CRS-1 と IOS XR の動作に関するベスト プラクティス

内容

概要
前提条件
要件
使用するコンポーネント
表記法
Cisco IOS XR の概要
プロセスとスレッド
プロセスとスレッドの状態
同期メッセージ パッシング
ブロックされたプロセスとプロセスの状態
重要なプロセスとその機能
Netio
Group Services Process（GSP）
BCDL バルク コンテンツ ダウンローダ
Lightweight Messaging（LWM）
envmon
CRS-1 ファブリックの概要
ファブリック プレーン
ファブリック モニタリング
コントロール プレーンの概要
Catalyst 6500 の設定
マルチシャーシ コントロール プレーンの管理
ROMMON および Monlib
アップグレードの手順
PLIM および MSC の概要
PLIM の加入過多
構成管理
セキュリティ
LPTS
内部パケット転送の方法
アウトオブバンド
関連情報

概要

このドキュメントでは、次について説明します。

- プロセスとスレッド
- CRS-1 ファブリック
- コントロール プレーン
- Rommon と Monlib
- Physical Layer Interface Module（PLIM; 物理レイヤ インターフェイス モジュール）と Modular Service Card（MSC; モジュラ サービス カード）
- 構成管理
- セキュリティ
- アウトオブバンド
- Simple Network Management Protocol（SNMP）

前提条件

要件

Cisco IOS® XRに関する知識があることが推奨されます。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Cisco IOS XR ソフトウェア
- CRS-1

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

表記法

ドキュメント表記の詳細は、『シスコ テクニカル ティップスの表記法』を参照してください。

Cisco IOS XR の概要

Cisco IOS XR は、規模の拡大縮小が可能な設計になっています。カーネルはマイクロカーネル アーキテクチャであり、プロセス管理、スケジューリング、信号、タイマーなど、不可欠なサービスだけが提供されます。ファイル システム、ドライバ、プロトコル スタック、アプリケーションなど、他のすべてのサービスはリソース マネージャとしてみなされ、メモリ保護されたユーザ領域で実行されます。プログラムの設計によっては、これらの他のサービスを実行時に追加したり削除したりできます。マイクロカーネルのフットプリントは 12 KB に過ぎません。マイクロカーネルと基本オペレーティング システムは QNX Software Systems によって提供され、Neutrino と呼ばれています。QNX は、リアルタイム オペレーティング システム設計を専門としています。マイクロカーネルはプリエンプティブであり、スケジューラは優先順位ベースです。このため、プロセス間でのコンテキスト スイッチングが非常に高速になり、必要な場合には常に最高の優先順位のスレッドが CPU にアクセスできます。これらは Cisco IOS XR で活用される利点の一部です。ただし、最大の利点としては、オペレーティング システム コア内部の Inter Process Communication（IPC; プロセス間通信）の継承設計が挙げられます。

Neutrino はメッセージ パッシング型のオペレーティング システムであり、すべてのスレッド間におけるプロセス間通信の基本的な手段はメッセージです。特定のサーバがサービスを提供する場合、メッセージの交換用にチャネルが作成されます。クライアントは、関連ファイルの記述子に直接マッピングすることでサーバ チャネルに接続して、サービスを使用します。クライアントとサーバの間のすべての通信は、同じメカニズムによって実行されます。CRS-1 などのスーパーコンピュータにとっては、このことが非常に大きな利点となります。標準の UNIX カーネルでローカルの読み取り動作が実行される場合は、次を検討してください。

- カーネルへのソフトウェア割り込み。
- カーネルによりファイル システムに手配される。
- データが受信される。

リモートの場合は次を検討します。

- カーネルへのソフトウェア割り込み。
- カーネルにより NFS が手配する。
- NFS によりネットワーキングの構成要素が呼び出される。
- リモートでネットワーキングの構成要素が手配される。
- NFS が呼び出される。
- カーネルによりファイル システムが手配される。

ローカル読み取りのセマンティックスとリモート読み取りのセマンティックスは同一ではありません。ファイル ロッキングと設定権限の引数とパラメータは異なります。

次のように QNX のローカルの場合を検討します。

- カーネルへのソフトウェア割り込み。
- カーネルによってファイル システムへのメッセージ パッシングが実行される。

次のように非ローカルの場合を検討します。

- カーネルへのソフトウェア割り込み。
- カーネルが IPC 転送メカニズムである QNET に移る。
- QNET がカーネルに移る。
- カーネルによりファイル システムが手配される。

引数引き渡しとファイル システム パラメータに関するすべてのセマンティックスは同一です。すべては IPC インターフェイスでデカップリングされているため、クライアントとサーバを完全に分離することが可能です。これは、時間や場所を選ばずに任意のプロセスが実行可能であることを意味します。要求を処理している特定のルート プロセッサの負荷が高すぎる場合は、DRP で稼働している別の CPU にこの処理を簡単に移行できます。異なる CPU 上でさまざまな処理を実行するスーパー コンピュータは、他のノードと簡単に通信できる複数のノードに分散しています。このインフラストラクチャは規模の拡大縮小を行う機会を提供するために配置されています。Cisco ではこの利点を活用して、メッセージ パッシング カーネルの基本的な動作に関連する付加的なソフトウェアを作成し、CRS ルータによる何千ものノードへの規模の拡大を可能にしました。ノード（この場合は CPU）では、Route Process（RP; ルート プロセス）、Distributed Route Processor（DRP; 分散ルート プロセッサ）、Modular Services Card（MSC; ジュラ サービス カード）、または Switch Processor（SP; スイッチ プロセッサ）のいずれであっても、OS のインスタンスが稼働します。

プロセスとスレッド

Cisco IOS XR 内では、プロセスとは 1 つ以上のスレッドが含まれる、メモリ上の保護領域を指します。プログラマの観点からは、スレッドで作業が実行され、各スレッドでは特定のタスクを実

行するための論理的な実行パスが遂行されます。実行のフローでスレッドに必要なメモリはスレッドが動作するプロセスに属するもので、他のプロセス スレッドからは保護されています。スレッドは実行の単位であり、スタックおよびレジスタなどの実行コンテキストが含まれています。プロセスは仮想アドレス空間を共有するスレッドのグループであり、プロセスに単一のスレッドだけを含ませることも可能ですが、多くの場合は複数のスレッドが含まれています。あるプロセス内のメモリに別のプロセス内の他のスレッドが書き込もうとすると、この違反プロセスは削除されます。プロセス内で複数のスレッドが動作している場合、そのスレッドはプロセス内の同じメモリにアクセスできるので、結果としてそのスレッドは別のスレッドのデータに上書きできることになります。同じプロセス内でのこのようなスレッドを回避するためには、リソースへの同期を維持する手順を実行します。

サービスに対する相互排他性を保証するために、スレッドでは Mutual Exclusion（MUTEX）と呼ばれるオブジェクトが使用されます。MUTEX を備えたスレッドとは、例のようにメモリの特定領域への書き込みが可能なスレッドです。MUTEX を備えていない他のスレッドは書き込みができません。Semaphore、Conditional Variable 別名 Condvar、Barrier、および Sleepon など、リソースへの同期を実現する他のメカニズムも存在します。これらについてはこのドキュメントでは説明していませんが、これらの役割の一部として同期サービスの提供があります。ここで説明した原理を Cisco IOS に当てはめてみると、Cisco IOS は多くのスレッドを動作させている単一のプロセスであり、これらのすべてのスレッドは同じメモリ領域にアクセスできます。ただし、Cisco IOS ではこれらのスレッドはプロセスと呼ばれます。

プロセスとスレッドの状態

Cisco IOS XR には、サービスを提供するサーバと、サービスを利用するクライアントが存在します。1 つのプロセスに、同じサービスを提供する多くのスレッドを含めることができます。別のプロセスには、任意の時点で特定のサービスを要求する可能性のある多くのクライアントを含めることができます。サーバへは常にアクセスできるわけではなく、クライアントがサービスへのアクセスを要求する場合、サーバが解放されるのを待つことになります。このような状態では、クライアントはブロックされていると呼ばれます。これをブロッキング クライアント サーバ モデルと呼びます。クライアントがブロックされる原因には、クライアントが MUTEX などのリソースを待っていたり、サーバがまだ応答していなかったりすることがあります。

ospf プロセス内のスレッド ステータスを確認するには、show process ospf コマンドを発行します。

```
RP/0/RP1/CPU0:CWDCRS#show process ospf
                  Job Id: 250
                     PID: 110795
         Executable path: /disk0/hfr-rout-3.2.3/bin/ospf
             Instance #: 1
             Version ID: 00.00.0000
                Respawn: ON
          Respawn count: 1
  Max. spawns per minute: 12
           Last started: Tue Jul 18 13:10:06 2006
          Process state: Run
          Package state: Normal
      Started on config: cfg/gl/ipv4-ospf/proc/101/ord_a/routerid
                   core: TEXT SHAREDMEM MAINMEM
              Max. core: 0
              Placement: ON
           startup_path: /pkg/startup/ospf.startup
                  Ready: 1.591s
              Available: 5.595s
```

```
        Process cpu time: 89.051 user, 0.254 kernel, 89.305 total
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
250    1      40K  10 Receive       0:00:11:0509 ospf
250    2      40K  10 Receive       0:01:08:0937 ospf
250    3      40K  10 Receive       0:00:03:0380 ospf
250    4      40K  10 Condvar       0:00:00:0003 ospf
250    5      40K  10 Receive       0:00:05:0222 ospf
```

OSPFプロセスにはジョブID(JID)が付与され、これは250です。これは、実行中のルータや通常は特定のバージョンのCisco IOS XRでは変更されません。ospf プロセス内には 5 つのスレッドが存在し、それぞれに独自の Thread ID (TID) が付与されています。 リストには、各スレッドのスタック容量、各スレッドの優先順位、および各スレッドの状態が示されています。

同期メッセージ パッシング

QNX がメッセージ パッシング型のオペレーティング システムであることはすでに述べてあります。実際には、QNX は同期メッセージ パッシング型のオペレーティング システムです。オペレーティング システムの問題の多くは、同期メッセージングに反映されます。これは、同期メッセージ パッシングにより問題が引き起こされているという意味ではなく、問題の症状が同期メッセージ パッシングで発現するという意味です。同期動作なので、CRS-1 オペレータはライフサイクルや状態情報へ簡単にアクセスでき、トラブルシューティング プロセスに役立てることができます。メッセージ パッシングのライフ サイクルは次のようなものです。

- サーバによってメッセージ チャネルが作成される。
- クライアントがサーバのチャネルに接続する (posix open に類似) 。
- クライアントによってサーバにメッセージが送信され (MsgSend) 、応答を待ちながらブロックされる。
- サーバではクライアントからのメッセージが受信され (MsgReceive) 、メッセージが処理され、クライントへ応答が行われる。
- クライアントではブロックが解除され、サーバからの応答が処理される。

このブロッキング クライアント/サーバ モデルは同期メッセージ パッシングです。つまり、クライアントではメッセージが送信されると、ブロックされます。サーバでメッセージが受信され、処理されてからクライアントに応答が返されると、クライアントではブロックが解除されます。詳細情報は次のとおりです。

- サーバは「RECEIVE」状態で待機する。
- クライアントでサーバにメッセージが送信され、「BLOCKED」状態になる。
- サーバではメッセージが受信され、ブロックが解除される (「RECEIVE」状態で待機している場合) 。
- クライアントは「REPLY」状態に移行する。
- サーバは「RUNNING」状態に移行する。
- サーバでメッセージが処理される。
- サーバからクライアントに応答される。
- クライアントでブロックが解除される。

クライアントやサーバの状態を表示するには、show process コマンドを発行します。

```
RP/0/RP1/CPU0:CWDCRS#show processes
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
1      1       0K  0  Ready       320:04:04:0649 procnto-600-smp-cisco-instr
1      3       0K  10 Nanosleep     0:00:00:0043 procnto-600-smp-cisco-instr
1      5       0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      7       0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      8       0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
```

```
1      11      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      12      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      13      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      14      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      15      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      16      0K  10 Receive       0:02:01:0207 procnto-600-smp-cisco-instr
1      17      0K  10 Receive       0:00:00:0015 procnto-600-smp-cisco-instr
1      21      0K  10 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      23      0K  10 Running       0:07:34:0799 procnto-600-smp-cisco-instr
1      26      0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1      31      0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1      33      0K  10 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      39      0K  10 Receive       0:13:36:0166 procnto-600-smp-cisco-instr
1      46      0K  10 Receive       0:06:32:0015 procnto-600-smp-cisco-instr
1      47      0K  56 Receive       0:00:00:0029 procnto-600-smp-cisco-instr
1      48      0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1      72      0K  10 Receive       0:00:00:0691 procnto-600-smp-cisco-instr
1      73      0K  10 Receive       0:00:00:0016 procnto-600-smp-cisco-instr
1      78      0K  10 Receive       0:09:18:0334 procnto-600-smp-cisco-instr
1      91      0K  10 Receive       0:09:42:0972 procnto-600-smp-cisco-instr
1      95      0K  10 Receive       0:00:00:0011 procnto-600-smp-cisco-instr
1      103     0K  10 Receive       0:00:00:0008 procnto-600-smp-cisco-instr
74     1       8K  63 Nanosleep     0:00:00:0001 wd-mbi
53     1      28K  10 Receive       0:00:08:0904 dllmgr
53     2      28K  10 Nanosleep     0:00:00:0155 dllmgr
53     3      28K  10 Receive       0:00:03:0026 dllmgr
53     4      28K  10 Receive       0:00:09:0066 dllmgr
53     5      28K  10 Receive       0:00:01:0199 dllmgr
270    1      36K  10 Receive       0:00:36:0091 qsm
270    2      36K  10 Receive       0:00:13:0533 qsm
270    5      36K  10 Receive       0:01:01:0619 qsm
270    7      36K  10 Nanosleep     0:00:22:0439 qsm
270    8      36K  10 Receive       0:00:32:0577 qsm
67     1      52K  19 Receive       0:00:35:0047 pkgfs
67     2      52K  10 Sigwaitinfo   0:00:00:0000 pkgfs
67     3      52K  19 Receive       0:00:30:0526 pkgfs
67     4      52K  10 Receive       0:00:30:0161 pkgfs
67     5      52K  10 Receive       0:00:25:0976 pkgfs
68     1       8K  10 Receive       0:00:00:0003 devc-pty
52     1      40K  16 Receive       0:00:00:0844 devc-conaux
52     2      40K  16 Sigwaitinfo   0:00:00:0000 devc-conaux
52     3      40K  16 Receive       0:00:02:0981 devc-conaux
52     4      40K  16 Sigwaitinfo   0:00:00:0000 devc-conaux
52     5      40K  21 Receive       0:00:03:0159 devc-conaux
65545  2      24K  10 Receive       0:00:00:0487 pkgfs
65546  1      12K  16 Reply         0:00:00:0008 ksh
66     1       8K  10 Sigwaitinfo   0:00:00:0005 pipe
66     3       8K  10 Receive       0:00:00:0000 pipe
66     4       8K  16 Receive       0:00:00:0059 pipe
66     5       8K  10 Receive       0:00:00:0149 pipe
66     6       8K  10 Receive       0:00:00:0136 pipe
71     1      16K  10 Receive       0:00:09:0250 shmwin_svr
71     2      16K  10 Receive       0:00:09:0940 shmwin_svr
61     1       8K  10 Receive       0:00:00:0006 mqueue
```

ブロックされたプロセスとプロセスの状態

ブロックされた状態にあるプロセスを表示するには、show process blocked コマンドを発行します。

```
RP/0/RP1/CPU0:CWDCRS#show processes blocked
```

```
  Jid       Pid Tid                 Name State  Blocked-on
65546      4106   1                  ksh Reply    4104  devc-conaux
  105     61495   2              attachd Reply   24597  eth_server
  105     61495   3              attachd Reply    8205  mqueue
  316     65606   1          tftp_server Reply    8205  mqueue
  233     90269   2              lpts_fm Reply   90223  lpts_pa
  325    110790   1            udp_snmpd Reply   90257  udp
  253    110797   4              ospfv3 Reply   90254  raw_ip
  337    245977   2              fdiagd Reply   24597  eth_server
  337    245977   3              fdiagd Reply    8205  mqueue
65762   5996770   1                exec Reply       1  kernel
65774   6029550   1                more Reply    8203  pipe
65778   6029554   1      show_processes Reply       1  kernel
RP/0/RP1/CPU0:CWDCRS#
```

同期メッセージ パッシングでは、異なるスレッド間でのプロセス間通信のライフ サイクルが簡単に追跡できます。スレッドは、いつの時点においても何らかの状態にあります。ブロックされた状態は、問題の症状を示している可能性があります。スレッドがブロックされた状態であっても必ずしも問題が発生しているわけではないので、show process blocked コマンドを発行する必要も、Cisco テクニカルサポートでサービスリクエストをオープンする必要もありません。ブロックされたスレッドはきわめて正常な状態です。

前述の出力に注目してください。リストの最初のスレッドに注目して、そのスレッドが ksh であり、応答が devc-conaux でブロックされていることに注意してください。クライアント（この場合は ksh）では devc-conaux プロセスにメッセージが送信され、サーバ（この場合は devc-conaux）では応答が行われるまで、ksh 応答がブロックされます。ksh はコンソールや AUX ポートで使用される UNIX シェルです。ksh ではコンソールからの入力が待機されますが、オペレータが入力を行わないためにコンソールからの入力が発生しない場合、ksh では何らかの入力が処理される時点までブロックされた状態が続きます。処理が行われたら、ksh は devc-conaux でブロックされた応答に戻ります。

これは正常な状態であり、問題を示しているわけではありません。ポイントは、ブロックされたスレッドは正常であり、その判断は XR のバージョン、使用するシステムの種類、設定した内容と show process blocked コマンドの出力をだれがどのように変更しているのかによって異なる点にあります。show process blocked コマンドの使用は、OS に関連した問題に対するトラブルシューティングを開始するためには優れた方法です。たとえば CPU の使用率が高いなどの問題が発生する場合は、正常値から外れている項目を表示するためにこのコマンドを使用します。

機能しているルータにとってはどのような状態が正常なのかを理解してください。これにより、プロセスのライフサイクルにトラブルシューティングを行う場合に、比較を行うためのベースラインが提供されます。

スレッドは、いつの時点においても何らかの状態にあります。次の表に、状態の一覧を示します。

状態	意味
DEAD	ダウン状態です。カーネルではスレッドのリソースの解放を待機中です。
実行中	CPU 上でアクティブに実行中です。
READY	CPU 上で実行中ではありませんが、実行の用意が整っています。
STOPPED	一時停止中（SIGSTOP シグナル）です。
SEND	サーバでメッセージが受信されるのを待機中です。

受信	クライアントでメッセージが送信されるのを待機中です。
REPLY	サーバからメッセージが応答されるのを待機中です。
STACK	スタックがさらに割り当てられるのを待機中です。
WAITPAGE	プロセス マネージャによってページ フォールトが解決されるのを待機中です。
SIGSUSPEND	シグナルを待機中です。
SIGWAITINFO	シグナルを待機中です。
NANOSLEEP	一定時間スリープ状態です。
MUTEX	MUTEX の取得を待機中です。
CONDVAR	条件変数にシグナルが送られるのを待機中です。
加入	別のスレッドの終了を待機中です。
INTR	割り込みを待機中です。
SEM	セマフォの取得を待機中です。

重要なプロセスとその機能

Cisco IOS XR には多くのプロセスがあります。次に、重要なプロセスの一部とその機能を説明します。

ウォッチドッグ システム モニタ（WDSysmon）

これは、プロセスのハングと使用可能メモリ低下の状況を検出するために提供されるサービスです。使用可能メモリの低下は、メモリ リークやその他の外来的な状況の結果として発生する可能性があります。ハングは、プロセス デッドロック、無限ループ、カーネル ロックアップ、スケジューリング エラーなど、さまざまな状況の結果として発生する可能性があります。マルチ スレッドの環境では、システムがデッドロック状態または単にデッドロックと呼ばれる状態になる可能性があります。デッドロックが発生する可能性があるのは、リソース コンテンションのために 1 つ以上のスレッドが処理を継続できない場合です。たとえば、スレッドAはスレッドBにメッセージを送信し、同時にスレッドBはスレッドAにメッセージを送信できます。両方のスレッドは互いに待機し、ブロック状態になり、両方のスレッドは永続的に待機します。これは 2 つのスレッドが関与する単純な状況ですが、多くのスレッドが使用するリソースをサーバが処理していて、別のスレッドでそのリソースがブロックされている場合、そのリソースへのアクセスを要求する多くのスレッドはサーバ上で送信がブロックされた待機状態となる可能性があります。

デッドロックは少数のスレッド間で発生する可能性があるものですが、結果として他のスレッドに影響を与える可能性があります。デッドロックは優れたプログラム設計により回避されるものですが、プログラムの設計と記述の適切さの程度には関係しません。場合によっては、特定のタイミングでのデータに依存するイベントの特定のシーケンスによりデッドロックが引き起こされる可能性があります。デッドロックは常に確定的に発生するわけではないので、通常、再現することが非常に困難です。WDSysmonには、Neutrinoがサポートする最高の優先順位で実行されるスレッドが多数あります。優先順位63で実行すると、優先順位ベースのプリエンプティブスケジ

ユーリング環境でスレッドがCPU時間を取得できます。WDSysmon はハードウェア ウオッチドッグ機能を使用して動作し、ハング状況を検出するソフトウェア プロセスを監視します。このような状況が検出されると、WDSysmon では状況に関する詳細情報が収集され、プロセスやカーネルのコアダンプ生成、syslog への書き込み、スクリプトの実行、デッドロックされたプロセスの削除が実行されます。問題の程度によっては、システムの動作を維持するためにルート プロセッサのスイッチオーバーを開始することができます。

```
RP/0/RP1/CPU0:CWDCRS#show processes wdsysmon
                  Job Id: 331
                     PID: 36908
         Executable path: /disk0/hfr-base-3.2.3/sbin/wdsysmon
              Instance #: 1
              Version ID: 00.00.0000
                 Respawn: ON
           Respawn count: 1
  Max. spawns per minute: 12
            Last started: Tue Jul 18 13:07:36 2006
           Process state: Run
           Package state: Normal
                    core: SPARSE
                Max. core: 0
                   Level: 40
              Mandatory: ON
            startup_path: /pkg/startup/wdsysmon.startup
            memory limit: 10240
                   Ready: 0.705s
        Process cpu time: 4988.295 user, 991.503 kernel, 5979.798 total

JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
331    1      84K  19 Receive       0:00:00:0029 wdsysmon
331    2      84K  10 Receive       0:17:34:0212 wdsysmon
331    3      84K  10 Receive       0:00:00:0110 wdsysmon
331    4      84K  10 Receive       1:05:26:0803 wdsysmon
331    5      84K  19 Receive       0:00:06:0722 wdsysmon
331    6      84K  10 Receive       0:00:00:0110 wdsysmon
331    7      84K  63 Receive       0:00:00:0002 wdsysmon
331    8      84K  11 Receive       0:00:00:0305 wdsysmon
331    9      84K  20 Sem           0:00:00:0000 wdsysmon
```

プロセス WDSysmon には 9 つのスレッドがあります。優先度10で4回、他の4回は11、19、20、および63です。プロセスが設計されると、プログラマはプロセス内の各スレッドに与える優先度を慎重に考慮します。すでに説明したとおり、スケジューラは優先順位に基づいているので、より高い優先順位のスレッドが低い優先順位のスレッドよりも常に優先されます。優先順位 63 はスレッドが実行可能な最高の優先順位であり、この場合ではスレッド 7 が該当します。スレッド 7 は CPU ホグを追跡する監視スレッドです。このスレッドはその監視対象である他のスレッドよりも高い優先順位で実行される必要がありますが、そうでない場合には実行の機会がまったく失われ、実行が意図された手順に従うことができなくなります。

Netio

Cisco IOS には、ファースト スイッチングとプロセス スイッチングのコンセプトが存在します。ファースト スイッチングは CEF コードを使用し、割り込み時に発生します。プロセス スイッチングは IP スイッチング コードである ip_input を使用する、スケジュールされたプロセスです。より高性能なプラットフォームでは、CEF スイッチングがハードウェアで実行され、ip_input は CPU でスケジュールされます。Cisco IOS XR では、Netio が ip_input に相当します。

```
P/0/RP1/CPU0:CWDCRS#show processes netio
```

```
                  Job Id: 241
                     PID: 65602
          Executable path: /disk0/hfr-base-3.2.3/sbin/netio
               Instance #: 1
                    Args: d
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 1
  Max. spawns per minute: 12
             Last started: Tue Jul 18 13:07:53 2006
            Process state: Run
            Package state: Normal
                     core: DUMPFALLBACK COPY SPARSE
                Max. core: 0
                    Level: 56
                Mandatory: ON
             startup_path: /pkg/startup/netio.startup
                    Ready: 17.094s
         Process cpu time: 188.659 user, 5.436 kernel, 194.095 total
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
241    1     152K  10 Receive       0:00:13:0757 netio
241    2     152K  10 Receive       0:00:10:0756 netio
241    3     152K  10 Condvar       0:00:08:0094 netio
241    4     152K  10 Receive       0:00:22:0016 netio
241    5     152K  10 Receive       0:00:00:0001 netio
241    6     152K  10 Receive       0:00:04:0920 netio
241    7     152K  10 Receive       0:00:03:0507 netio
241    8     152K  10 Receive       0:00:02:0139 netio
241    9     152K  10 Receive       0:01:44:0654 netio
241    10    152K  10 Receive       0:00:00:0310 netio
241    11    152K  10 Receive       0:00:13:0241 netio
241    12    152K  10 Receive       0:00:05:0258 netio
```

Group Services Process (GSP)

それぞれ独自のカーネル インスタンスが実行される数千のノードを持つどのスーパーコンピュータでも、通信に対するニーズが存在します。インターネットでは、1 対多の通信がマルチキャスト プロトコルを介して効率的に実行されます。GSPは、CRS-1内のIPCに使用される内部マルチキャストプロトコルです。GSPは、1対多の信頼性の高いグループ通信を提供し、非同期セマンティクスでコネクションレス化します。これによって、GSP は何千ものノードへと規模を拡大できます。

```
RP/0/RP1/CPU0:CWDCRS#show processes gsp
                  Job Id: 171
                     PID: 65604
          Executable path: /disk0/hfr-base-3.2.3/bin/gsp
               Instance #: 1
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 1
  Max. spawns per minute: 12
             Last started: Tue Jul 18 13:07:53 2006
            Process state: Run
            Package state: Normal
                     core: TEXT SHAREDMEM MAINMEM
                Max. core: 0
                    Level: 80
                Mandatory: ON
             startup_path: /pkg/startup/gsp-rp.startup
                    Ready: 5.259s
                Available: 16.613s
```

```
        Process cpu time: 988.265 user, 0.792 kernel, 989.057 total
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
171    1     152K  30 Receive       0:00:51:0815 gsp
171    3     152K  10 Condvar       0:00:00:0025 gsp
171    4     152K  10 Receive       0:00:08:0594 gsp
171    5     152K  10 Condvar       0:01:33:0274 gsp
171    6     152K  10 Condvar       0:00:55:0051 gsp
171    7     152K  10 Receive       0:02:24:0894 gsp
171    8     152K  10 Receive       0:00:09:0561 gsp
171    9     152K  10 Condvar       0:02:33:0815 gsp
171    10    152K  10 Condvar       0:02:20:0794 gsp
171    11    152K  10 Condvar       0:02:27:0880 gsp
171    12    152K  30 Receive       0:00:46:0276 gsp
171    13    152K  30 Receive       0:00:45:0727 gsp
171    14    152K  30 Receive       0:00:49:0596 gsp
171    15    152K  30 Receive       0:00:38:0276 gsp
171    16    152K  10 Receive       0:00:02:0774 gsp
```

BCDL バルク コンテンツ ダウンローダ

RP や MSC など、さまざまなノードへ確実にデータをマルチキャストするには、BCDL が使用されます。BCDL では、基本の転送として GSP が使用されます。BCDL によってメッセージの順序を追った配信が保証されます。BCDL 内部には、エージェント、プロデュサ、およびコンシューマが存在します。エージェントは、コンシューマへマルチキャストする前にデータを取得してバッファするために、プロデューサと通信するプロセスです。プロデューサは全員が必要とするデータを制作するプロセスで、コンシューマはプロデューサによって提供されるデータを受け取ることに関与するプロセスです。BCDL は Cisco IOS XR ソフトウェアのアップグレード中に使用されます。

Lightweight Messaging (LWM)

LWM は Cisco が作成したメッセージングの形式で、オペレーティング システムやトランスポート層との非依存性を目的として、Neutrino との間および相互間でのプロセス間通信を行うアプリケーション間に抽象化層を作成する設計になっています。Cisco が OS ベンダーを QNX から他社へ変更する場合、基礎となるオペレーティング システムの基本機能間の抽象化層がオペレーティング システムへの依存を解消するのに役立ち、他のオペレーティング システムへのポーティングの手助けとなります。LWM では同期が保証されたメッセージ配信が提供されますが、これはネイティブの Neutrino メッセージ パッシングと同様、受信者が応答するまで送信者がブロックする原因となります。

また、LWM では 40 ビット パルスでの非同期のメッセージ配信も提供されます。非同期メッセージは非同期で送信されますが、これはメッセージがキューイングされて送信者でのブロックが発生しないことを意味します。ただし、サーバから次に使用できるメッセージのポーリングが行われる場合には、サーバではメッセージが非同期には受信されないことになります。LWM はクライアント/サーバとして構造化されています。サーバは、メッセージを傍受する耳を提供するチャネルを作成し、この作成したばかりのチャンネルでループによるメッセージ受信のリスニングが実行されている間は待機します。メッセージが受信されると、ブロックが解除されてクライアント識別子が取得されますが、これは事実上、受信されたメッセージからの受信 ID と同じものです。次に、サーバによって一部の処理が実行され、後でクライアント識別子へのメッセージ応答が行われます。

クライアント側では、メッセージ接続が行われます。接続先へ識別子を通過させてからメッセージ送信が行われて、ブロックされます。サーバでの処理が終了すると、応答が行われ、クライアントのブロックが解除されます。事実上、これは Neutrino のネイティブ メッセージ パッシングと同じなので、抽象化層は非常に薄いものになります。

LWM は高度なパフォーマンスを目的とした最小限のシステム コールとコンテキスト スイッチを使用した設計であり、Cisco IOS XR 環境では IPC の優先方式です。

envmon

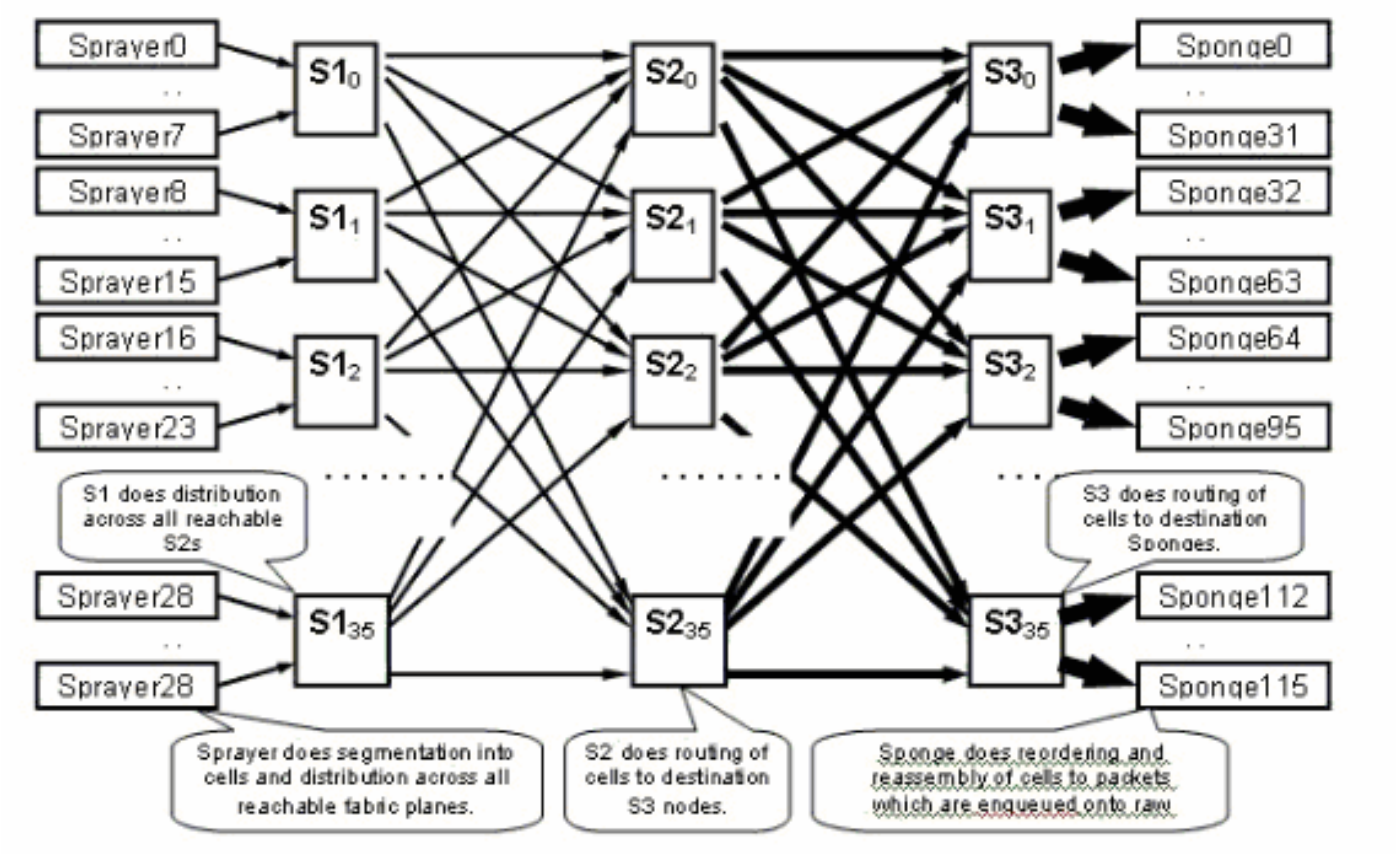
最も基礎的なレベルでは、環境モニタリング システムは物理的なパラメータ（温度、電圧、ファン速度など）が動作範囲から外れる場合に警告を行い、ハードウェアが破損する可能性がある臨界値に近づいているハードウェアをシャットダウンする役割を担っています。環境モニタリング システムは、このタスクを実行するために、使用できる各ハードウェア センサーを定期的に監視し、測定された値をカード固有のしきい値と比較して、必要に応じて警告を行います。システムの初期化において開始された、定期的にシャーシ内のすべてのハードウェア センサー（電圧、温度、ファン速度など）にポーリングを行う永続的なプロセスによって、このデータは外部の管理クライアントに提供されます。また、この定期的なプロセスではセンサーの値を警告のしきい値と比較し、フォールト マネージャによる後続の処理のために、環境に関する警告をシステム データベースへ発行します。センサーの値が危険なほど範囲から外れている場合は、環境モニタリング プロセスによってカードがシャット ダウンされることがあります。

CRS-1 ファブリックの概要

- マルチステージファブリック：3ステージベントポロジ
- 輻輳を最小化するファブリック内部でのダイナミック ルーティング
- セルベース：136バイトセル、120バイトデータペイロード
- トラフィックの分離を改善し、ファブリック内のバッファリング要件を最小化するフロー制御
- ステージからステージへのスピードアップ配信
- 2つのトラフィックキャストをサポート（ユニキャストおよびマルチキャスト）
- キャストあたり 2 つのトラフィック優先順位のサポート（高い優先順位と低い優先順位）
- 1M ファブリック マルチキャスト グループのサポート（FGID）
- コスト効率の高い耐障害性：1+1ではなくファブリックプレーンを使用したN+1またはN+kの冗長性を大幅にコストを増加

単一シャーシ モードで稼働している場合は、S1、S2、および S3 の各 ASIC は同じファブリック カードに配置されます。また、一般的に、このカードは S123 カードとも呼ばれます。マルチシャーシ構成の場合は、S2 が分離され、Fabric Card Chassis（FCC; ファブリック カード シャーシ）に配置されます。 この設定では、プレーンを構成するために 2 枚のファブリック カード（S2 カードおよび S13 カード）が必要です。各 MSC は、冗長性を提供するために 8 つのファブリック プレーンに接続されるので、1 つ以上のプレーンが失われた場合でもトラフィックはファブリックを通過しますが、ファブリックを通過できる集約トラフィックは少なくなります。CRS は、7 つのプレーンだけを使用しても、大部分のパケット サイズに対して回線レートで動作を続けることができます。バックプレッシャは奇数と偶数のプレーンを介してファブリックに送信されます。奇数および偶数のプレーンの 2 つのプレーンよりも少ない装備しかないシステムを稼働することは推奨されません。2 つ未満のプレーンの設定はサポートされない設定です。

ファブリック プレーン



上のダイアグラムは、1 つのプレーンを表しています。このダイアグラムを 8 倍する必要があります。これは、LC のスプレイヤ (ingressq) ASIC が 8 つの S1 (プレーンあたり 1 つの S1) に接続されることを意味します。 各ファブリック プレーンの S1 は、8 つのスプレイヤに接続されます。

- シャーシの上位 8 つの LC
- 下位 8 つの LC

16スロットのLCシャーシごとに16のS1があります。上のLCの場合は8 (プレーンごとに1つ)、下のLCの場合は8です。

単一の 16 スロット シャーシでは、S123 ファブリック カードが 2 つの S1、2 つの S2、および 4 つの S3 が備わっています。これはファブリックのスピードアップ計算の一部です。トラフィックが入る際にファブリックを終了させる可能性がある、2 倍のトラフィックが存在します。1 スプレイヤと比較すると、現在、LC あたり 2 つのスポンジ (fabricq) も存在します。これにより、複数の入力 LC によって出力 LC が過負荷になる場合に、出力 LC でのバッファリングが可能になります。出力 LC では、ファブリックからのこの追加帯域幅を吸収できます。

ファブリック モニタリング

プレーンの可用性と接続性：

```
admin show controller fabric plane all
admin show controller fabric connectivity all detail
```

プレーンがセルを送受信しているかどうか、また、一部のエラーが増加しているかどうかを確認します。

```
admin show controllers fabric plane all statistics
```

前述のコマンドの略語を次に示します。

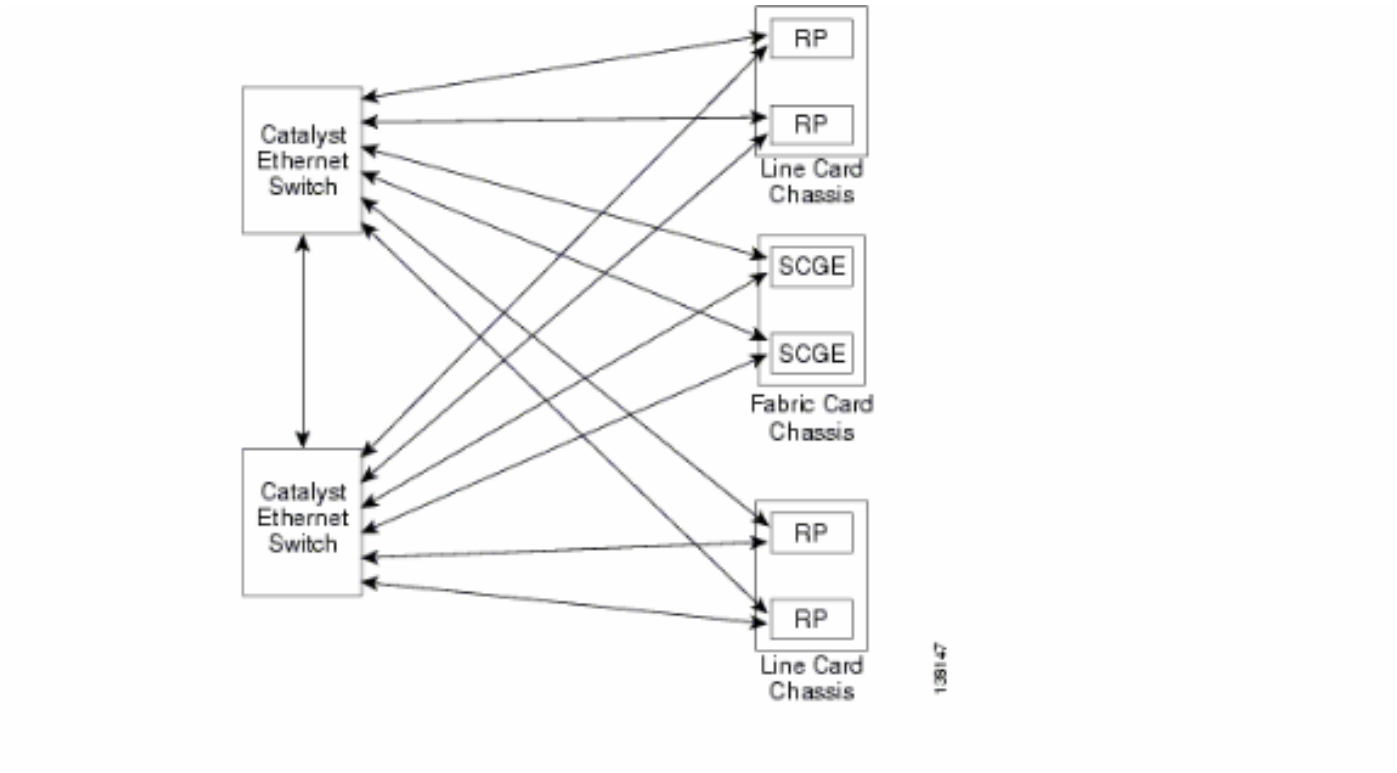
- CE：修正可能なエラー
- UCE:Uncorrectableエラー
- PE：パリティエラー

ブートアップ時にはいくつかエラーが発生する可能性があるため、それらが検出されても気にする必要はありません。これらのフィールドは実行時には増加しないはずです。これらのフィールドが増加している場合、ファブリック内の問題を示している可能性があります。ファブリック プレーンごとのエラー詳細を取得するには、次のコマンドを発行します。

```
admin show controllers fabric plane <0-7> statistics detail
```

コントロール プレーンの概要

現在、ラインカード シャーシおよびファブリック シャーシの間のコントロール プレーンの接続は、RP（LCC）および SCGE（FCC）上のギガビット イーサネット ポートを経由しています。ポート間の相互接続は、2 つ以上のギガビット イーサネット ポートを介して接続が可能な Catalyst 6500 スイッチのペアを経由して提供されます。



Catalyst 6500 の設定

マルチシャーシ コントロール プレーンに使用される Catalyst スイッチに推奨される設定を次に示します。

- すべてのポートで単一の VLAN が使用される。
- すべてのポートがアクセス モードで実行される（トランキングなし）。
- ループ防止用にスパニング ツリー 802.1w/s が使用される。
- 2 つのスイッチを相互接続するために 2 つ以上のリンクが使用され、ループ防止用に STP が

使用される。チャネリングは推奨されません。
- IOS-XR では標準ベースの 802.1s がサポートされないので、CRS-1 RP および SCGE に接続するポートでは先行標準モードが使用される。
- スイッチ間およびスイッチと RP/SCGE の間で接続するポート上では、UDLD を有効にする必要がある。
- CRS-1 ではデフォルトで UDLD が有効である。

マルチシェルフ システムで Catalyst 6500 を設定する方法については、『マルチシェルフ システムでの Cisco IOS XR ソフトウェアの始動』を参照してください。

マルチシャーシ コントロール プレーンの管理

マルチシャーシ システムのコントロール プレーン接続を提供する Catalyst 6504-E シャーシは、下記の管理サービス用に設定されます。

- 各 PoP で LAN スイッチに接続する、ポート ギガビット 1/2 を介したインバンド管理。狭い範囲のサブネットとプロトコルにだけアクセスが許可されます。
- システムの時刻を設定するために NTP が使用される。
- 標準ホストへの syslog が実行される。
- 不可欠な機能用に SNMP ポーリングとトラップを有効にできる。

注：動作中のCatalystに変更を加える必要はありません。計画済みの変更には事前テストを行う必要がありますが、このテストはメンテナンスの時間帯に行うことを強く推奨いたします。

次に管理設定の例を示します。

```
#In-band management connectivity
interface GigabitEthernet2/1
 description *CRS Multi-chassis Management Ethernet - DO NOT TOUCH*
 ip address [ip address] [netmask]
 ip access-group control_only in
!
!
ip access-list extended control_only
 permit udp [ip address] [netmask] any eq snmp
 permit udp [ip address] [netmask] eq ntp any
 permit tcp [ip address] [netmask] any eq telnet

#NTP

ntp update-calendar
ntp server [ip address]

#Syslog
logging source-interface Loopback0
logging [ip address]
logging buffered 4096000 debugging
no logging console

#RADIUS
aaa new-model
aaa authentication login default radius enable
enable password {password}
radius-server host [ip address] auth-port 1645 acct-port 1646
radius-server key {key}

#Telnet and console access
```

```
!
access-list 3 permit [ip address]
!
line con 0
 exec-timeout 30 0
 password {password}
line vty 0 4
 access-class 3 in
 exec-timeout 0 0
password {password}
```

ROMMON および Monlib

Cisco monlib は実行可能プログラムであり、デバイスに格納されていて、実行には ROMMON により RAM にロードされます。ROMMON では、デバイス上のファイルへのアクセスに monlib が使用されます。ROMMON のバージョンはアップグレード可能ですが、アップグレードは Cisco テクニカルサポートの推奨に基づいて行う必要があります。最新の ROMMON バージョンは 1.40 です。

アップグレードの手順

次のステップを実行します。

1. ROMMON バイナリを Cisco CRS-1 ROMMON（登録ユーザ専用）からダウンロードします。
2. TAR ファイルを展開し、6 つの BIN ファイルを Disk0 の CRS ルート ディレクトリにコピーします。

```
RP/0/RP0/Router#dir disk0:/*.bin

Directory of disk0:

65920       -rwx  360464      Fri Oct 28 12:58:02 2005  rommon-hfr-ppc7450-sc-dsmp-A.bin
66112       -rwx  360464      Fri Oct 28 12:58:03 2005  rommon-hfr-ppc7450-sc-dsmp-B.bin
66240       -rwx  376848      Fri Oct 28 12:58:05 2005  rommon-hfr-ppc7455-asmp-A.bin
66368       -rwx  376848      Fri Oct 28 12:58:06 2005  rommon-hfr-ppc7455-asmp-B.bin
66976       -rwx  253904      Fri Oct 28 12:58:08 2005  rommon-hfr-ppc8255-sp-A.bin
67104       -rwx  253492      Fri Oct 28 12:58:08 2005  rommon-hfr-ppc8255-sp-B.bin
```

3. **show diagコマンドを使用します | inc ROM|NODE|PLIMコマンドを発行**して、現在のrommonバージョンを表示します。

```
RP/0/RP0/CPU0:ROUTER(admin)#show diag | inc ROM|NODE|PLIM
NODE 0/0/SP : MSC(SP)
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
PLIM 0/0/CPU0 : 4OC192-POS/DPT
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/2/SP : MSC(SP)
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
PLIM 0/2/CPU0 : 8-10GbE
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/4/SP : Unknown Card Type
NODE 0/6/SP : MSC(SP)
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
PLIM 0/6/CPU0 : 16OC48-POS/DPT
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/RP0/CPU0 : RP
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/RP1/CPU0 : RP
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/SM0/SP : FC/S
```

```
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
NODE 0/SM1/SP : FC/S
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
NODE 0/SM2/SP : FC/S
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
NODE 0/SM3/SP : FC/S
ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
```

4. ROMMON をアップグレードするには、ADMIN モードに切り替えて upgrade rommon a all disk0 コマンドを使用します。

```
RP/0/RP0/CPU0:ROUTER#admin
RP/0/RP0/CPU0:ROUTER(admin)#upgrade rommon a all disk0
Please do not power cycle, reload the router or reset any nodes until
 all upgrades are completed.
Please check the syslog to make sure that all nodes are upgraded successfully.
If you need to perform multiple upgrades, please wait for current upgrade
 to be completed before proceeding to another upgrade.
Failure to do so may render the cards under upgrade to be unusable.
```

5. ADMINモードを終了し、show logを入力します。 | inc「OK, ROMMON A」と入力し、すべてのノードが正常にアップグレードされたことを確認します。正常にアップグレードされていないノードがある場合は、手順 4 に戻って再プログラムします。

```
RP/0/RP0/CPU0:ROUTER#show logging | inc "OK, ROMMON A"
RP/0/RP0/CPU0:Oct 28 14:40:57.223 PST8: upgrade_daemon[380][360]: OK, ROMMON A is
programmed successfully. SP/0/0/SP:Oct 28 14:40:58.249 PST8: upgrade_daemon[125][121]: OK,
ROMMON A is programmed successfully. SP/0/2/SP:Oct 28 14:40:58.251 PST8:
upgrade_daemon[125][121]: OK, ROMMON A is programmed successfully. LC/0/6/CPU0:Oct 28
14:40:58.336 PST8: upgrade_daemon[244][233]: OK, ROMMON A is programmed successfully.
LC/0/2/CPU0:Oct 28 14:40:58.365 PST8: upgrade_daemon[244][233]: OK, ROMMON A is programmed
successfully. SP/0/SM0/SP:Oct 28 14:40:58.439 PST8: upgrade_daemon[125][121]: OK, ROMMON A
is programmed successfully. SP/0/SM1/SP:Oct 28 14:40:58.524 PST8: upgrade_daemon[125][121]:
OK, ROMMON A is programmed successfully. LC/0/0/CPU0:Oct 28 14:40:58.530 PST8:
upgrade_daemon[244][233]: OK, ROMMON A is programmed successfully. RP/0/RP1/CPU0:Oct 28
14:40:58.593 PST8: upgrade_daemon[380][360]: OK, ROMMON A is programmed successfully.
SP/0/6/SP:Oct 28 14:40:58.822 PST8: upgrade_daemon[125][121]: OK, ROMMON A is programmed
successfully. SP/0/SM2/SP:Oct 28 14:40:58.890 PST8: upgrade_daemon[125][121]: OK, ROMMON A
is programmed successfully. SP/0/SM3/SP:Oct 28 14:40:59.519 PST8: upgrade_daemon[125][121]:
OK, ROMMON A is programmed successfully.
```

6. ROMMON をアップグレードするには、ADMIN モードに切り替えて upgrade rommon b all disk0 コマンドを使用します。

```
RP/0/RP0/CPU0:ROUTER#admin
RP/0/RP0/CPU0:ROUTER(admin)#upgrade rommon b all disk0
Please do not power cycle, reload the router or reset any nodes until
 all upgrades are completed.
Please check the syslog to make sure that all nodes are upgraded successfully.
If you need to perform multiple upgrades, please wait for current upgrade
 to be completed before proceeding to another upgrade.
Failure to do so may render the cards under upgrade to be unusable.
```

7. ADMINモードを終了し、show logを入力します。 | inc「OK, ROMMON B」をクリックし、すべてのノードが正常にアップグレードされたことを確認します。正常にアップグレードされていないノードがある場合は、手順 4 に戻って再プログラムします。

```
RP/0/RP0/CPU0:Router#show logging | inc "OK, ROMMON B"
RP/0/RP0/CPU0:Oct 28 13:27:00.783 PST8: upgrade_daemon[380][360]: OK,
ROMMON B is programmed successfully.
LC/0/6/CPU0:Oct 28 13:27:01.720 PST8: upgrade_daemon[244][233]: OK,
ROMMON B is programmed successfully.
SP/0/2/SP:Oct 28 13:27:01.755 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
LC/0/2/CPU0:Oct 28 13:27:01.775 PST8: upgrade_daemon[244][233]: OK,
ROMMON B is programmed successfully.
SP/0/0/SP:Oct 28 13:27:01.792 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
```

```
SP/0/SM0/SP:Oct 28 13:27:01.955 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
LC/0/0/CPU0:Oct 28 13:27:01.975 PST8: upgrade_daemon[244][233]: OK,
ROMMON B is programmed successfully.
SP/0/6/SP:Oct 28 13:27:01.989 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
SP/0/SM1/SP:Oct 28 13:27:02.087 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
RP/0/RP1/CPU0:Oct 28 13:27:02.106 PST8: upgrade_daemon[380][360]: OK,
ROMMON B is programmed successfully.
SP/0/SM3/SP:Oct 28 13:27:02.695 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
SP/0/SM2/SP:Oct 28 13:27:02.821 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
```

8. upgrade コマンドでは、ブートフラッシュの特殊な予約済みセクションに新しい ROMMON が焼き付けられるだけです。、新しい ROMMON は、カードがリロードされるまで非アクティブのままです。したがって、ユーザがカードをリロードすると、新しい ROMMON がアクティブになります。一度に 1 つずつ各ノードをリセットするか、ルータ全体をリセットするだけで、これが実行されます。

```
Reload Router:
RP/0/RP0/CPU0:ROUTER#hw-module node 0/RP0/CPU0 or 0/RP1/CPU0 reload (depends on which on is
in Standby Mode.
RP/0/RP0/CPU0:ROUTER#reload
!--- Issue right after the first command. Updating Commit Database. Please wait...[OK]
Proceed with reload? [confirm] !--- Reload each Node. For Fan Controllers (FCx), !--- Alarm
Modules (AMx), Fabric Cards (SMx), and RPs (RPx), !--- you must wait until the reloaded
node is fully reloaded !--- before you reset the next node of the pair. But non-pairs !---
can be reloaded without waiting. RP/0/RP0/CPU0:ROUTER#hw-module node 0/RP0/CPU0 or
0/RP1/CPU0 reload
!--- This depends on which on is in Standby Mode. RP/0/RP0/CPU0:ROUTER#hw-module node
0/FC0/SP
RP/0/RP0/CPU0:ROUTER#hw-module node 0/AM0/SP
RP/0/RP0/CPU0:ROUTER#hw-module node 0/SM0/SP
!--- Do not reset the MSC and Fabric Cards at the same time. RP/0/RP0/CPU0:ROUTER#hw-module
node 0/0/CPU
```

9. show diagコマンドを使用します | inc ROM|NODE|PLIMコマンドを発行し、現在の ROMMONバージョンを確認します。

```
RP/0/RP1/CPU0:CRS-B(admin)#show diag | inc ROM|NODE|PLIM
NODE 0/0/SP : MSC(SP)
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
PLIM 0/0/CPU0 : 4OC192-POS/DPT
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]
NODE 0/2/SP : MSC(SP)
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
PLIM 0/2/CPU0 : 8-10GbE
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]
NODE 0/6/SP : MSC(SP)
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
PLIM 0/6/CPU0 : 16OC48-POS/DPT
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]
NODE 0/RP0/CPU0 : RP
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]
NODE 0/RP1/CPU0 : RP
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]]
NODE 0/SM0/SP : FC/S
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
NODE 0/SM1/SP : FC/S
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
NODE 0/SM2/SP : FC/S
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
```

```
NODE 0/SM3/SP : FC/S
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
```

注：CRS-8およびファブリックシャーシでは、ROMMONによってファン速度もデフォルトの4000 RPMに設定されます。

PLIM および MSC の概要

次に CRS-1 ルータでのパケット フローを紹介しますが、下記の用語は特に区別なく使用されています。

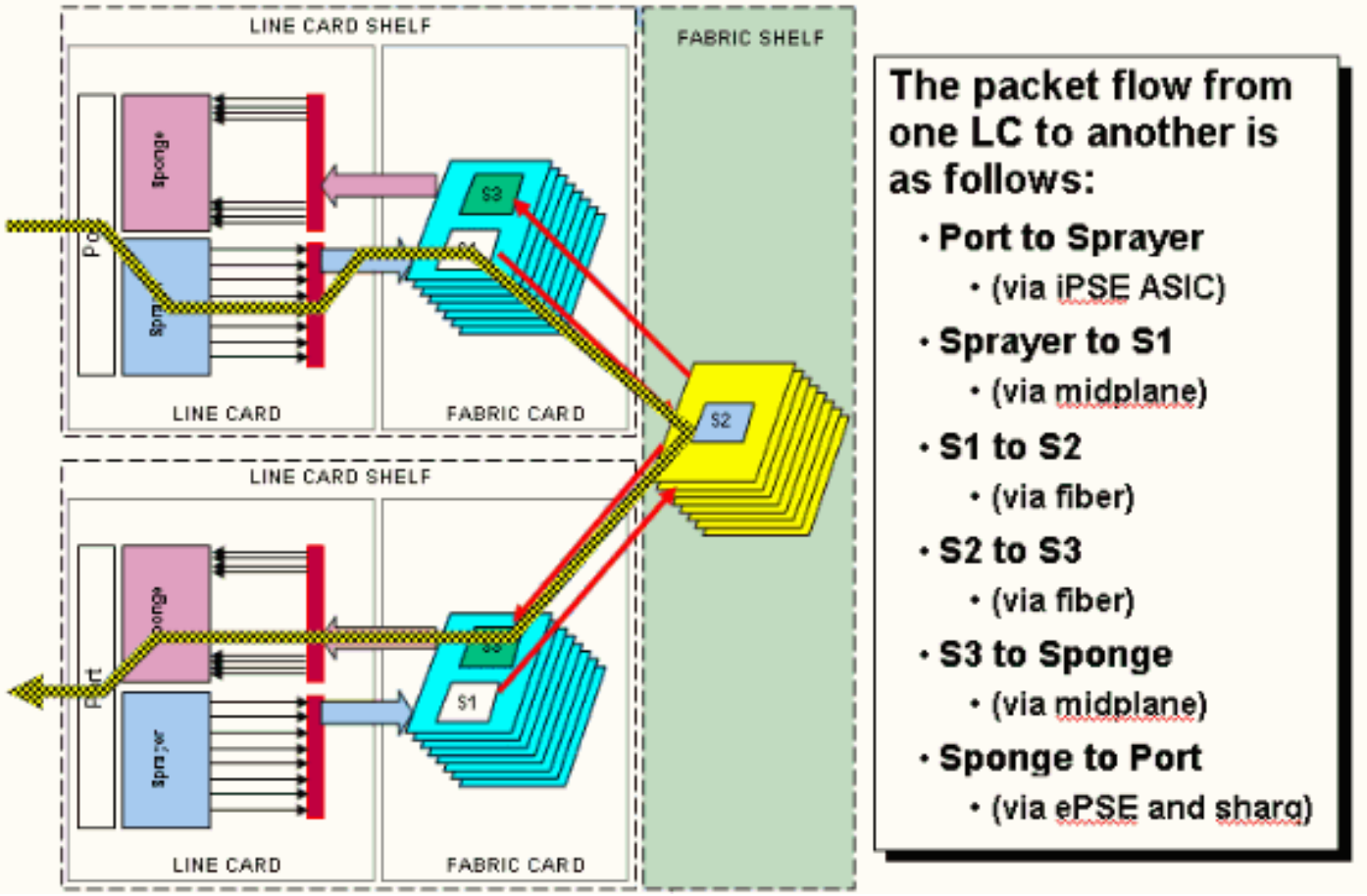
IngressQ ASIC は、Sprayer ASIC とも呼ばれます。

FabricQ ASIC は、Sponge ASIC とも呼ばれます。

EgressQ ASIC は、Sharq ASIC とも呼ばれます。

SPP は Packet Switch Engine（PSE; パケット交換エンジン）ASIC とも呼ばれます。

Rx PLIM > Rx SPP > Ingress Q > Fabric > Fabric Q > Tx SPP > Egress Q > Tx PLIM (Spplyer) (Sharq)



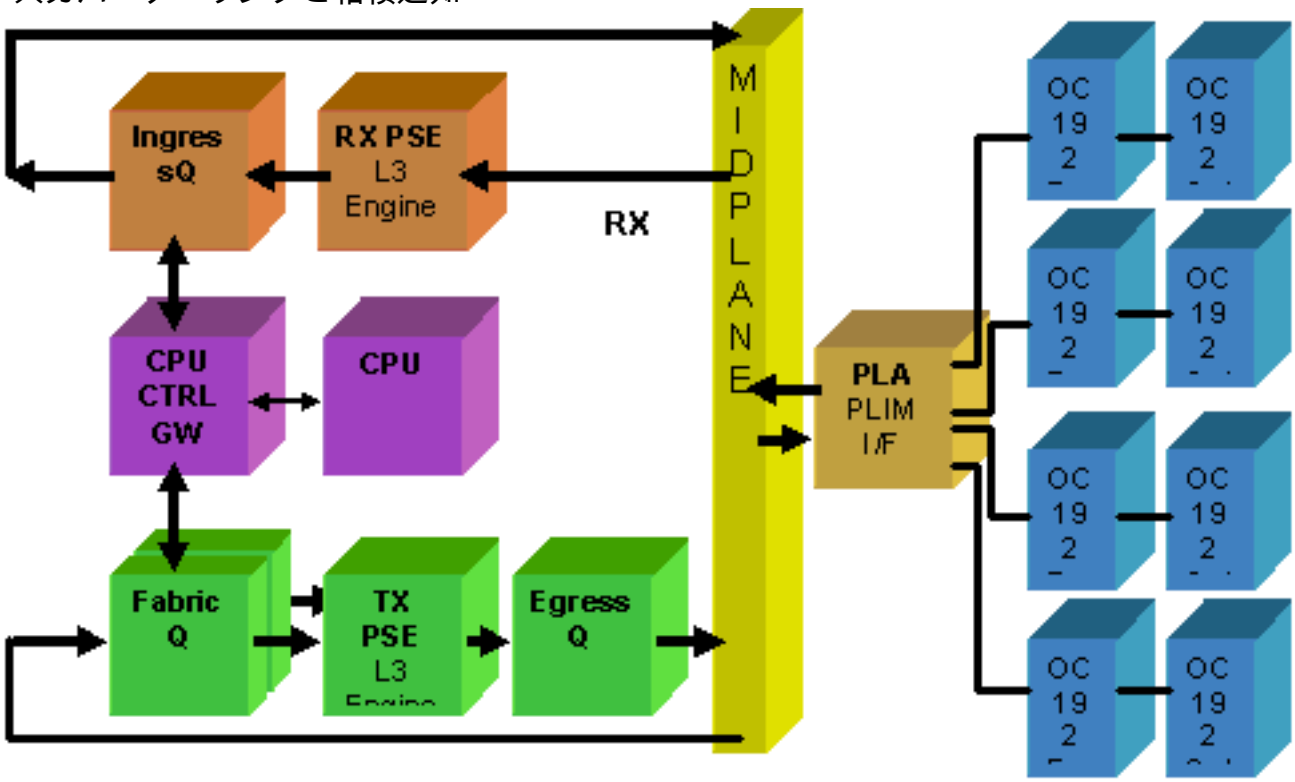
パケットは Physical Layer Interface Module（PLIM; 物理レイヤ インターフェイス モジュール）で受信されます。

PLIM には、PLIM が接続する MSC の物理インターフェイスがあります。PLIM と MSC は、シャーシ バックプレーンを介して接続された別々のカードです。結果として、特定の MSC のインターフェイスの種類は、その MSC が接続する PLIM の種類によって定義されます。PLIM の種類に

従って、カードにはインターフェイスの物理メディアとフレーム同期を提供するさまざまな ASIC がたくさん含まれます。PLIM ASIC の目的は、MSC と物理接続の間にインターフェイスを提供することです。PLIM では、光ファイバの終端処理、光電気変換の実行、SDH/Sonet/イーサネット/HDLC/PPP などでのメディア フレーム同期の終端、CRC のチェック、バッファ ヘッダーと呼ばれる制御情報の追加、および残りのビットの MSC への転送が実行されます。PLIM では、HDLC または PPP のキープアライブのソースとシンクは実行されません。これらは MSC の CPU で処理されます。

PLIM では、次の機能も提供されます。

- 1/10 ギガビット イーサネットの MAC フィルタリング
- 1/10 ギガビット イーサネットの入力/出力 MAC アカウンティング
- 1/10 ギガビット イーサネットの VLAN フィルタリング
- 1/10 ギガビット イーサネットの VLAN アカウンティング
- 入力バッファリングと輻輳通知



PLIM の加入過多

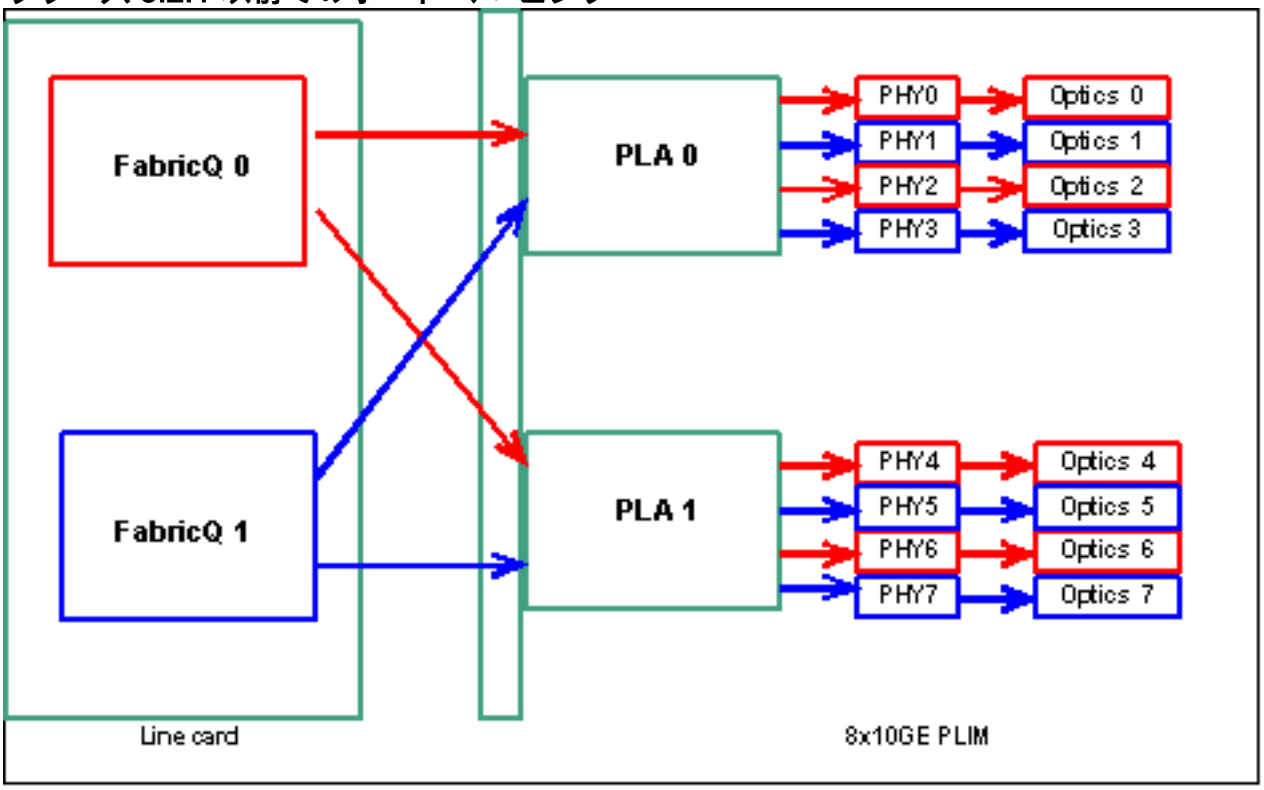
10GE PLIM

8 X 10G の PLIM では、約 80 Gbps のトラフィックを終端処理する機能が提供される一方、MSC の転送機能は最大 40 Gbps です。PLIM で使用できるすべてのポートにデータが入力される場合は加入過多が発生し、プレミアム トラフィックが誤って廃棄されないことを保証する QoS モデリングがきわめて重要になります。状況によっては、加入過多の選択の余地はなく、回避が必要

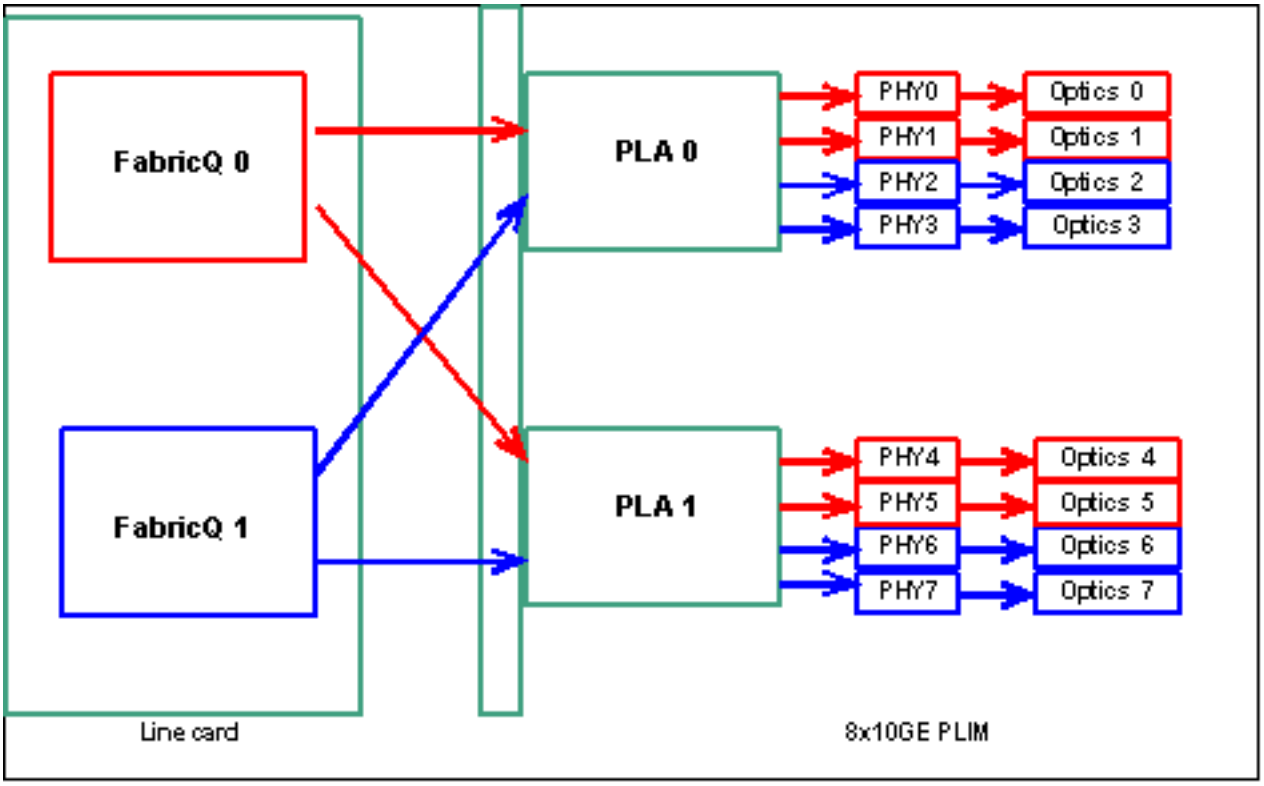
です。これを実行するには、8 つのポートのうち 4 つのポートだけを使用する必要があります。また、MSC および PLIM 内部で最適な帯域幅がこの 4 つのポートで使用できるように注意する必要もあります。

注：ポートマッピングはリリース3.2.2以降で変更されています。下記のダイアグラムを参照してください。

リリース 3.2.1 以前でのポート マッピング



リリース 3.2.2 以降でのポート マッピング



前述のように、物理ポートに対するサービスの提供は 2 つの FabricQ ASIC のどちらかが行います。ASIC へのポートの割り当ては静的に定義され、変更できません。また、8 X 10G の PLIM には PLA ASIC が 2 つあります。1つ目のPLAサービスポート0 ~ 3、2つ目のサービス4 ~ 7。8 X 10G PLIMでの1つのPLAの帯域幅容量は約24 Gbpsです。単一の FabricQ ASIC のスイッチング キャパシティは、約 62 Mpps です。

ポート 0 ～ 3 またはポート 4 ～ 7 を使用する場合、PLA (24 Gbps) の帯域幅容量は 4 つのポートすべてで共有され、これにより全体的なスループットが制限されます。ポート0、2、4、6(3.2.1まで)または0、1、4、5(3.2.2以降)を入力すると、これらのポートすべてが1つのFabricQ ASICによって処理されます。スイッチング容量は62 Mppsですが、これもスループット容量を制限します。

最適なパフォーマンスを達成するには、PLA と FabricQ ASIC の両方で最高の効率性を実現するようにポートを使用することが最適です。

SIP-800/SPA

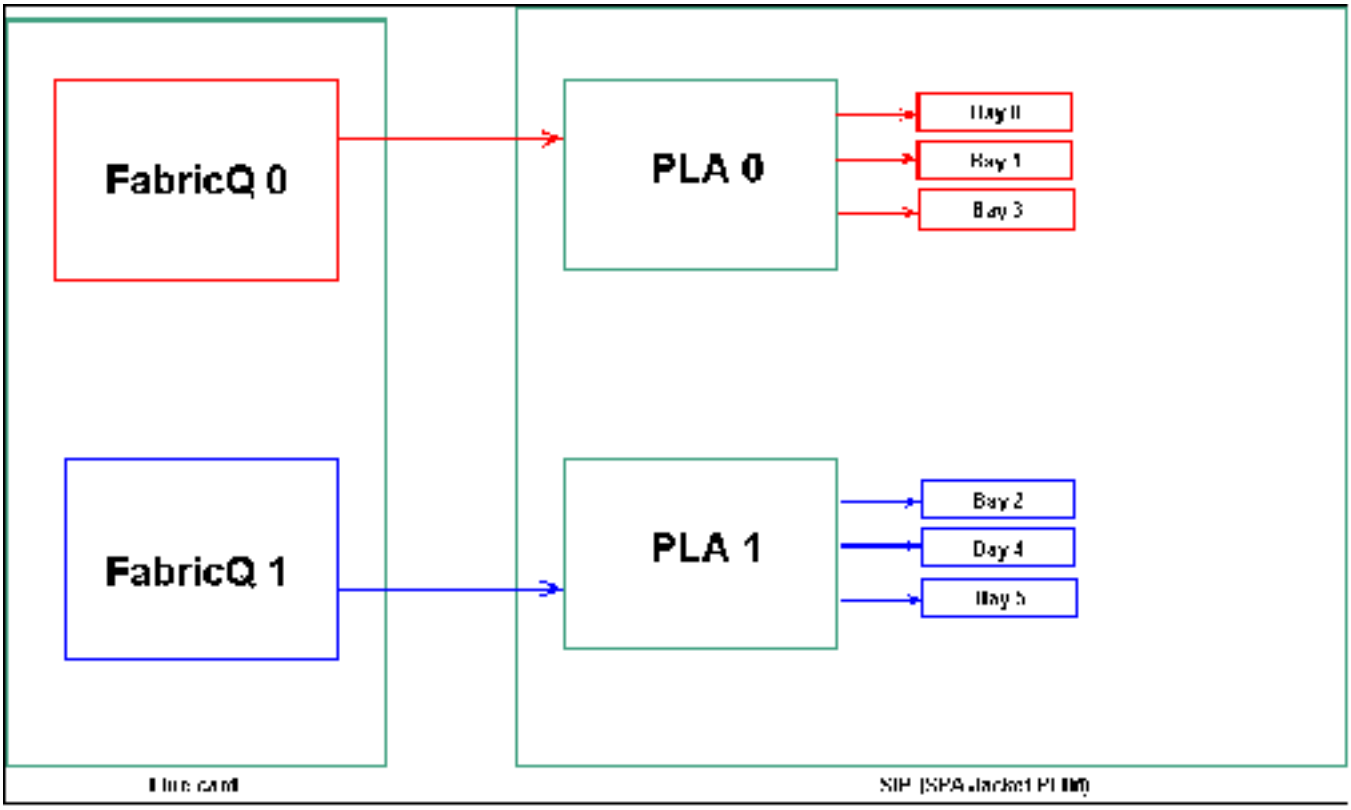
SIP-800 PLIM では、Service Port Adapter (SPA) と呼ばれるモジュラ インターフェイス カードを使用して動作する機能が提供されています。 SIP-800 では、理論上 60 Gbps インターフェイス キャパシティを備えた 6 つの SPA ベイが提供されます。MSC の転送キャパシティは、最大 40 Gbps です。SIP-800 のすべてのベイが使用されると、SPA の種類によっては加入過多が発生し、プレミアム トラフィックが誤って廃棄されないようにするには QoS モデリングがきわめて重要になる可能性があります。

注：オーバーサブスクリプションは、POSインターフェイスではサポートされていません。ただ

し、適切なスループット キャパシティが確実に提供されるようにするには、10 Gb POS SPA の配置が適切です。10 GbイーサネットSPAはIOS-XRリリース3.4でのみサポートされています。このSPAはオーバーサブスクリプション機能を提供します。

状況によっては、加入過多の選択の余地はなく、回避が必要です。これを実行するには、6 つのベイのうち 4 つのベイだけを使用する必要があります。また、MSC および PLIM 内部で最適な帯域幅がこの 4 つのポートのいずれでも使用できるように注意する必要があります。

SPA ベイ マッピング



前述のように、物理ポートに対するサービスの提供は 2 つの FabricQ ASIC のどちらかが行います。ASIC へのポートの割り当ては静的に定義され、変更できません。また、SIP-800 PLIM には PLA ASIC が 2 つあります。最初のPLAサービスポート0、1および3、2番目のサービス2、4および5。

SIP-800 PLIM での単一 PLA の帯域幅容量は、約 24 Gbps です。単一の FabricQ ASIC のスイッチング キャパシティは、約 62 Mpps です。

ポート0、1、3またはポート2、4、5を入力すると、PLA(24 Gbps)の帯域幅容量が、全体のスループットを制限する3つのポートすべてで共有されます。これらのポート グループには、それぞれの単一の FabricQ によりサービスが提供されるため、ポート グループの最大のパケット レートは 62 Mpps になります。最適な帯域幅を達成するには、PLA で最高の効率性を実現するようにポートを使用することが最適です。

推奨される配置：

	SPA ベイ番号	SPA ベイ番号	SPA ベイ番号	SPA ベイ番号
オプション 1	0	1	4	5
オプション 2	1	0	3	4

カードに4つ以上のSPAを装着する場合は、上記のオプションのいずれかを実行して、2つのポートグループ(0、1、3、2、4、5)間のインターフェイスを分散することをお勧めします。 次に、次のSPAモジュールを、0、1、3、2、4、5ポートグループのいずれかのオープンポートに配置する必要があります。

DWDM XENPACK

リリース 3.2.2 以降では、DWDM XENPACK のインストールが可能となり、これにより、調整可能な光モジュールが提供されます。このような XENPACK モジュールの冷却要件として、インストールされたモジュール間に空きスロットを置く必要があります。また、単一の DWDM XENPACK モジュールがインストールされると、XENPACK モジュールが DWDM デバイスではない場合でも最大で 4 つのポートが使用できます。したがって、これは FabricQ での PLA からポートへのマッピングに直接に影響します。この要件には注意する必要があり、次の表で検討されています。

推奨される配置：

	光ファイバ ポート番号	光ファイバ ポート番号	光ファイバ ポート番号	光ファイバ ポート番号
オプション 1 または DWDM XENPACK	0	0	5	7
オプション 2	1	3	4	6

3.2.2 以降または 3.3 がインストールされている場合、FabricQ マッピングの変更は避けてください。このため、通常のモジュールと DWDM XENPACK のモジュールのどちらにも、さらに簡単な配置パターンを使用できます。

	光ファイバ ポート番号	光ファイバ ポート番号	光ファイバ ポート番号	光ファイバ ポート番号
オプション 1	0	0	4	6
オプション 2	1	3	5	7

カードに4つ以上の非DWDM XENPACKポートを装着する場合は、リストされているオプションのいずれかを実行して、2つのポートグループ(0 ~ 3、4 ~ 7)間で光インターフェイスモジュールを展開することをお勧めします。 0 〜 3 または 4 〜 7 のポート グループの空ポートのいずれかに、次の光インターフェイス モジュールを配置する必要があります。光インターフェイス モジュールの 5 番に 0 〜 3 のポート グループを使用する場合、光インターフェイス モジュールの 6 番

を 4 ～ 7 のポート グループに配置する必要があります。

詳細については、『DWDM XENPAK モジュール』を参照してください。

構成管理

IOS-XR のコンフィギュレーションは 2 段階の設定によって実行されますが、第 1 段階ではユーザがコンフィギュレーションを入力します。この段階では、CLI でコンフィギュレーション構文の確認だけが行われます。この段階で入力されるコンフィギュレーションが認識されるのは、CLI/XML などのコンフィギュレーション エージェント プロセスだけです。このコンフィギュレーションは sysdb サーバには書き込まれないので、確認は行われません。バックエンド アプリケーションには通知されず、この段階ではコンフィギュレーションにアクセスしたり、コンフィギュレーションを認識したりすることはできません。

第 2 段階では、ユーザがコンフィギュレーションを明示的にコミットします。この段階では、コンフィギュレーションが sysdb サーバに書き込まれ、バックエンド アプリケーションによってコンフィギュレーションが確認され、sysdb によって通知が生成されます。第 1 段階で入力されたコンフィギュレーションをコミットする前に、コンフィギュレーション セッションを中止することができます。したがって、第 1 段階で入力されたすべてのコンフィギュレーションが常に第 2 段階で常にコミットされると仮定することは危険です。

また、第 1 段階や第 2 段階では、ルータの動作や実行コンフィギュレーションを複数のユーザが修正できます。したがって、第 1 段階でのコンフィギュレーションや動作状態が実行されるルータのテストは、実際にコンフィギュレーションがコミットされる第 2 段階では有効ではない場合があります。

コンフィギュレーション ファイル システム

Configuration File System (CFS; コンフィギュレーション ファイル システム) は、ルータのコンフィギュレーションを格納するために使用される一連のファイルとディレクトリです。CFS はディレクトリ disk0:/config/ の下に格納されますが、これは RP で使用されるデフォルトのメディアです。CFS 内のファイルとディレクトリはルータには内部的なものでなので、ユーザによる修正や削除は避ける必要があります。これにより、コンフィギュレーションが失われたり破損したりして、サービスに影響が及ぶ可能性があります。

各コミット後、CFS にスタンバイ RP へのチェックポイントが行われます。このことは、フェールオーバー後にルータのコンフィギュレーション ファイルを保存するのに役立ちます。

ルータのブートアップ中、最後にアクティブであったコンフィギュレーションが CFS に格納されたコンフィギュレーション コミット データベースから適用されます。ルータによって自動的に実行されるため、各設定のコミット後にアクティブなコンフィギュレーションをユーザが手動で保存する必要はありません。

ブートアップ時でのコンフィギュレーションの適用中に、コンフィギュレーションを変更することは推奨できません。コンフィギュレーションの適用が完了していない場合は、ルータにログオンすると次のメッセージが表示されます。

System Configuration Process

このデバイスのスタートアップ コンフィギュレーションは、現在ロード中です。これには数分かかる場合があります。完了時には通知されます。このプロセスが完了するまでは、デバイスのコ

ンフィギュレーション変更は行わないでください。場合によっては、最後にアクティブであったコンフィギュレーションを CFS から復元するのではなく、ユーザ提供の ASCII コンフィギュレーション ファイルからルータのコンフィギュレーションを復元することが望ましい場合があります。

次の手順で、強制的にコンフィギュレーション ファイルを適用できます。

```
using the "-a" option with the boot command. This option forces
    the use of the specified file only for this boot.

    rommon>boot <image> -a <config-file-path>

    setting the value of "IOX_CONFIG_FILE" boot variable to the
    path of configuration file. This forces the use of the specified file
    for all boots while this variable is set.

    rommon>IOX_CONFIG_FILE=




    rommon>boot <image>
```

ルータ設定を復元する際に、1つ以上の設定項目が有効にならない可能性があります。失敗したすべてのコンフィギュレーションは CFS に保存され、次のリロードまで維持されます。

失敗したコンフィギュレーションを確認して、エラーを修正し、コンフィギュレーションを再度適用することができます。

これらは、ルータの起動時に失敗したコンフィギュレーションに対処するためのヒントの一部です。

IOX では、コンフィギュレーションを失敗したコンフィギュレーションとして次の 3 つの原因に分類できます。

1. 構文エラー：パーサーは構文エラーを生成します。これは通常、CLIコマンドに互換性がないことを示します。構文エラーを修正して、コンフィギュレーションを再度適用する必要があります。
2. セマンティックエラー：セマンティックエラーは、ルータの起動中にコンフィギュレーションマネージャが設定を復元したときに、バックエンドコンポーネントによって生成されます。コンフィギュレーションが実行コンフィギュレーションの一部として受け入れられることを保証する役割は、cfgmgr にはないことに注意することが重要です。cfgmgr は単なる中継者で、バックエンド コンポーネントによって生成されるセマンティックスの失敗を報告するだけです。失敗の理由を分析して失敗の原因を判断するのは、それぞれのバックエンドコンポーネントのオーナーの責任です。ユーザは設定モードから**describe <CLI commands>を実行**し、バックエンドコンポーネントの検証者の所有者を簡単に見つけることができます。たとえば、失敗したコンフィギュレーションとして router bgp 217 が表示される場合、describe コマンドによってコンポーネント ベリファイアが ipv4-bgp コンポーネントであることが表示されます。

```
RP/0/0/CPU0:router#configure terminal
RP/0/0/CPU0:router(config)#describe router bgp 217
The command is defined in bgpv4_cmds.parser

Node 0/0/CPU0 has file bgpv4_cmds.parser for boot package /gsr-os-mbi-3.3.87/mbi12000-rp.vm
```

```
from gsr-rout
Package:
    gsr-rout
        gsr-rout V3.3.87[Default]  Routing Package
        Vendor : Cisco Systems
        Desc   : Routing Package
        Build  : Built on Mon Apr  3 16:17:28 UTC 2006
        Source : By ena-view3 in /vws/vpr/mletchwo/cfgmgr_33_bugfix for c2.95.3-p8
        Card(s): RP, DRP, DRPSC
        Restart information:
          Default:
            parallel impacted processes restart
Component:
    ipv4-bgp V[fwd-33/66]  IPv4 Border Gateway Protocol (BGP)

File: bgpv4_cmds.parser


User needs ALL of the following taskids:

        bgp (READ WRITE)

It will take the following actions:
  Create/Set the configuration item:
         Path: gl/ip-bgp/0xd9/gbl/edm/ord_a/running
        Value: 0x1
Enter the submode:
        bgp
RP/0/0/CPU0:router(config)#
```

3. 適用エラー：設定は正常に確認され、実行コンフィギュレーションの一部として受け入れられましたが、バックエンドコンポーネントが何らかの理由で動作状態を更新できません。コンフィギュレーションは正しく確認されたので実行コンフィギュレーションに表示されますが、バックエンド動作エラーのために失敗した設定としても表示されます。describe コマンドを、コンポーネント適用のオーナーを検索するために、適用が失敗した CLI で再度実行できます。起動時に失敗したコンフィギュレーションを確認して再度適用するには、下記手順を実行します。リリース 3.2 の場合、失敗したコンフィギュレーションを再度適用するために、オペレータは次の手順を使用できます。オペレータは、ルータの起動時に保存された、失敗したコンフィギュレーションを確認するために show configuration failed startup コマンドを使用できます。オペレータは**show configuration failed startup noerror | file myfailed.cfg**コマンドを発行します。オペレータは、この失敗したコンフィギュレーションを再度適用するために、コンフィギュレーション モードに切り替えて、load/commit コマンドを使用する必要があります。

```
RP/0/0/CPU0:router(config)#load myfailed.cfg
Loading.
197 bytes parsed in 1 sec (191)bytes/sec
RP/0/0/CPU0:router(config)#commit
```

リリース 3.3 イメージの場合、オペレータは次のアップデートされた手順を使用できます。オペレータは、失敗したコンフィギュレーションを確認して再度適用するために、show configuration failed startup コマンドと load configuration failed startup コマンドを使用する必要があります。

```
RP/0/0/CPU0:router#show configuration failed startup
!! CONFIGURATION FAILED DUE TO SYNTAX/AUTHORIZATION ERRORS
telnet vrf default ipv4
server max-servers 5 interface POS0/7/0/3 router static
address-family ipv4 unicast
 0.0.0.0/0 172.18.189.1
```

```
!! CONFIGURATION FAILED DUE TO SEMANTIC ERRORS
router bgp 217 !!%
Process did not respond to sysmgr !
RP/0/0/CPU0:router#

RP/0/0/CPU0:router(config)#load configuration failed startup noerror
Loading.
263 bytes parsed in 1 sec (259)bytes/sec
RP/0/0/CPU0:mike3(config-bgp)#show configuration
Building configuration...
telnet vrf default ipv4 server max-servers 5 router static
address-family ipv4 unicast
  0.0.0.0/0 172.18.189.1
 !
!
router bgp 217
!
end

RP/0/0/CPU0:router(config-bgp)#commit
```

カーネル ダンプ機能

デフォルトでは、プロセスがクラッシュすると IOS-XR によってコア ダンプがハードディスクに書き込まれますが、カーネル自体がクラッシュした場合にはこれは書き込まれません。マルチシャーシシステムの場合、この機能は現在ラインカードシャーシ0でのみサポートされています。他のシャーシは将来のソフトウェアリリースのソフトウェアでサポートされる予定です。

標準コンフィギュレーションと admin モード コンフィギュレーションの両方で次の設定を使用して、RP と MSC のどちらのカーネル ダンプも有効にすることが推奨されます。

```
exception kernel memory kernel filepath harddisk:
exception dump-tftp-route port 0 host-address 10.0.2.1/16 destination 10.0.2.1 next-hop 10.0.2.1
tftp-srvr-addr 10.0.2.1
```

カーネル ダンプ設定

カーネル ダンプ設定により、カーネル クラッシュに対して次が発生します。

1. RP がクラッシュし、RP のハードディスクではディスクのルート ディレクトリ内にダンプが書き込まれる。
2. MSC がクラッシュすると、RP0 のハードディスクではディスクのルート ディレクトリ内にダンプが書き込まれる。

これは、non-stop forwarding (NSF) がルーティング プロトコルに設定されているため、RP フェールオーバーの回数には影響しません。コアの書き込み中のクラッシュの後、クラッシュした RP やラインカードが再び使用できるようになるには、さらに数分の時間が必要になる場合があります。

標準モードと admin モードの両方のコンフィギュレーションに対するこの設定の追加の例を次に示します。admin モードのコンフィギュレーションでは、DRP の使用が必要になることに注意してください。

次の出力に、カーネル ダンプ設定の例を示します。

```
RP/0/RP0/CPU0:crs1#configure
RP/0/RP0/CPU0:crs1(config)#exception kernel memory kernel filepat$
RP/0/RP0/CPU0:crs1(config)#exception dump-tftp-route port 0 host-$
RP/0/RP0/CPU0:crs1(config)#commit
RP/0/RP0/CPU0:crs1(config)#
RP/0/RP0/CPU0:crs1#admin
RP/0/RP0/CPU0:crs1(admin)#configure
Session                   Line        User      Date                      Lock
00000201-000bb0db-00000000  snmp        hfr-owne  Wed Apr  5 10:14:44 2006
RP/0/RP0/CPU0:crs1(admin-config)#exception kernel memory kernel f$
RP/0/RP0/CPU0:crs1(admin-config)#exception dump-tftp-route port 0$
RP/0/RP0/CPU0:crs1(admin-config)#commit
RP/0/RP0/CPU0:crs1(admin-config)#
RP/0/RP0/CPU0:crs1(admin)#
```

セキュリティ

LPTS

Local Packet Transport Services (LPTS) では、ローカルの宛先のパケットが処理されます。LPTS はさまざまなコンポーネントで構成されています。

1. 主なコンポーネントは、ポート調停プロセスと呼ばれています。たとえば、BGP や IS-IS など、異なるプロトコル プロセスからのソケット要求を受信し、これらのプロセスのすべてのバインディング情報を追跡します。たとえば、BGP プロセスがソケット ナンバー 179 を受信する場合、PA ではその情報を BGP プロセスから取得し、IFIB 内のそのプロセスにバインディングを割り当てます。
2. IFIB は、LPTS プロセスの別のコンポーネントです。どのプロセスが、どの特定のポートを受信しているかのバインディング情報を維持します。IFIB はポート調停プロセスによって生成され、ポート調停によって保持されます。IFIB によってこの情報の複数のサブセットが生成されます。1 番目のサブセットは IFIB のスライスです。このスライスは IPv4 プロトコルなどに関連付けることができます。次に、スライスは該当するフロー マネージャに送信され、パケットを適切なプロセスに転送するために IFIB スライスが使用されます。2 番目のサブセットは事前 IFIB であり、これにより、適切なプロセス (プロセスが 1 つだけ存在する場合) や適切なフロー マネージャへの LC によるパケットの転送が可能になります。
3. たとえば、BGP の複数のプロセスなど、ルックアップが簡単ではない場合、パケットをさらに配布するのにフロー マネージャが役立ちます。各フロー マネージャは、IFIB の単一または複数のスライスを持ち、IFIB のスライスに関連付けられた該当するプロセスにパケットを適切に転送します。
4. 宛先ポートにエントリが定義されていない場合、そのパケットは廃棄されるか、フロー マネージャに転送されます。ポートに関連付けられたポリシーが存在する場合、パケットは関連付けられたポートなしで転送されます。フロー マネージャは新しいセッション エントリの生成を支援します。

内部パケット転送の方法

レイヤ 2 (HDLC、PPP) フローとレイヤ 4 (ICMP/PING) フローの 2 種類のフロー、およびルーティング フローが存在します。

1. レイヤ2 HDLC/PPP：これらのパケットはプロトコルIDによって識別され、スプレイヤの

CPUキューに直接送信されます。レイヤ 2 プロトコル パケットには高い優先順位が付与され、CPU によって取得されて (Squid 経由) 処理されます。したがって、レイヤ 2 のキープアライブは CPU 経由の LC を介して直接応答されます。これによって、応答のために RP を経由する必要がなくなり、配布されたインターフェイス管理のテーマに沿って実行されます。

2. ICMP (レイヤ 4) パケットは LC 内で受信され、ルックアップを介して IFBI を通過し、スプレイヤ上の CPU キューに送信されます。これらのパケットは CPU に送信されて (Squid 経由) 処理されます。次に、応答はファブリックを経由して転送されるために、スプレイヤ出力キューを介して送信されます。これは別のアプリケーションにも情報が必要な場合です (ファブリックを介した複製)。 ファブリックを介した後、パケットは適切なスポンジとコントロール キューを経由して適切な出力 LC に宛てられます。
3. ルーティング フローは IFIB 内でルックアップされ、いずれかが制御パケット用に予約された出力シェーピング キュー (8000 のキュー) へ送信されます。これは非シェーピング キューであり、単にフルになるたびに処理が行われるだけです。 – 優先度が高い。次に、パケットは高い優先順位のキュー上のファブリックを介して、スポンジ上の一連の CPU に送信され (スプレイヤ上の Squid キューに類似)、適切なプロセス、フロー マネージャ、または実際のプロセスによって処理されます。応用は、出力ラインカード スポンジを経て、さらにライン カード経て戻されます。出力 LC スポンジには、制御パケットを処理するために特殊なキューが用意されています。スポンジ内のキューは、出力ポートごとに、高い優先順位、制御パケット、および低い優先順位のパケットに分けられます。
4. PSE には、レイヤ 4、レイヤ 2、およびルーティング パケットのレート制限用に設定された一連のポリシャが用意されています。これらは事前設定されていますが、将来的にはユーザによる設定が可能になります。

LPTS の最も一般的な問題の 1 つは、ルータに ping を実行する際に廃棄されるパケットです。通常、LPTS ポリシャでは、これらのパケットにレート制限が行われます。次は確認する場合です。

```
RP/0/RP0/CPU0:ss01-crs-1_P1#ping 192.168.3.14 size 8000 count 100
Type escape sequence to abort.
Sending 100, 8000-byte ICMP Echos to 192.168.3.14, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!
!!!!!!!!!!!!!!!.!!!!!!!!!!!!!
Success rate is 97 percent (97/100), round-trip min/avg/max = 1/2/5 ms
RP/0/RP0/CPU0:ss01-crs-1_P1#show lpts pifib hardware entry statistics location 0/5/CPU0 | excl
0/0

* - Vital; L4 - Layer4 Protocol; Intf - Interface;
DestAddr - Destination Fabric Address;
na - Not Applicable or Not Available

Local, Remote Address.Port          L4    Intf         DestAddr     Pkts/Drops
---------------------------------- ----- ------------ ------------ --- any
any   any          Punt       100/3
224.0.0.5 any                         any   PO0/5/1/0    0x3e         4/0
224.0.0.5 any                         any   PO0/5/1/1    0x3e         4/0
<further output elided>
```

IPSec

IP パケットは、本来はセキュリティで保護されていません。IP セキュリティは IP パケットを保護するために使用される方式です。CRS-1 の IP セキュリティはソフトウェア転送パスに実装されているため、IP セキュリティ セッションは RP/DRP で終端されます。CRS-1 につき合計 500 の IPsec セッションがサポートされます。この数は CPU の処理速度と割り当てられたリソース

によって異なります。これにはソフトウェア制限がなく、ローカルに発信され、RP でローカルに終端したトラフィックだけが IP セキュリティ処理の対象となります。この種類のトラフィックには IP セキュリティ トランスポート モードまたはトンネル モードのどちらも使用できますが、IP セキュリティ処理ではオーバーヘッドが小さいため、IP セキュリティ トランスポート モードが優先されます。

リリース 3.3.0 では、BGP over IPsec および OSPFv3 over IPsec の暗号化がサポートされています。

IP セキュリティの実装の詳細については、『Cisco IOS XR システム セキュリティ設定ガイド』を参照してください。

注：IPsecには暗号パイが必要です(たとえば、hfr-k9sec-p.pie-3.3.1)。

アウトオブバンド

コンソールと AUX アクセス

CRS-1 RP/SC では、アウトオブバンド管理の目的にコンソールと AUX ポートが使用でき、さらに IP 経由のアウトオブバンドには管理イーサネット ポートも使用できます。

各 RP/SCGE のコンソールと AUX ポート（シャーシにつき 2 つ）は、コンソール サーバに接続可能です。これは、単一のシャーシ システムに 4 つのコンソール ポートが必要であり、マルチシャーシ システムに 12 のポートと 2 つの追加ポート（Catalyst 6504-E のスーパーバイザ エンジンの場合）が必要であることを意味します。

AUX ポート接続は、IOS-XR カーネルへのアクセスを提供し、コンソール ポート経由ではシステム回復が不可能な場合にそれを実現するため、重要です。AUX ポートを介したアクセスは、システム上でローカルに定義されたユーザだけが使用でき、さらにユーザにルートシステムや Cisco サポート レベルのアクセス権が備わっている場合にのみ使用可能です。また、ユーザにはシークレット パスワードを定義しておく必要もあります。

仮想端末アクセス

Telnetおよびセキュアシェル(SSH)を使用して、vtyポート経由でCRS-1に到達できます。デフォルトではどちらも無効になっているので、ユーザが明示的にこれらを有効にする必要があります。

注：IPsecには暗号パイが必要です(たとえば、hfr-k9sec-p.pie-3.3.1)。

SSH を有効にするには、まず、この例で示すように RSA キーおよび DSA キーを生成します。

```
RP/0/RP1/CPU0:Crs-1#crypto key zeroize dsa
% Found no keys in configuration.
RP/0/RP1/CPU0:Crs-1#crypto key zeroize rsa
% Found no keys in configuration.

RP/0/RP1/CPU0:Crs-1#crypto key generate rsa general-keys
The name for the keys will be: the_default
  Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keypair.
  Choosing a key modulus greater than 512 may take a few minutes.
```

```
How many bits in the modulus [1024]:
Generating RSA keys ...
Done w/ crypto generate keypair
[OK]

RP/0/RP1/CPU0:Crs-1#crypto key generate dsa
The name for the keys will be: the_default
  Choose the size of your DSA key modulus. Modulus size can be 512, 768, or 1024 bits. Choosing
a key modulus
How many bits in the modulus [1024]:
Generating DSA keys ...
Done w/ crypto generate keypair
[OK]

!--- VTY access via SSH & telnet can be configured as shown here. vty-pool default 0 4 ssh
server ! line default secret cisco users group root-system users group cisco-support exec-
timeout 30 0 transport input telnet ssh ! ! telnet ipv4 server
```

関連情報

- ルータ サポート
- テクニカル サポートとドキュメント – Cisco Systems