



CHAPTER 6

RPR スーパーバイザ エンジンの冗長構成の設定

この章では、Route Processor Redundancy (RPR) を使用してスーパーバイザ エンジンの冗長構成を設定する方法について説明します。



(注)

- この章で使用しているコマンドの構文および使用方法の詳細については、次の URL で『*Catalyst Supervisor Engine 32 PISA Cisco IOS Command Reference, Release 12.2ZY*』を参照してください。
<http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2ZY/command/reference/cmdref.html>
- Stateful Switchover (SSO) を備えた Nonstop Forwarding (NSF) については、第 5 章「NSF with SSO スーパーバイザ エンジンの冗長構成の設定」を参照してください。

この章で説明する内容は、次のとおりです。

- 「RPR の概要」 (P.6-1)
- 「スーパーバイザ エンジンの冗長構成に関する注意事項および制約事項」 (P.6-3)
- 「スーパーバイザ エンジンの冗長構成の設定」 (P.6-5)
- 「FSU の実行」 (P.6-6)
- 「冗長スーパーバイザ エンジンへのファイルのコピー」 (P.6-8)

RPR の概要

ここでは、RPR を使用したスーパーバイザ エンジンの冗長構成の概要について説明します。

- 「スーパーバイザ エンジンの冗長構成の概要」 (P.6-2)
- 「RPR の動作」 (P.6-2)
- 「スーパーバイザ エンジンの設定の同期化」 (P.6-3)

スーパーバイザ エンジンの冗長構成の概要



(注)

冗長スーパーバイザ エンジンがスタンバイ モードの場合、冗長スーパーバイザ エンジンの 2 つのギガビットイーサネット インターフェイスは常にアクティブです。

Catalyst 6500 シリーズ スイッチは、プライマリ スーパーバイザ エンジンが故障した場合に冗長スーパーバイザ エンジンがテイクオーバーすることにより、耐障害性をサポートします。RPR は 2 分以上のスイッチオーバー時間をサポートします。

次のイベントが発生すると、スイッチオーバーが行われます。

- アクティブ スーパーバイザ エンジンでのハードウェア障害
- スーパーバイザ エンジン間のクロック同期損失
- 手動スイッチオーバー

RPR の動作

RPR は次の機能をサポートします。

- 自動スタートアップおよびアクティブ スーパーバイザ エンジンと冗長スーパーバイザ エンジン間の `bootvar` の同期化
- スーパーバイザ エンジンのアクティブ ステータスまたは冗長ステータスを検出および決定するハードウェア信号
- アクティブ スーパーバイザ エンジンから冗長スーパーバイザ エンジンへ、60 秒間隔でクロック同期化を実行
- 冗長スーパーバイザ エンジンは、起動してもすべてのサブシステムが稼動するわけではなく、アクティブ スーパーバイザ エンジンが故障した場合に、完全に動作可能になります。
- 故障した装置の代わりに動作可能なスーパーバイザ エンジンが、冗長スーパーバイザ エンジンになります。
- Fast Software Upgrade (FSU) のサポート ([「FSU の実行」\(P.6-6\)](#) を参照)。

スイッチの電源投入時に、2 つのスーパーバイザ エンジン間で RPR が稼動します。最初に起動したスーパーバイザ エンジンが RPR アクティブ スーパーバイザ エンジンになります。MSFC および PFC は完全に動作可能になります。冗長スーパーバイザ エンジン上の PISA および PFC3B はリセットされますが、動作可能にはなりません。

スイッチオーバーが行われると、冗長スーパーバイザ エンジンが完全に動作可能になり、次の動作が行われます。

- すべてのスイッチ モジュールの電源が再びオンになります。
- PISA 上の残りのサブシステム (レイヤ 2 およびレイヤ 3 プロトコルを含む) が起動されます。
- Access Control List (ACL; アクセス制御リスト) がスーパーバイザ エンジンのハードウェアに再度プログラミングされます。



(注)

スイッチオーバー時には、一部のアドレス ステートが失われ、ダイナミックに再確認したあとで復元されるので、トラフィックが一時中断されます。

スーパーバイザ エンジンの設定の同期化



(注)

SNMP を通じて行われた設定変更は、冗長スーパーバイザ エンジンと同期化されません。SNMP を通じてスイッチを設定したあと、**running-config** ファイルをアクティブ スーパーバイザ エンジンの **startup-config** ファイルにコピーして、冗長スーパーバイザ エンジンの **startup-config** ファイルの同期化を引き起こします。

RPR モードの動作時には、2 つのスーパーバイザ エンジン間で **startup-config** ファイルおよび **config-register** コンフィギュレーションがデフォルトで同期化されます。スイッチオーバー時には、新しいアクティブ スーパーバイザ エンジンが現在の設定を使用します。

スーパーバイザ エンジンの冗長構成に関する注意事項および制約事項

ここでは、スーパーバイザ エンジンの冗長構成に関する注意事項および制約事項について説明します。

- 「冗長構成の注意事項および制約事項」(P.6-3)
- 「ハードウェア設定時の注意事項および制約事項」(P.6-4)
- 「コンフィギュレーション モードに関する制約事項」(P.6-4)

冗長構成の注意事項および制約事項

次の注意事項と制約事項は、RPR 冗長モードに適用されます。

- 冗長スーパーバイザ エンジンがスタンバイ モードの場合、冗長スーパーバイザ エンジンの 2 つのギガビット イーサネット インターフェイスは常にアクティブです。
- スーパーバイザ エンジンを冗長構成にしても、スーパーバイザ エンジンのミラーリングやロードバランスは行われません。スーパーバイザ エンジンのうちの 1 台だけがアクティブになります。
- SNMP を通じて行われた設定変更は、冗長スーパーバイザ エンジンと同期化されません。SNMP を通じてスイッチを設定したあと、**running-config** ファイルをアクティブ スーパーバイザ エンジンの **startup-config** ファイルにコピーして、冗長スーパーバイザ エンジンの **startup-config** ファイルの同期化を引き起こします。
- スーパーバイザ エンジンのスイッチオーバーは、障害のあるスーパーバイザ エンジンがコア ダンプを完了したあとに行われます。コア ダンプには最大で 15 分間かかります。スイッチオーバー時間を短縮するには、スーパーバイザ エンジンでコア ダンプをディセーブルにします。

ハードウェア設定時の注意事項および制約事項

冗長運用を行うには、次の注意事項および制約事項に従う必要があります。

- スーパーバイザ エンジンおよび PISA で実行する Cisco IOS は、スーパーバイザ エンジンおよび PISA ルータが同一である冗長構成をサポートします。スーパーバイザ エンジンおよび PISA ルータが異なる場合、片方が最初に起動されてアクティブになり、もう一方がリセット状態で保留されます。
- 各スーパーバイザ エンジンが単独でスイッチを稼働させるためのリソースを備えているスーパーバイザ エンジンのすべてのリソース（すべてのフラッシュ装置を含む）が重複している必要があります。
- スーパーバイザ エンジンごとに個別のコンソール接続を行ってください。コンソール ポートに Y 字ケーブルを接続しないでください。
- 両方のスーパーバイザ エンジン内のシステム イメージが同じである必要があります（「冗長スーパーバイザ エンジンへのファイルのコピー」(P.6-8) を参照）。



(注) 新たに取り付けられた冗長スーパーバイザ エンジン上で Catalyst オペレーティング システムがインストールされている場合は、アクティブなスーパーバイザ エンジンを取り外して、冗長スーパーバイザ エンジンだけが搭載されている状態でスイッチを起動します。最新のリリース ノートの手順に従って、Catalyst オペレーティング システムから冗長スーパーバイザ エンジンを変換してください。

- startup-config のコンフィギュレーション レジスタが自動起動用に設定されている必要があります（「ブート フィールドの変更」(P.3-24) を参照）。



(注) ネットワークからの起動はサポートされていません。

コンフィギュレーション モードに関する制約事項

スタートアップ同期プロセス中は、設定に関して次の制約事項が適用されます。

- スタートアップ（一括）同期中は、設定を変更できません。このプロセス中に設定を変更しようとすると、次のメッセージが生成されます。

```
Config mode locked out till standby initializes
```

- スーパーバイザ エンジンのスイッチオーバー時に設定を変更した場合、その変更内容は失われます。

スーパーバイザ エンジンの冗長構成の設定

ここでは、スーパーバイザ エンジンの冗長構成を設定する手順について説明します。

- 「冗長運用の設定」 (P.6-5)
- 「スーパーバイザ エンジンの設定の同期化」 (P.6-6)
- 「冗長ステータスの表示」 (P.6-6)

冗長運用の設定

冗長運用を設定する手順は、次のとおりです。

	コマンド	目的
ステップ 1	Router (config) # redundancy	冗長コンフィギュレーション モードを開始します。
ステップ 2	Router (config-red) # mode rpr	RPR モードを設定します。このコマンドを入力すると、冗長スーパーバイザ エンジンがリロードされ、RPR モードでの処理が開始されます。
ステップ 3	Router# show running-config	RPR モードがイネーブルになっていることを確認します。
ステップ 4	Router# show redundancy states	動作中の冗長モードを表示します。

次に、システムを RPR 用に設定して、冗長ステータスを表示する例を示します。

```

Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# redundancy
Router(config-red)# mode rpr
Router(config-red)# end
Router# show redundancy states
    my state = 13 -ACTIVE
    peer state = 1 -DISABLED
        Mode = Simplex
        Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Route Processor Redundancy
Redundancy Mode (Configured)  = Route Processor Redundancy
    Split Mode = Disabled
    Manual Swact = Disabled Reason: Simplex mode
    Communications = Down Reason: Simplex mode

    client count = 11
    client_notification_TMR = 30000 milliseconds
        keep_alive TMR = 4000 milliseconds
        keep_alive count = 0
        keep_alive threshold = 7
        RF debug mask = 0x0

Router#

```

スーパーバイザ エンジンの設定の同期化

通常動作時には、2つのスーパーバイザ エンジン間で `startup-config` および `config-register` 設定がデフォルトで同期化されます。スイッチオーバー時には、新しいアクティブ スーパーバイザ エンジンが現在の設定を使用します。



(注) デフォルトの `auto-sync` 設定は変更しないでください。

冗長ステータスの表示

冗長ステータスを表示するには、次の作業を行います。

コマンド	目的
Router# <code>show redundancy states</code>	冗長ステータスを表示します。

次に、冗長ステータスを表示する例を示します。

```
Router# show redundancy states
my state = 13 -ACTIVE
    peer state = 8 -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Route Processor Redundancy Plus
Redundancy Mode (Configured) = Route Processor Redundancy Plus
    Split Mode = Disabled
    Manual Swact = Enabled
    Communications = Up

    client count = 11
    client_notification_TMR = 30000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
        keep_alive threshold = 18
        RF debug mask = 0x0

Router#
```

FSU の実行

RPR でサポートされている FSU 手順を使用すると、システムをリロードしなくても、スーパーバイザ エンジン上の Cisco IOS イメージをアップグレードできます。



(注) EHSA から RPR へのアップグレードを初めて実行する場合は、両方のスーパーバイザ エンジンをリロードする必要があります。EHSA から FSU への移行はサポートされていません。

FSU を実行するには、次の作業を実行します。

	コマンド	目的
ステップ 1	<pre>Router# copy source_device:source_filename {disk0 disk1}:target_filename または Router# copy source_device:source_filename sup-bootflash:target_filename または Router# copy source_device:source_filename slavedisk0:target_filename または Router# copy source_device:source_filename slavesup-bootflash:target_filename</pre>	<p>新しい Cisco IOS イメージを両方のスーパーバイザ エンジン上のブートフラッシュにコピーします。</p>
ステップ 2	<pre>Router# config terminal Router(config)# config-register 0x2102 Router(config)# boot system flash device:file_name</pre>	<p>新しいイメージを起動するように、スーパーバイザ エンジンを設定します。</p>
ステップ 3	<pre>Router# copy running-config start-config</pre>	<p>設定を保存します。</p>
ステップ 4	<pre>Router# hw-module {module num} reset</pre>	<p>冗長スーパーバイザ エンジンをリロードして、再びオンライン状態に戻します（新しいバージョンの Cisco IOS ソフトウェアを実行します）。</p> <p>(注) 冗長スーパーバイザ エンジンをリロードする前に、すべての設定の同期変更が完了するまで、十分に待機してください。</p>
ステップ 5	<pre>Router# redundancy force-switchover</pre>	<p>冗長スーパーバイザ エンジンへのスイッチオーバーを手動で実行します。冗長スーパーバイザ エンジンが新しいアクティブ スーパーバイザ エンジンになり、新しい Cisco IOS イメージが稼動します。モジュールがリロードされ、モジュール ソフトウェアが新しいアクティブ スーパーバイザ エンジンからダウンロードされます。</p> <p>それまでアクティブだったスーパーバイザ エンジンが新しいイメージで再起動され、冗長スーパーバイザ エンジンになります。</p> <p>(注) EHSA から RPR への FSU 処理を実行するには、ステップ 5 で reload コマンドを実行します。</p>

次に、FSU の実行例を示します。

```
Router# config terminal
Router(config)# config-register 0x2102
Router(config)# boot system flash disk0:image_name
Router# copy running-config start-config
Router# hw-module reset
Router# redundancy force-switchover
Router#
```

冗長スーパーバイザ エンジンへのファイルのコピー

次のコマンドを使用して、冗長スーパーバイザ エンジン上の **disk0:** 装置にファイルをコピーします。

```
Router# copy source_device:source_filename slavedisk0:target_filename
```

次のコマンドを使用して、冗長スーパーバイザ エンジン上の **bootdisk:** 装置にファイルをコピーします。

```
Router# copy source_device:source_filename slavesup-bootdisk:target_filename
```

次のコマンドを使用して、冗長 PISA 上の **bootdisk:** 装置にファイルをコピーします。

```
Router# copy source_device:source_filename slavebootdisk:target_filename
```