Chapter 36

Host Redundancy Overview

You monitor redundant Routing Engines, host modules, and host subsystems to provide a standby Routing Engine and controller component that will switch from standby to active, assuming mastership, with limited running downtime when a failure occurs.

This chapter provides an overview of how redundant host modules, host subsystems, and Routing Engines work on various routing platforms. Additionally, it describes the topics as listed in Table 106.

Table 106: Checklist for Host Redundancy

Monitor Redundant Routing Engine Tasks	Command or Action
Understanding Redundancy for the Routing Engine, Host Module, and Host Subsystem on page 465	
M10i Router Redundant Routing Engines and HCMs on page 465	
M20 Router Redundant Routing Engines and SSBs on page 466	
M40e and M160 Router Redundant Host Modules on page 467	
M320 Router, T320 Router, and T640 Routing Node Redundant Host Subsystems on page 468	
Routing Engine, Host Module, and Host Subsystem Redundancy Connections on page 469	
Redundancy Connection for an M10i Router on page 470	
Redundancy Connection for an M20 Router on page 471	
Redundancy Connection for an M40e or M160 Router on page 472	
Redundancy Connection for an M320 Router on page 473	
Redundancy Connection for a T320 Router and T640 Routing Node on page 474	
Determining Which Routing Engine You Are Logged In To on page 475	
1. Display Routing Engine Status on page 476	show chassis routing-engine
2. Display the Router Hardware on page 477	show chassis hardware
Determining Routing Engine Mastership on page 477	
1. Determine the Routing Engine Mastership By Checking Status on page 478	show chassis routing-engine
2. Determine Routing Engine Mastership By Checking the LEDs on page 478	Physically check the LEDs on either the craft interface or the Routing Engine (depending on which chassis the Routing Engine is installed).
3. Log In To Backup Routing Engine If graceful-switchover is Configured on page 479	request routing-engine login other-routing-engine

Monitor Redundant Routing Engine Tasks	Command or Action
Manually Configuring Master and Backup Routing Engines on page 479	For slot 0: [edit] set chassis redundancy routing-engine 1 master commit For slot 1: [edit] set chassis redundancy routing-engine 0 backup commit
Manually Switching Routing Engine Mastership on page 482	request chassis routing-engine master (acquire | release | switch)
Determining Why Mastership Switched on page 483	show log mastership
Configuring the Backup Routing Engine to Assume Mastership on Failure of Keepalives on page 486	[edit] set chassis redundancy failover on-loss-of-keepalives set chassis redundancy keepalive-time 300 commit
Avoiding Redundancy Problems on page 487	
1. Operate the Same Type of Routing Engine and JUNOS Software on page 487	The active and standby Routing Engines must be the same type of Routing Engine and must operate the same version of JUNOS software.
2. Use the Groups Configuration on page 487	[edit] set groups *group-name*
3. Synchronize Configurations on page 489	[edit] commit synchronize
4. Copy a Configuration File from One Routing Engine to Another on page 489	file copy *<source>* *<destination>*
5. Use the Proper Shutdown Process on a Backup Routing Engine on page 490	request system halt

See Also

Monitoring Redundant MCSs on page 567

Monitoring Redundant Routing Engines on page 491

Monitoring Redundant Control Boards on page 559

Understanding Redundancy for the Routing Engine, Host Module, and Host Subsystem

Purpose To learn how redundant Routing Engines, host modules, and host subsystems work on various routing platforms. You monitor these components to provide a standby Routing Engine and controller component that will switch from standby to active, assuming mastership when a failure brings down the active master Routing Engine.

What Is a Routing Engine, Host Module, and Host Subsystem Redundancy Redundant Routing Engines are two Routing Engines that are installed in the same routing platform. One functions as the master, while the other stands by as a backup should the master Routing Engine fail. (See "M10i Router Redundant Routing Engines and HCMs" on page 465 and "M20 Router Redundant Routing Engines and SSBs" on page 466.)

Redundant host modules are two Routing Engine and Miscellaneous Control Subsystem (MCS) pairs installed in the same routing platform. One pair functions as master, while the other stands by as a backup should the master Routing Engine fail. (See "M40e and M160 Router Redundant Host Modules" on page 467.)

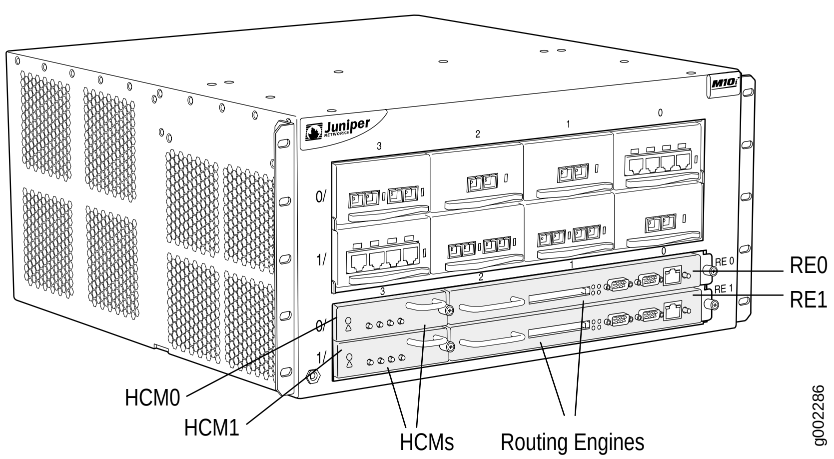
Redundant host subsystems are two Routing Engine and Control Board pairs installed in the same routing platform. One pair functions as master, while the other stands by as backup should the master Routing Engine fail. (See "M320 Router, T320 Router, and T640 Routing Node Redundant Host Subsystems" on page 468.)

The M5, M10, M7i, and M40 routers do not support Routing Engine, host module, or host subsystem redundancy.

M10i Router Redundant Routing Engines and HCMs

On the M10i router, the High-Availability Chassis Manager (HCM) works with its companion Routing Engine to provide control and monitoring functions for router components. The router can have one or two HCMs and Routing Engines. (See Figure 183 and "Redundancy Connection for an M10i Router" on page 470.)

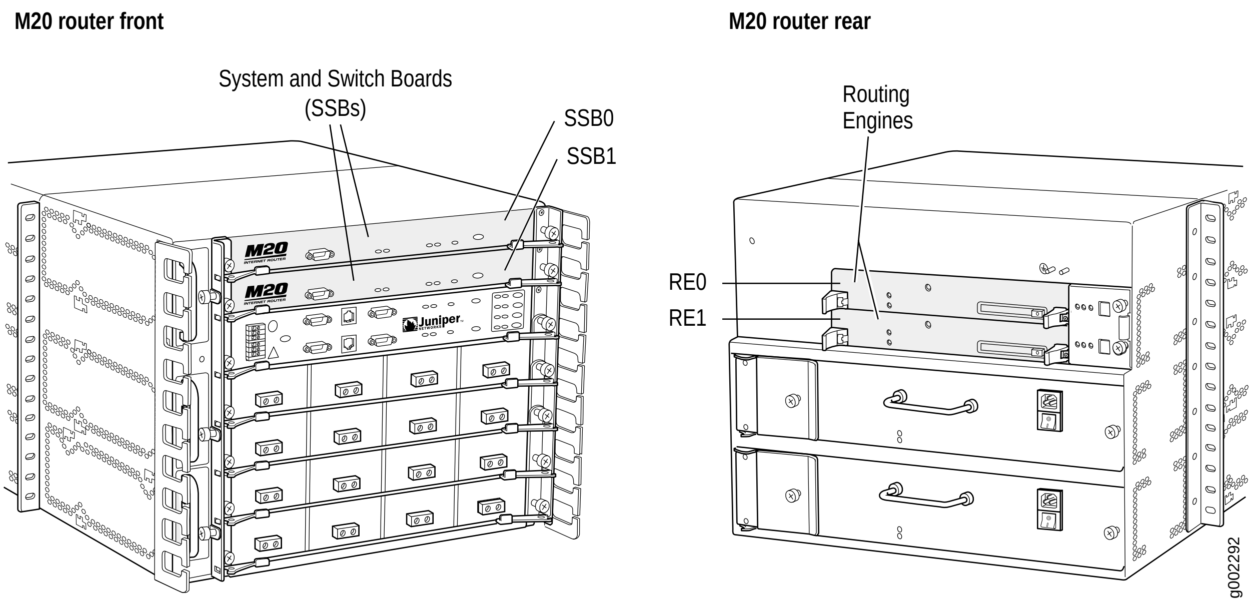
Figure 183: M10i Router Redundant Routing Engines and HCMs



M20 Router Redundant Routing Engines and SSBs

The M20 router can have one or two Routing Engines. The System and Switch Boards (SSBs) communicate with the Routing Engines. (See Figure 184 and "Redundancy Connection for an M20 Router" on page 471.)

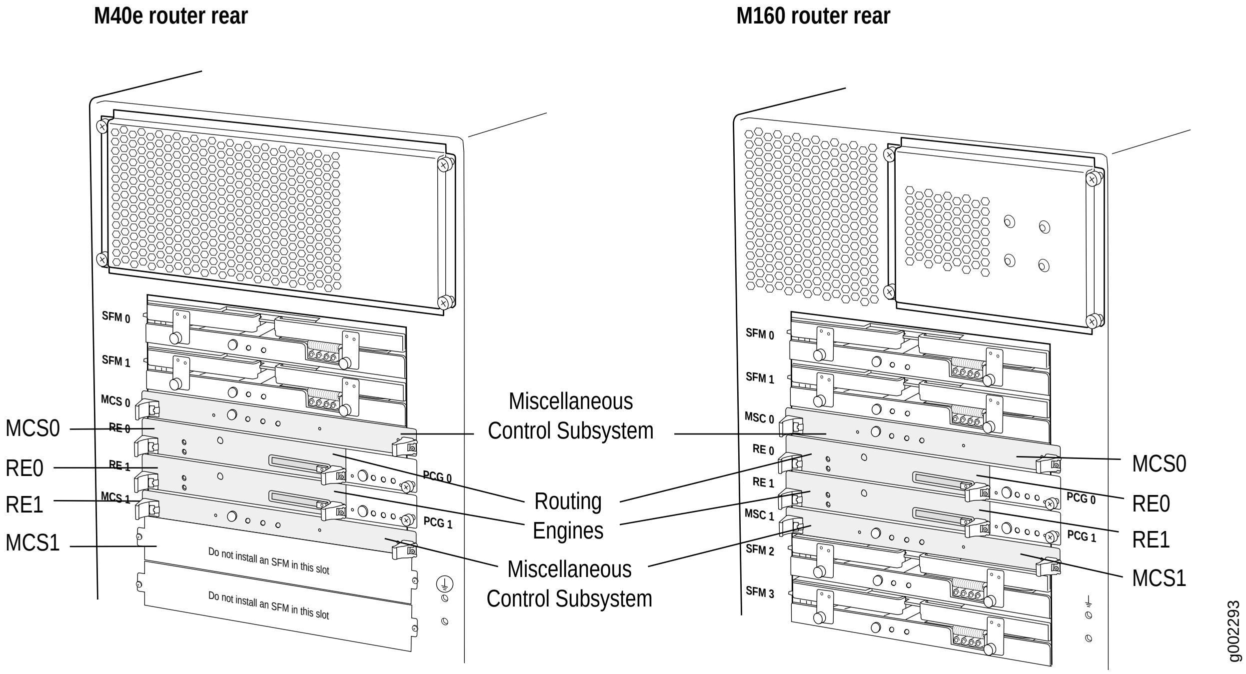
Figure 184: M20 Router Redundant Routing Engines and SSBs



M40e and M160 Router Redundant Host Modules

On M40e and M160 routers, the host module consists of a paired Routing Engine and MCS. One pair functions as master, while the other stands by as a backup should the master Routing Engine fail. (See Figure 185 and “Redundancy Connection for an M40e or M160 Router” on page 472.)

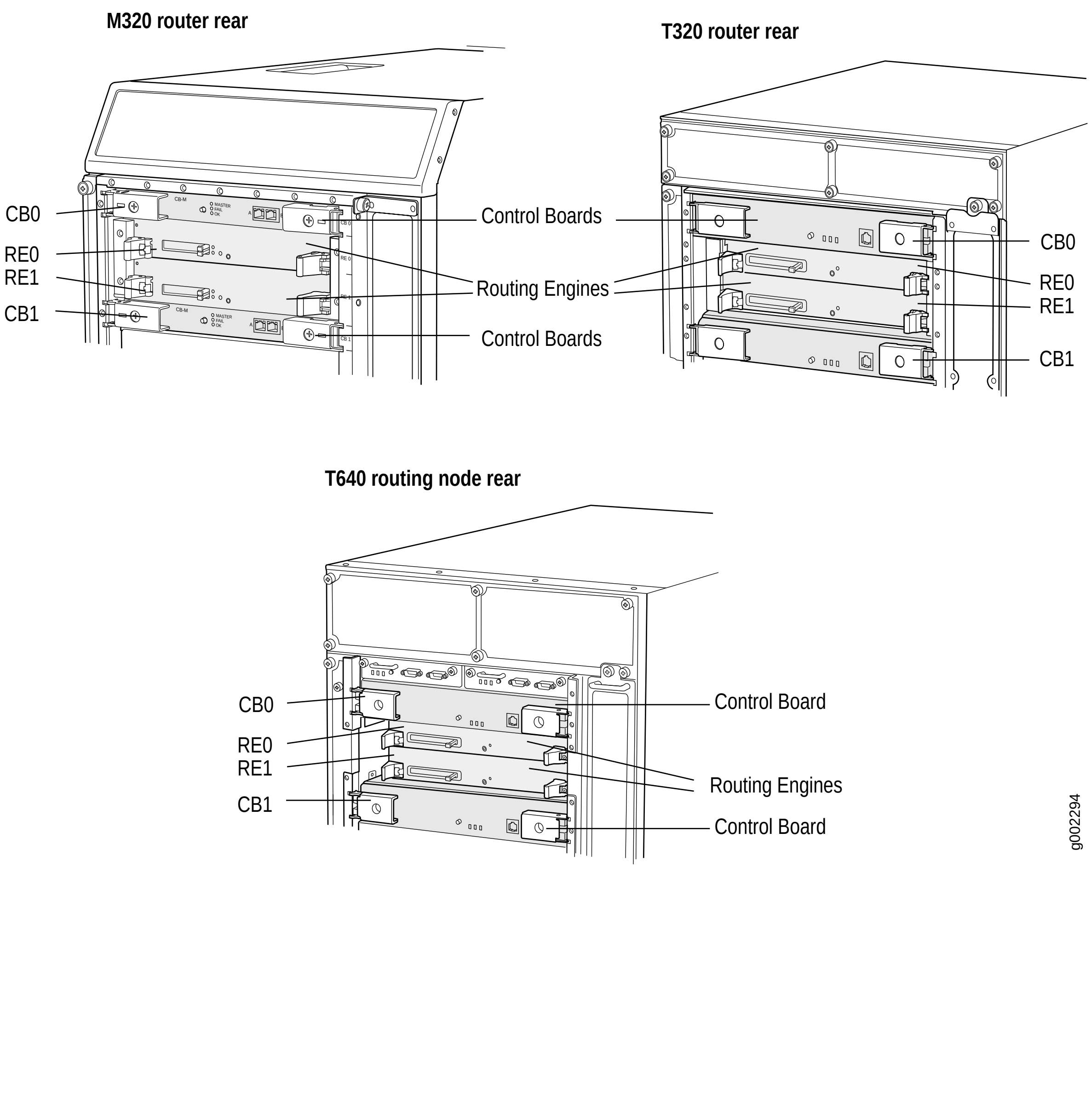
Figure 185: M40e and M160 Router Redundant Host Modules



M320 Router, T320 Router, and T640 Routing Node Redundant Host Subsystems

On the M320 router, T320 router, and the T640 routing node, the host subsystem consists of a Routing Engine and Control Board functioning as a unit. Two host subsystems can be installed in each routing platform. One pair functions as master, while the other stands by as backup should the master Routing Engine fail. (See Figure 186, "Redundancy Connection for an M320 Router" on page 473, and "Redundancy Connection for a T320 Router and T640 Routing Node" on page 474.)

Figure 186: M320 Router, T320 Router, and T640 Routing Node Redundant Host Subsystems



Routing Engine, Host Module, and Host Subsystem Redundancy Connections

It is important to understand how a redundant Routing Engine, redundant host module, or redundant subsystem communicates with its active counterparts to avoid severing the connection used for communication. Severing the connection can potentially trigger a failover protection.

For example, the M160 router active host module (the Routing Engine and the MCS) has the running configuration on it and communicates with the MCS, which in turn communicates with the Flexible PIC Concentrator (FPC) and the Switching and Forwarding Modules (SFMs). The host modules send keepalive messages to each other, checking the operating state. Each host module issues keepalive responses, letting the other host module know that it is up and operating. If keepalive responses are not returned to the standby host module (response times will vary depending upon the time settings specified in the set chassis redundancy keepalive-time statement), the standby host module can become the active host module. (See “Redundancy Connection for an M40e or M160 Router” on page 472.)

You also can configure failover on the router to switch mastership if a critical process fails. If a critical process on the active host module terminates, the standby host module routing becomes the active host module. You can configure processes for which this should happen. For example, you can use the set interface-control failover other-routing-engine statement at the [edit system processes] hierarchy level to configure failover for the interface control daemon.

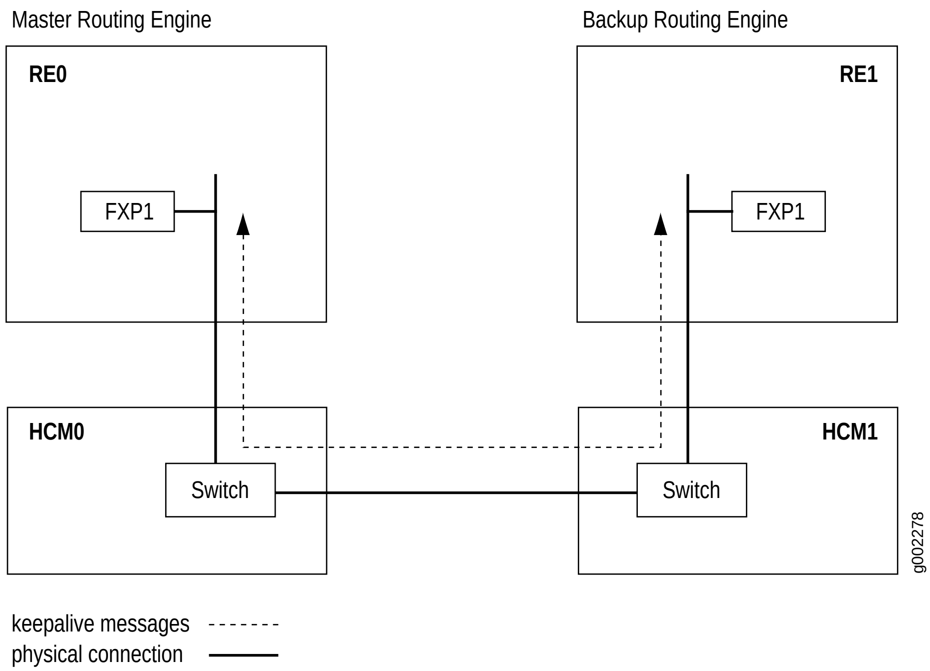
For information about setting keepalive parameters, see “Configuring the Backup Routing Engine to Assume Mastership on Failure of Keepalives” on page 486.

This section includes the following information:

Redundancy Connection for an M10i Router

Figure 187 shows the connection between the master and backup Routing Engines on an M10i router. Keepalive messages are sent between Routing Engines via the interconnected HCM switches. In this way, the master and the backup Routing Engines exchange state information.

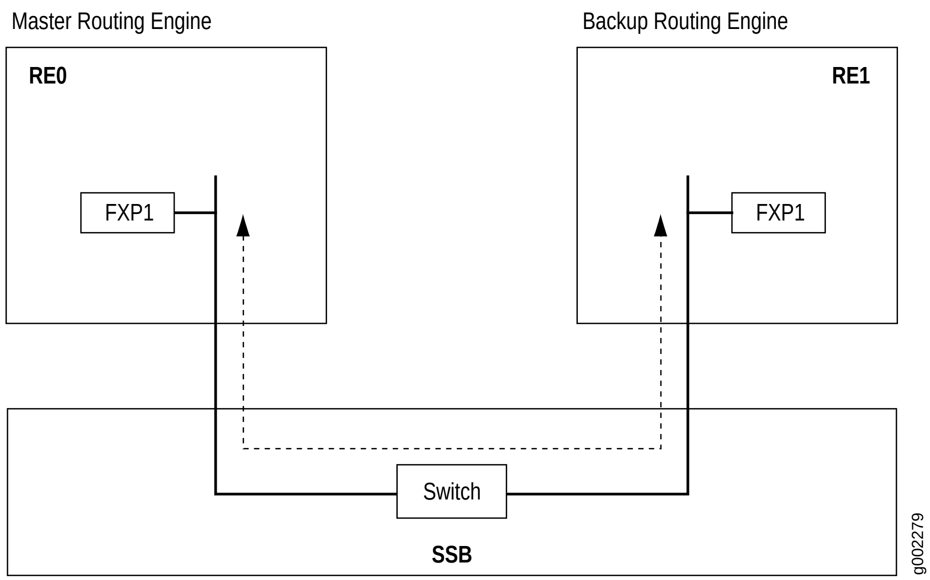
Figure 187: Redundancy Connection for an M10i Router



Redundancy Connection for an M20 Router

Figure 188 shows the connection between the master and backup Routing Engines on an M20 router. Keepalive messages are sent between the master and backup Routing Engine through the switch on the SSB. In this way, the master and the backup Routing Engines exchange state information.

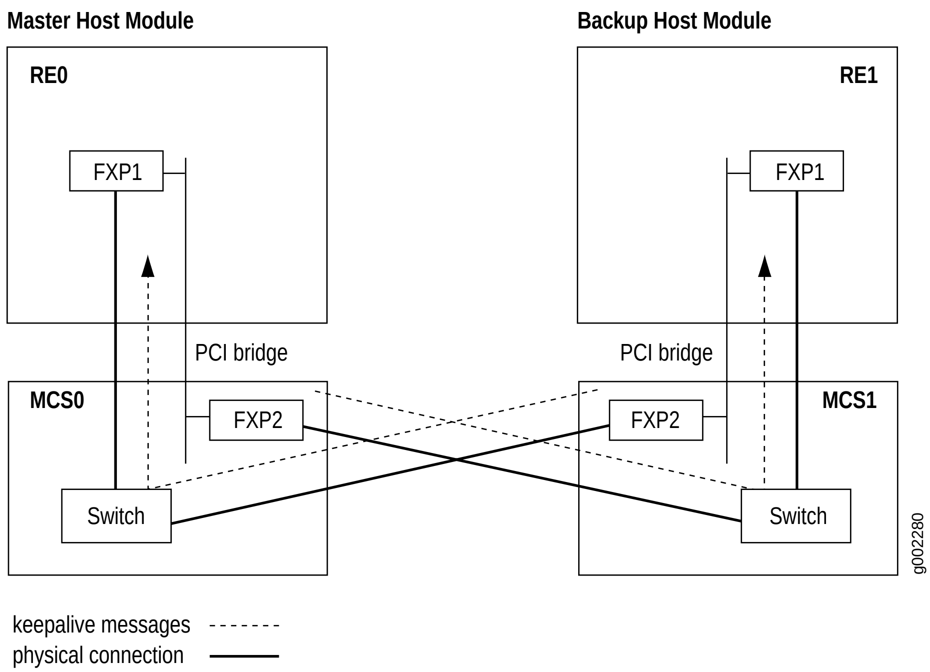
Figure 188: Redundancy Connection for an M20 Router



Redundancy Connection for an M40e or M160 Router

Figure 189 shows the connection between the master and backup host modules on an M40e or M160 router. Keepalive messages are sent from one Routing Engine to the other over the fpx2 interface found across the Peripheral Component Interconnect (PCI) bridge. The keepalive message is received by the other host module via the fpx1 interface. A keepalive response is sent back over the fpx2 interface to the other Routing Engine. In this way, the master and the backup host modules exchange state information.

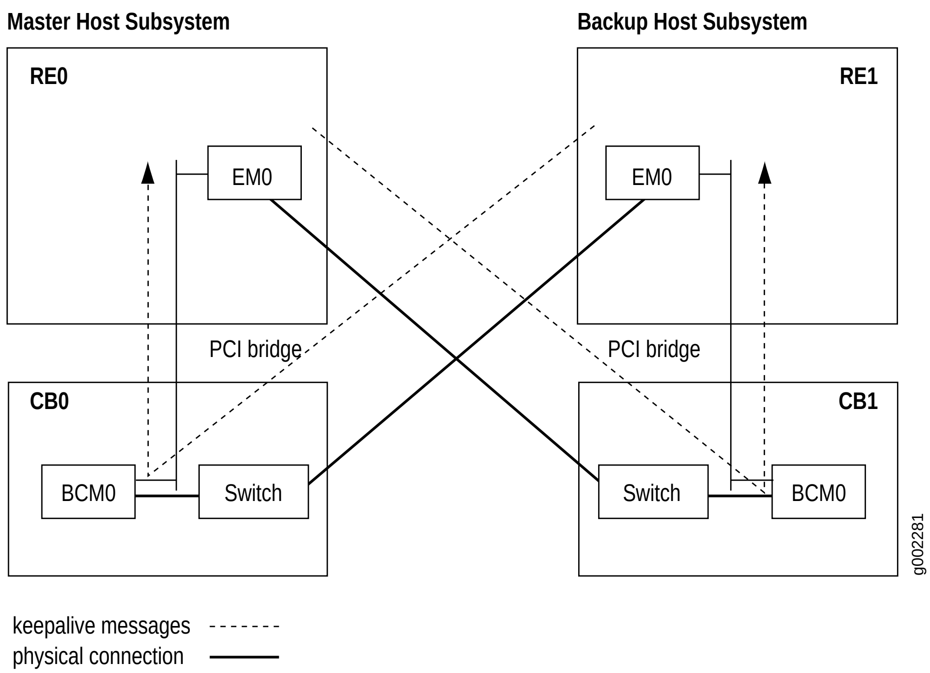
Figure 189: Redundancy Connection for an M40e or M160 Router



Redundancy Connection for an M320 Router

Figure 190 shows the connection between the master and backup host subsystems on an M320 router. Keepalive messages are sent from the Routing Engine over the em0 interface. The keepalive message is forwarded to the other host subsystem via the bcm0 interface on the Control Board. A keepalive response is sent back over the em0 interface to the other Routing Engine. In this way, the master and the backup host subsystems exchange state information.

Figure 190: Redundancy Connection for an M320 Router



Redundancy Connection for a T320 Router and T640 Routing Node

Figure 191 shows the connection between the master and backup host modules on a T320 router or a T640 routing node with a Routing Engine 600 (RE-600). Keepalive messages are sent from one Routing Engine to the other over the fxp2 interface found on the Routing Engine. The keepalive message is received by the other host module via the fpx1 interface. A keepalive message is sent back over the fxp2 interface of the other Routing Engine. In this way, the master and the backup host subsystems exchange state information.

Figure 191: Redundancy Connection for a T320 Router or T640 Routing Node (RE-600)

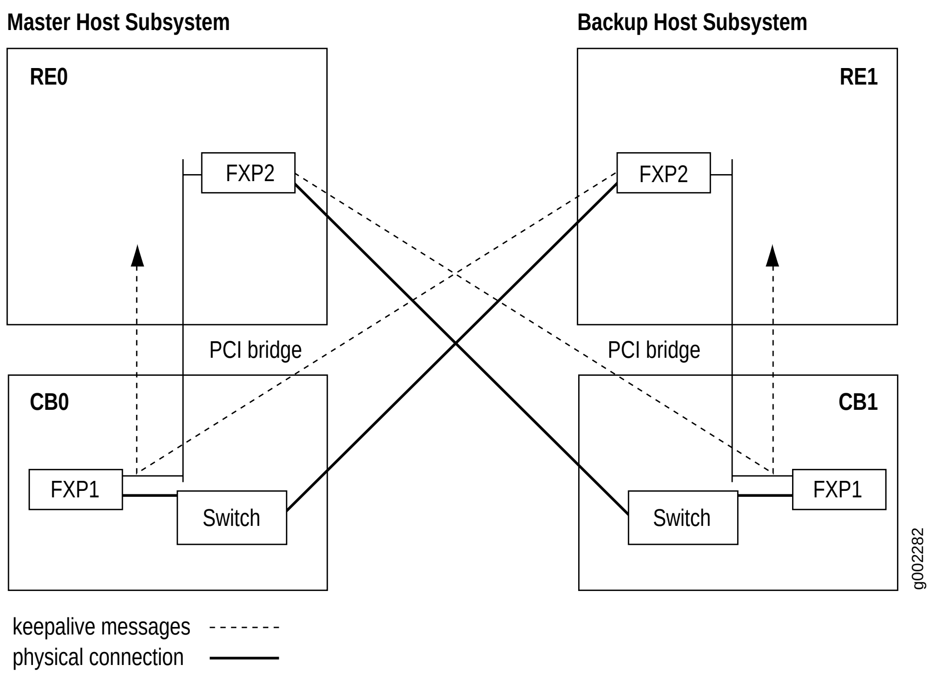
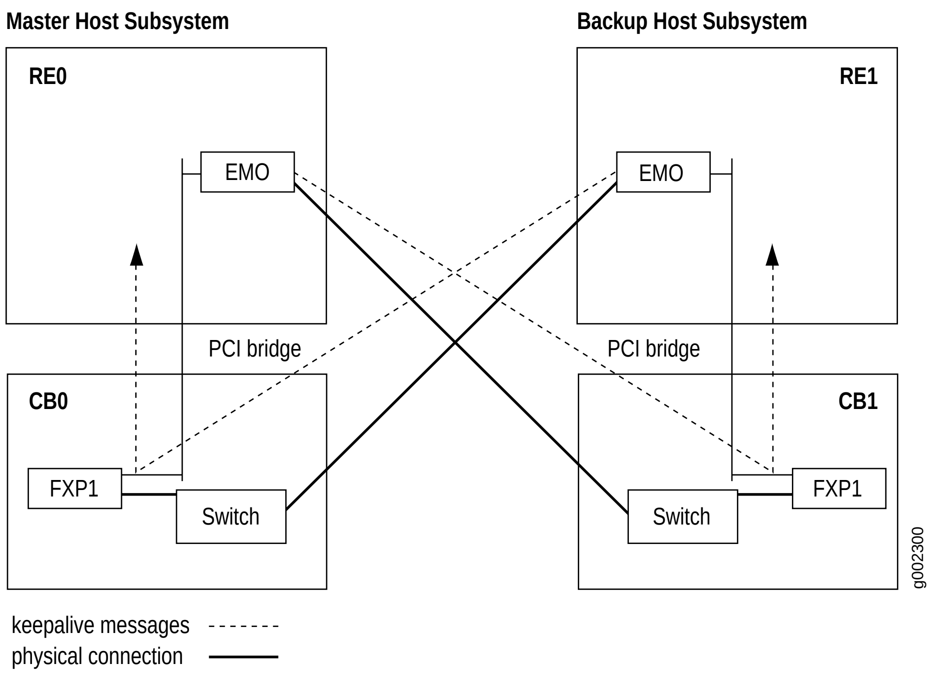


Figure 192 shows the connection between the master and backup host modules on a T320 router or a T640 routing node with a Routing Engine 1600 (RE-1600).

Figure 192: Redundancy Connection for a T320 Router or T640 Routing Node (RE-1600)



Determining Which Routing Engine You Are Logged In To

Steps To Take

1. Display Routing Engine Status on page 476
2. Display the Router Hardware on page 477

Step 1: Display Routing Engine Status

Action To determine which Routing Engine you are logged in to, use the following CLI command:

```
user@host> show chassis routing-engine
```

Sample Output

```
user@host> show chassis routing-engine

Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                    29 degrees C / 84 degrees F
    DRAM                           2048 MB
    Memory utilization             11 percent
    CPU utilization:
      User                         0 percent
      Background                   0 percent
      Kernel                       2 percent
      Interrupt                    0 percent
      Idle                         98 percent
    Model                          RE-3.0
    Serial ID                      P10865701859
    Start time                     2004-04-15 18:45:12 UTC
    Uptime                         6 days, 3 hours, 56 minutes, 8 seconds
Routing Engine status:
  Slot 1:
    Current state                  Backup
    Election priority              Backup (default)
    Temperature                    26 degrees C / 78 degrees F
    DRAM                           2048 MB
    Memory utilization             10 percent
    CPU utilization:
[...Output truncated...]
```

What It Means The output from the show chassis hardware command indicates that you are logged in to the master Routing Engine because this command can only be used on the master Routing Engine.

If you are not logged in to the master Routing Engine, you will see the following command output:

```
user@host> show chassis hardware

error: Aborted! This command can only be used on the master routing engine.
```

Step 2: Display the Router Hardware

Action To determine which Routing Engine you are logged in to, use the following JUNOS software command-line interface (CLI) command:

```
user@host> show chassis hardware
```

```
Hardware inventory:
Item          Version  Part number  Serial number     Description
Chassis                      65565         M320
Midplane        REV 05   710-009120   RB0662          M320 Midplane
FPM GBUS        REV 04   710-005928   HV7564          M320 Board
FPM Display     REV 05   710-009351   HY0996          M320 FPM Display
CIP           REV 04   710-005926   HV2440          M320 CIP
PEM 0           Rev 03   740-009148   QD17663         DC Power Entry Module
PEM 1           Rev 03   740-009148   QD17664         DC Power Entry Module
PEM 2           Rev 03   740-009148   QD17662         DC Power Entry Module
PEM 3           Rev 03   740-009148   QD16006         DC Power Entry Module
Routing Engine 0 REV 05   740-008883   P11123900322     RE-4.0
Routing Engine 1 REV 05   740-008883   P11123900311     RE-4.0
CB 0            REV 07   710-009115   HW8716          M320 Control Board
CB 1            REV 07   710-009115   HW8693          M320 Control Board
[...Output truncated...]
```

What it Means The output from the show chassis hardware command indicates that you are logged in to the master Routing Engine because this command can only be used on the master Routing Engine.

If you are not logged in to the master Routing Engine, you will see the following command output:

```
user@host> show chassis hardware

error: Aborted! This command can only be used on the master routing engine.
```

Determining Routing Engine Mastership

Steps To Take To determine Routing Engine mastership, follow these steps:

1. Determine the Routing Engine Mastership By Checking Status on page 478
2. Determine Routing Engine Mastership By Checking the LEDs on page 478
3. Log In To Backup Routing Engine If graceful-switchover is Configured on page 479

Step 1: Determine the Routing Engine Mastership By Checking Status

Action To determine Routing Engine mastership, use the following CLI command:

user@host> **show chassis routing-engine**

Sample Output

```
user@host> show chassis routing-engine

Routing Engine status:
  Slot 0:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 29 degrees C / 84 degrees F
    DRAM                      2048 MB
    Memory utilization          11 percent
    CPU utilization:
      User                      0 percent
      Background                0 percent
      Kernel                    2 percent
      Interrupt                 0 percent
      Idle                     98 percent
    Model                       RE-3.0
    Serial ID                   P10865701859
    Start time                  2004-04-15 18:45:12 UTC
    Uptime                      6 days, 3 hours, 56 minutes, 8 seconds
Routing Engine status:
  Slot 1:
    Current state                  Backup
    Election priority              Backup (default)
    Temperature                 26 degrees C / 78 degrees F
    DRAM                      2048 MB
    Memory utilization          10 percent
    CPU utilization:
[...Output truncated...]
```

What It Means The command output displays which Routing Engine is master (the one in Slot 0 RE0) and which is backup (the one in Slot 1 RE1) plus other hardware and operational status information.

Step 2: Determine Routing Engine Mastership By Checking the LEDs

Action Physically check the LEDs on either the craft interface or the Routing Engine (depending on which chassis the Routing Engine is installed). The Routing Engine that displays an illuminated Master LED is the master Routing Engine. For the location and interpretation of LEDs, see “Monitoring the Routing Engine Status” on page 136.

Step 3: Log In To Backup Routing Engine If graceful-switchover is Configured

If graceful-switchover is configured, the CLI command prompt will look as follows:

```
{backup}
user@host-re0>

{master}
user@host-re1>
```

With RE1 as master and RE0 as backup.

Action If you are logged in to the master Routing Engine, log in to the backup Routing Engine by using the following CLI command:

user@host> **request routing-engine login other-routing-engine**

Sample Output

```
user@host> request routing-engine login other-routing-engine
ıPassword: ########
{backup}
user@host-re0>
```

What It Means You are now logged in to the backup Routing Engine in slot RE0.

Manually Configuring Master and Backup Routing Engines

For routers with two Routing Engines, you can configure which Routing Engine is the master and which is the backup. By default, the Routing Engine in slot 0 is the master (RE0) and the one in slot 1 is the backup (RE1).

To modify the default configuration, include the routing-engine statement at the [edit chassis redundancy] hierarchy level:

[edit chassis redundancy]
routing-engine *slot-number* (master **|** backup **|** disabled);

slot-number can be 0 or 1. To configure the Routing Engine to be the master, specify the master option. To configure it to be the backup, specify the backup option. To switch between the master and the backup Routing Engines, you must modify the configuration and then activate it by issuing the commit command.

The running state of a Routing Engine (master, backup, or disabled) is determined by mastership election upon system boot.

Master—If a Routing Engine is configured as master, it has full functionality. It receives and transmits routing information, builds and maintains routing tables, communicates with interfaces and Packet Forwarding Engine components, and has full control over the chassis. Once a Routing Engine becomes master, it resets the switch plane (SSB, SCB, and SFM) and downloads its current version of the microkernel to the Packet Forwarding Engine components, guaranteeing software compatibility.

Backup—If a Routing Engine is configured to be the backup, it does not maintain routing tables or communicate with Packet Forwarding Engine or chassis components. However, it runs through its memory check and boot sequence to the point of displaying a login prompt. A backup Routing Engine supports full management access through the Ethernet, console, and auxiliary ports, and can communicate with the master Routing Engine. Additionally, a backup Routing Engine responds to the Routing Engine request chassis routing-engine master switch command. The backup Routing Engine maintains a connection with the master Routing Engine and monitors the master Routing Engine. If the connection is broken, you can switch mastership by entering the switchover command. If the master Routing Engine is hot-swapped out of the system, the backup takes over control of the system as the new master Routing Engine. Once a Routing Engine becomes master, it resets the switch plane and downloads its own version of the microkernel to the Packet Forwarding Engine components.

Disabled—A disabled Routing Engine has progressed through its memory check and boot sequence to the point of displaying a login prompt (similar to backup state) but does not respond to a request chassis routing-engine master switch command. A Routing Engine in disabled state supports full management access through the Ethernet, console, and auxiliary ports, and can communicate with the master Routing Engine. A disabled Routing Engine does not participate in a mastership election. To move from disabled state to backup state, the Routing Engine must be reconfigured to be the backup Routing Engine.

Action To configure RE1 to be the default master, issue the following CLI command in configuration mode at the [edit] hierarchy level:

For slot 0:

[edit]
user@host# **set chassis redundancy routing-engine 1 master**

[edit]
user@host# **commit**

For slot 1:

[edit]
user@host# **set chassis redundancy routing-engine 0 backup**

[edit]
user@host# **commit**

Action To view the Routing Engine mastership/backup status, use the following CLI command in operational mode:

user@host> **show chassis routing-engine**

Sample Output

```
user@host> show chassis routing-engine
Routing Engine status:
  Slot 0:
    Current state                  Backup
    Election priority              Backup (default)
    Temperature                 26 degrees C / 78 degrees F
    DRAM                      2048 MB
    Memory utilization          12 percent
    CPU utilization:
      User                      0 percent
      Background                0 percent
      Kernel                    1 percent
      Interrupt                 0 percent
      Idle                     99 percent
    Serial ID                   210929000142
    Start time                  2004-05-12 13:14:30 PDT
    Uptime                      5 days, 22 hours, 7 minutes, 9 seconds
    Load averages:                1 minute   5 minute  15 minute
                               0.07       0.02       0.00
Routing Engine status:
  Slot 1:
    Current state                  Master
    Election priority              Master (default)
    Temperature                 27 degrees C / 80 degrees F
    DRAM                      2048 MB
    Memory utilization          13 percent
    CPU utilization:
      User                      0 percent
      Background                0 percent
      Kernel                    0 percent
      Interrupt                 0 percent
      Idle                    100 percent
    Serial ID                   210929000143
    Start time                  2004-04-05 17:08:41 PDT
    Uptime                      42 days, 18 hours, 12 minutes, 45 seconds
```

What It Means

Each Routing Engine only checks its own configuration. Therefore, you must configure the redundancy settings on both Routing Engines correctly for the system to operate properly.

If both Routing Engines are configured as master, whichever Routing Engine comes up first will be the master. When the second Routing Engine comes up, it will try to assume mastership. However, the current master Routing Engine will reject this request, and the second Routing Engine will become the backup.

If both Routing Engines are configured as backup and come up after bootup, neither Routing Engine becomes master. The only way for either to become master is if one of the host module components (such as the Routing Engine) is physically removed, or if a Routing Engine has failover on-loss-of-keepalives configured and the connection between Routing Engines is interrupted for a period of time. The resulting timeout due to a loss of keepalives will force one of the Routing Engines to become the master. See "Configuring the Backup Routing Engine to Assume Mastership on Failure of Keepalives" on page 486 for more information.

Manually Switching Routing Engine Mastership

Action To manually switch the Routing Engine mastership, use one of the following CLI commands.

From the backup Routing Engine, request the backup Routing Engine to acquire mastership:

user@host> **request chassis routing-engine master acquire**

```
user@host> request chassis routing-engine master acquire
warning: Traffic will be interrupted while the PFE is re-initialized
Attempt to become the master routing engine ? [yes,no] (no)

Resolving mastership...
Complete. The local routing engine becomes the master.
```

From the master Routing Engine, request the backup Routing Engine to acquire mastership:

user@host> **request chassis routing-engine master release**

```
user@host> request chassis routing-engine master release

Traffic will be interrupted while the PFE is re-initialized
Request the other routing engine become master ? [yes,no] (no)

Resolving mastership...
Complete. The other routing engine becomes the master.
```

Switch mastership from either the backup or master Routing Engine:

user@host> **request chassis routing-engine master switch**

If graceful-switchover is not configured, the command output looks as follows:

```
user@host> request chassis routing-engine master switch
warning: Traffic will be interrupted while the PFE is re-initialized
Toggle mastership between routing engines ? [yes,no] (no) yes

Resolving mastership...
Complete. The local routing engine becomes the master.

user@host> request chassis routing-engine master switch
```

If graceful-switchover is configured the command output looks as follows:

```
Toggle mastership between routing engines ? [yes,no] (no) yes

Resolving mastership...
Complete. The other routing engine becomes the master.
```

What It Means When you enter the request chassis routing-engine master acquire command on the backup Routing Engine, you see the following:

```
warning: Traffic will be interrupted while the PFE is re-initialized
Attempt to become the master routing engine ? [yes,no] (no).
```

The master Routing Engine gives up control of the system bus and goes into the backup state. The backup Routing Engine becomes master and restarts the Packet Forwarding Engine. You can then diagnose the original master Routing Engine for problems or prepare it for upgrade or reconfiguration. When switchover occurs, the backup Routing Engine does not run through its full boot cycle (only the packet forwarding components run through a full boot cycle).

When you enter the request chassis routing-engine master release command on the master Routing Engine, the system passes mastership to the backup Routing Engine. The master Routing Engine gives up control of the system bus and goes into the backup state. The backup Routing Engine becomes master and restarts the Packet Forwarding Engine. You can then diagnose the original master Routing Engine for problems or prepare it for upgrade or reconfiguration.

In all cases, once the switchover occurs, the new master Routing Engine reestablishes routing adjacencies, populates the routing table, and transfers forwarding table information to the Packet Forwarding Engine. When Routing Engine mastership changes, the Packet Forwarding Engine components are rebooted to reestablish communication links and download the microkernel to each component. When this occurs, forwarding is interrupted and packet buffers are flushed.

Determining Why Mastership Switched

Mastership can switch between the master Routing Engine and the backup Routing Engine for the following reasons:

Hardware problems.

The master Routing Engine is pulled.

Software issues, such as a Routing Engine kernel crash.

Action View the log file /var/log/mastership for redundancy logging. This file contains hardware and software transitions to help debug auto-redundancy issues.

user@host> **show log mastership**

Table 107 lists the event codes that can be displayed in the mastership log.

Table 107: Logging Events

Event Code	Description
E_NULL = 0	The event is a null event.
E_CFG_M	The Routing Engine is configured as master.
E_CFG_B	The Routing Engine is configured as backup.
E_CFG_D	The Routing Engine is configured as disabled.
E_MAXTRY	The maximum number of tries to acquire or release mastership was exceeded.
E_REQ_C	A claim mastership request was sent.
E_ACK_C	A claim mastership acknowledgement was received.

Event Code	Description
E_NAK_C	A claim mastership request was not acknowledged.
E_REQ_Y	Confirmation of mastership is requested.
E_ACK_Y	Mastership is acknowledged.
E_NAK_Y	Mastership is not acknowledged.
E_REQ_G	A giveup mastership request was sent by a Routing Engine.
E_ACK_G	The Routing Engine acknowledges giveup of mastership.
E_CMD_A	The command request chassis routing-engine master acquire was issued from the backup Routing Engine.
E_CMD_F	Force switchover command was issued.
E_CMD_R	The command request chassis routing-engine master release was issued from the master Routing Engine.
E_CMD_S	The command request chassis routing-engine master switch was issued from a Routing Engine.
E_NO_ORE	No other Routing Engine is detected.
E_TMOUT	A request timed out.
E_NO_IPC	Routing Engine connection was lost.
E_ORE_M	Other Routing Engine state was changed to master.
E_ORE_B	Other Routing Engine state was changed to backup.
E_ORE_D	Other Routing Engine state was changed to disabled.

Sample Output

```
user@host> show log mastership
Jan 12 21:50:05  clear-log[865]: logfile cleared
Jan 12 21:50:18 failed to receive keepalives from other RE for the last 60 sec Jan 12 21:50:23 failed to send RE info/keepalive: errno=22, total=6 in the last 20 sec
Jan 12 21:50:23 failed to send RE info/keepalive: errno=22, total=6 in the last 20 sec
Jan 12 21:50:34 event = E_CMD_R, state = master, param = 0x0 Jan 12 21:50:34 send "you are the master" request Jan 12 21:50:34 Failed to send RE mastership cmd. err = 65 Jan 12 21:50:34 Currentstate: master  NextState:giveup
        reason_code: 1
Jan 12 21:50:34 timestamp: Wed Jan 12 21:50:34 2000
Jan 12 21:50:34 new state = giveup
Jan 12 21:50:36 event = E_TMOUT, state = giveup, param = 0x0 Jan 12 21:50:36 send "you are the master" request
Jan 12 21:50:36 Failed to send RE mastership cmd. err = 65 Jan 12 21:50:36 Currentstate: giveup  NextState:giveup
        reason_code: 1
Jan 12 21:50:36 new state = giveup
Jan 12 21:50:38 event = E_TMOUT, state = giveup, param = 0x0
Jan 12 21:50:38 send "you are the master" request
Jan 12 21:50:38 Failed to send RE mastership cmd. err = 65
Jan 12 21:50:38 Currentstate: giveup  NextState:giveup
        reason_code: 1
Jan 12 21:50:38 new state = giveup
Jan 12 21:50:40 failed to receive keepalives from other RE for the last 80 sec Jan 12 21:50:41 event = E_TMOUT, state = giveup, param = 0x0
Jan 12 21:50:41 send "you are the master" request
Jan 12 21:50:41 Failed to send RE mastership cmd. err = 65
Jan 12 21:50:41 Currentstate: giveup  NextState:giveup
        reason_code: 1
Jan 12 21:50:41 new state = giveup
Jan 12 21:50:43 event = E_TMOUT, state = giveup, param = 0x0
Jan 12 21:50:43 send "you are the master" request
Jan 12 21:50:43 Failed to send RE mastership cmd. err = 65
Jan 12 21:50:43 Currentstate: giveup  NextState:giveup
        reason_code: 1
```

```
Jan 12 21:50:43 new state = giveup
Jan 12 21:50:46 failed to send RE info/keepalive: errno=35, total=7 in the last 20 sec
Jan 12 21:50:46 failed to send RE info/keepalive: errno=35, total=7 in the last 20 sec
Jan 12 21:50:46 event = E_TMOUT, state = giveup, param = 0x0
Jan 12 21:50:46 send "you are the master" request
Jan 12 21:50:46 Failed to send RE mastership cmd. err = 65
Jan 12 21:50:46 Currentstate: giveup  NextState:giveup
                reason_code: 1
Jan 12 21:50:46 new state = giveup
Jan 12 21:50:48 event = E_TMOUT, state = giveup, param = 0x0
Jan 12 21:50:48 send "you are the master" request
Jan 12 21:50:48 Failed to send RE mastership cmd. err = 65
Jan 12 21:50:48 Currentstate: giveup  NextState:giveup
                reason_code: 1
Jan 12 21:50:48 new state = giveup
Jan 12 21:50:50 event = E_TMOUT, state = giveup, param = 0x0
Jan 12 21:50:50 send "you are the master" request
Jan 12 21:50:50 Failed to send RE mastership cmd. err = 65
Jan 12 21:50:50 Currentstate: giveup  NextState:giveup
                reason_code: 1
Jan 12 21:50:50 new state = giveup
Jan 12 21:50:53 event = E_MAXTRY, state = giveup, param = 0x0
Jan 12 21:50:53 Currentstate: giveup  NextState:master
                reason_code: 1
Jan 12 21:50:53 timestamp: Wed
Jan 12 21:50:53 2000
Jan 12 21:50:53 new state = master
Jan 12 21:51:01 failed to receive keepalives from other RE for the last 100 sec Jan 12 21:51:06 failed to send RE
info/keepalive: errno=65, total=7 in the last 20 sec
Jan 12 21:51:06 failed to send RE info/keepalive: errno=65, total=7 in the last 20 sec
Jan 12 21:51:21 failed to receive keepalives from other RE for the last 120 sec Jan 12 21:51:26 failed to send RE
info/keepalive: errno=22, total=6 in the last 20 sec
Jan 12 21:51:26 failed to send RE info/keepalive: errno=22, total=6 in the last 20 sec
```

What It Means The beginning of the log shows that keepalives are not being responded to and the state of the Routing Engine changed from master to giveup after the request chassis routing-engine master release command was issued. However, the other Routing Engine is not taking over mastership because it is unreachable. Eventually a timeout (E_TMOUT) occurs until the Routing Engine reaches the maximum number of attempts permitted (E_MAXTRY). The output then shows the Routing Engine state changing from giveup back to master.

The output doesn’t indicate why the mastership switchover did not work. However, it is clear that the backup Routing Engine is unreachable.

Configuring the Backup Routing Engine to Assume Mastership on Failure of Keepalives

Action Configure the backup Routing Engine to automatically assume mastership if it detects a loss of keepalive responses with the set chassis routing-engine statement at the [edit] hierarchy level:

```
[edit]
user@host# set chassis redundancy failover on-loss-of-keepalives
```

NOTE: By default, a backup Routing Engine does not assume mastership when a loss of keepalive responses occurs.

Sample Output

```
[edit]
user@host# set chassis redundancy failover on-loss-of-keepalives

[edit]
user@host# set chassis redundancy keepalive-time 300

[edit]
user@host# commit
    commit complete
```

What it Means The results of issuing this command on the backup Routing Engine are as follows:

Every 20 seconds of keepalive loss, a message is added to the /var/log/mastership file.

After keepalive-time passes, the backup Routing Engine attempts to claim mastership.

When the backup Routing Engine claims mastership, it continues to be master even after the other Routing Engine configured as master has successfully resumed operation. Therefore, if the backup Routing Engine claims mastership, you must manually switch mastership.

The default time before failover will occur is set to 300 seconds (5 minutes). You can change the default keepalive time period with the set chassis redundancy keepalive-time *time-in-seconds* command (the range for keepalive-time is from 2 to 10,000 seconds).

Keepalive messages are sent every second.

Avoiding Redundancy Problems

Problems with reliable redundancy are more often caused by poor management of software rather than by hardware failure. The following operating guidelines reduce the likelihood of significant downtime due to Routing Engine redundancy conflicts.

Steps To Take

Step 1: Operate the Same Type of Routing Engine and JUNOS Software

The active and standby Routing Engines must be the same type of Routing Engine and must operate the same version of JUNOS software; otherwise, anomalies in operation can occur.

Step 2: Use the Groups Configuration

Action Apply a single configuration file to both Routing Engines using the groups *group-name* statement at the [edit] hierarchy level:

```
[edit]
user@host# set groups group-name
```

Where *group-name* is the name of the configuration group. To configure multiple groups, specify more than one *group-name*. On routers that support multiple Routing Engines, you can also specify two special group names:

re0—Configuration statements that are applied to the Routing Engine in slot 0.

re1—Configuration statements that are applied to the Routing Engine in slot 1.

The configuration specified in group re0 is only applied if the current Routing Engine is in slot 0; likewise, the configuration specified in group re1 is only applied if the current Routing Engine is in slot 1. Therefore, both Routing Engines can use the same configuration file, each using only the configuration statements that apply to it. Each re0 or re1 group contains at a minimum the configuration for the hostname and the management interface (fxp0). If each Routing Engine uses a different management interface, the group also should contain the configuration for the backup router and static routes.

To view the existing groups configuration, use the following CLI command in configuration mode:

```
[edit]
user@host# groups
user@host# show
```

Sample Output

```
[edit groups]
user@host# show
re0 {
    system {
        host-name foo-re0;
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 10.0.0.1/24;
                }
            }
        }
    }
}
re1 {
    system {
        host-name foo-re1;
    }
    interfaces {
        fxp0 {
            unit 0 {
                family inet {
                    address 10.0.0.2/24;
                }
            }
        }
    }
}
```

What it Means

Use the already-existing groups statement, and use re0 and re1 as keyword group names. Each Routing Engine applies the slot-specific group configuration information to its configurations.

In the main configuration body, add the rest of the configuration that will be the same on both Routing Engines. Do not include the configuration statements that you made in the group configurations (such as configurations for fxp0). If you configure items in the body of the statement that also exist in the groups statement, the configuration in the body takes precedence—the configuration from the group statement will not be inherited.

Action

Display the groups that were applied using the following configuration mode CLI command:

```
[edit]
user@host# show apply-groups
```

Sample Output

```
user@host# show apply-groups
apply-groups [ re0 re1 ];
```

Step 3: Synchronize Configurations

Action Synchronize configurations between two Routing Engines using the synchronize statement at the [edit] hierarchy level:

```
[edit]
user@host# commit synchronize
```

Sample Output

```
[edit]
root# commit synchronize
re1: configuration check succeeds
re0: configuration check succeeds
re1: commit complete
re0: commit complete
```

What it Means When this statement is selected, the configuration file is copied to the other Routing Engine, followed by a load override and a commit. No user intervention is required.

NOTE: Both Routing Engines must be running JUNOS software Release 5.1 or higher. Use the groups statement to ensure that differences in the configurations for RE0 and RE1 are applied.

Step 4: Copy a Configuration File from One Routing Engine to Another

Action You can copy a configuration file from one Routing Engine to another using the file copy command. The file is transferred through the internal Ethernet interface (FXP1 or FXP2, depending on the router):

```
user@host> file copy <source> <destination>
```

NOTE: Both Routing Engines must have jbase version 4.1 or higher loaded.

Sample Output Copy a file on RE0 to RE1:

```
user@re0> file copy /var/tmp/jinstall-6.0R3.3-domestic-signed.tgz re1:/var/tmp1/
```

Check the result on RE1:

```
user@re1> file list /var/tmp/
.pccardd=
jbundle-5.5R3.1-domestic.tgz*
jinstall-6.0R3.3-domestic-signed.tgz
sampled.pkts
```

What it Means The file jinstall-6.0R3.3-domestic-signed.tgz is copied from RE0 to RE1.

Step 5: Use the Proper Shutdown Process on a Backup Routing Engine

Action The request system halt command only shuts down the Routing Engine you are logged in to; the other Routing Engine is still running and may be performing file management or some other task that could create anomalies.

```
user@re0> request system halt
```

Sample Output

```
user@re0> request system halt
warning: This command will not halt the other routing-engine.
If planning to switch off power, use the both-routing-engines option.
Halt the system ? [yes,no] (no)

*** FINAL System shutdown message from root@utah ***
System going down IMMEDIATELY


shutdown: [pid 8669]
Shutdown NOW!
```

What It Means

This command only shuts down the Routing Engine you are logged in to. To shut down both Routing Engines, use the both-routing-engines option or log in to the other Routing Engine and perform the shutdown again.