF-aware interchassis redundancy. The VPN routing and forwarding (VRF) name at the active and standby devices must the same. The same VRF configuration must be available on both active and standby devices.