Procedure ottimali operative per CRS-1 e IOS XR

Sommario

Introduzione

Questo documento aiuta a comprendere quanto segue:

- Processo e thread
- CRS-1 Fabric
- Piano di controllo
- Rommon e Monlib
- Physical Layer Interface Module (PLIM) e Modular Service Card (MSC)
- Gestione della configurazione
- Sicurezza
- Fuori banda
- Protocollo SNMP (Simple Network Management Protocol)

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza di Cisco IOS® XR.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- Software Cisco IOS XR
- CRS-1

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni nei suggerimenti tecnici.

Panoramica di Cisco IOS XR

Cisco IOS XR è progettato per la scalabilità. Il kernel è un'architettura Microkernel che fornisce solo servizi essenziali come la gestione dei processi, la pianificazione, i segnali e i timer. Tutti gli altri servizi, quali file system, driver, stack di protocolli e applicazioni, vengono considerati gestori di risorse ed eseguiti nello spazio utente protetto dalla memoria. Questi altri servizi possono essere aggiunti o rimossi in fase di esecuzione, a seconda della struttura del programma. L'ingombro di Microkernel è di soli 12 kb. Il Microkernel e il sistema operativo sottostante provengono da QNX Software Systems ed è chiamato Neutrino. QNX è specializzato nella progettazione di sistemi operativi in tempo reale. Il Microkernel è prioritario e l'utilità di pianificazione è basata sulla priorità. In questo modo la commutazione di contesto tra i processi è molto rapida e i thread con priorità più alta hanno sempre accesso alla CPU quando necessario. Questi sono alcuni dei vantaggi di cui Cisco IOS XR si avvale. Tuttavia, il vantaggio più grande è la progettazione ereditaria delle comunicazioni tra processi all'interno del nucleo dei sistemi operativi.

Neutrino è un sistema operativo di passaggio di messaggi e i messaggi sono il mezzo di base delle comunicazioni interprocesso tra tutti i thread. Quando un determinato server desidera fornire un servizio, crea un canale per lo scambio di messaggi. I client si collegano al canale Server mappando direttamente al descrittore di file rilevante per utilizzare il servizio. Tutte le comunicazioni tra client e server avvengono mediante lo stesso meccanismo. Si tratta di un enorme vantaggio per un super computer, che CRS-1 è. Tenere presente quanto segue quando si esegue un'operazione di lettura locale su un kernel UNIX standard:

- Interruzione software nel kernel.
- Il kernel viene inviato nel file system.
- Ricezione dati.

Nel caso remoto, considerare quanto segue:

- Interruzione software nel kernel.
- Il kernel invia NFS.
- NFS chiama il componente di rete.
- Il telecomando invia il componente di rete.
- Viene chiamato NFS.
- Il kernel invia il file system.

La semantica per la lettura locale e la lettura remota non sono uguali. Gli argomenti e i parametri per il blocco dei file e l'impostazione delle autorizzazioni sono diversi.

Si consideri il caso locale QNX:

- Interruzione software nel kernel.
- Il kernel esegue il passaggio dei messaggi nel file system.

Considerare il caso non locale:

- Interruzione software nel kernel.
- Il kernel passa in QNET, che è il meccanismo di trasporto IPC.
- QNET entra nel kernel.
- Il kernel invia il file system.

Tutte le semantiche relative al passaggio degli argomenti e ai parametri del file system sono identiche. Tutto è stato scollegato dall'interfaccia IPC, che permette al client e al server di essere completamente separati. Ciò significa che qualsiasi processo può essere eseguito in qualsiasi momento e in qualsiasi momento. Se un particolare processore di routing è troppo occupato a gestire le richieste, è possibile migrare facilmente tali servizi su una CPU diversa in esecuzione su un DRP. Supercomputer che esegue servizi diversi su CPU diverse distribuite su più nodi in grado di comunicare facilmente con qualsiasi altro nodo. L'infrastruttura è stata creata per fornire l'opportunità di scalare. Cisco ha utilizzato questo vantaggio e ha scritto software aggiuntivo che si aggancia alle principali operazioni del kernel di passaggio dei messaggi che consente al router CRS di scalare migliaia di nodi, dove un nodo, in questo caso una CPU, esegue un'istanza del sistema operativo, che si tratti di un processo di routing (RP), un processore di routing distribuito (DRP), una scheda MSC (Modular Services Card) o un processore di switch (SP).

Processo e thread

All'interno dei limiti di Cisco IOS XR, un processo è un'area di memoria protetta contenente uno o più thread. Dal punto di vista dei programmatori, i thread eseguono il lavoro e ognuno completa un percorso di esecuzione logico per eseguire un'attività specifica. La memoria richiesta dai thread

durante il flusso di esecuzione appartiene al processo in cui operano, protetta da qualsiasi altro thread di processo. Un thread è un'unità di esecuzione, con un contesto di esecuzione che include uno stack e registri. Un processo è un gruppo di thread che condividono uno spazio degli indirizzi virtuali, sebbene un processo possa contenere un singolo thread ma più spesso ne contenga altri. Se un altro thread di un processo diverso tenta di scrivere nella memoria del processo, il processo in questione viene interrotto. Se nel processo operano più thread, questi hanno accesso alla stessa memoria all'interno del processo e possono quindi sovrascrivere i dati di un altro thread. Completare i passaggi di una procedura per mantenere la sincronizzazione con le risorse in modo da impedire che il thread venga inserito nello stesso processo.

Un thread utilizza un oggetto denominato Mutual Exclusion (MUTEX) per garantire l'esclusione reciproca dei servizi. Il thread che dispone di MUTEX è il thread in grado di scrivere in una particolare area di memoria come esempio. Non è possibile utilizzare altri thread che non dispongono di MUTEX. Ci sono anche altri meccanismi per garantire la sincronizzazione con le risorse, e questi sono Semafori, Variabili condizionali o Condvars, Barriere e Sleepon. Questi servizi non vengono discussi in questa sede, ma forniscono servizi di sincronizzazione come parte dei loro compiti. Se si equiparano i principi qui illustrati a Cisco IOS, Cisco IOS è un singolo processo che opera su più thread, con tutti i thread che hanno accesso allo stesso spazio di memoria. Tuttavia, Cisco IOS chiama questi processi di thread.

Stati processo e thread

In Cisco IOS XR esistono server che forniscono i servizi e client che li utilizzano. Un determinato processo può avere più thread che forniscono lo stesso servizio. Un altro processo può avere un numero di client che potrebbero richiedere un particolare servizio in qualsiasi momento. L'accesso ai server non è sempre disponibile e, se un client richiede l'accesso a un servizio, rimane in tale posizione e attende che il server sia libero. In questo caso, il client viene considerato bloccato. Si tratta di un modello server client bloccante. È possibile che il client sia bloccato perché attende una risorsa, ad esempio un MUTEX, o perché il server non ha ancora risposto.

Utilizzare il comando **show process ospf** per controllare lo stato dei thread nel processo ospf:

```
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
```

Si noti che al processo ospf viene assegnato un ID processo (JID), ovvero 250. Tale ID non cambia mai su un router in esecuzione e in genere su una particolare versione di Cisco IOS XR. All'interno del processo OSPF sono presenti cinque thread ciascuno con il proprio ID thread (TID). Di seguito viene riportato lo spazio dello stack per ciascun thread, la priorità di ciascun thread e il relativo stato.

Passaggio sincrono dei messaggi

Si è già detto che QNX è un sistema operativo che passa il messaggio. Si tratta in realtà di un messaggio sincrono che passa il sistema operativo. Molti problemi del sistema operativo si riflettono nella messaggistica sincrona. Non si dice che il passaggio sincrono dei messaggi causi problemi, ma piuttosto che il sintomo del problema si riflette nel passaggio sincrono dei messaggi. Essendo sincrona, le informazioni sul ciclo di vita o sullo stato del ciclo di vita sono facilmente accessibili all'operatore CRS-1, che assiste nel processo di risoluzione dei problemi. Il messaggio ciclo di vita di passaggio è simile al seguente:

- Un server crea un canale messaggi.
- Un client si connette al canale di un server (analogamente a posix open).
- Un client invia un messaggio a un server (MsgSend) e attende una risposta e si blocca.
- Il server riceve (MsgReceive) un messaggio da un client, lo elabora e risponde al client.
- Il client sblocca ed elabora la risposta dal server.

Questo modello client-server di blocco è il messaggio sincrono passato. Questo significa che il client invia un messaggio e blocca. Il server riceve il messaggio, lo elabora, risponde al client e quindi lo sblocca. Questi sono i dettagli specifici:

- Attese del server nello stato RECEIVE.
- Il client invia un messaggio al server e diventa BLOCCATO.
- Il server riceve il messaggio e lo sblocca se è in attesa nello stato di ricezione.
- Il client passa allo stato REPLY.
- Il server passa allo stato RUNNING.
- Il server elabora il messaggio.
- Il server risponde al client.
- Il client sblocca.

Utilizzare il comando **show process** per verificare lo stato del client e del server.

```
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
```

Stati processo e processo bloccati

Per verificare lo stato del processo bloccato, usare il comando **show process** locked.

```
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#
```

Il passaggio dei messaggi sincronizzato consente di tenere traccia in modo semplice del ciclo di vita della comunicazione tra processi tra i diversi thread. In qualsiasi momento, un thread può trovarsi in uno stato specifico. Uno stato bloccato può essere sintomo di un problema. Questo non significa che se un thread è in stato bloccato, il problema si verifica, quindi non usare il comando **show process locked** e aprire una richiesta di assistenza con il supporto tecnico Cisco. Anche i thread bloccati sono molto normali.

Osservate l'output precedente. Se si controlla il primo thread dell'elenco, si noti che si tratta del ksh e la relativa risposta viene bloccata su devc-conaux. Il client, il ksh in questo caso, ha inviato un messaggio al processo devc-conaux, il server, che è devc-conaux, mantiene la risposta ksh bloccata finché non risponde. Ksh è la shell UNIX utilizzata sulla console o sulla porta AUX. Ksh attende l'input dalla console e, se non è presente alcun input perché l'operatore non sta digitando, rimane bloccato fino a quando non elabora l'input. Dopo l'elaborazione, ksh torna a rispondere bloccata su devc-conaux.

Si tratta di un comportamento normale che non illustra un problema. Il punto è che i thread bloccati sono normali e dipende dalla versione XR, dal tipo di sistema in uso, dalla configurazione e da chi esegue le operazioni che alterano l'output del comando **show process locked**. L'uso del comando **show process locked** è un buon modo per iniziare a risolvere i problemi relativi al tipo di sistema operativo. Se si verifica un problema, ad esempio se la CPU è alta, utilizzare il comando precedente per verificare se si verifica un problema al di fuori del normale.

Comprendere le normali caratteristiche del router in uso. In questo modo è possibile ottenere una base di riferimento da utilizzare come confronto per la risoluzione dei problemi relativi ai cicli di vita dei processi.

In qualsiasi momento, un thread può trovarsi in uno stato particolare. Questa tabella fornisce un elenco degli stati:

Se lo Stato è:	Il thread è:
MORTO	Morto. Il kernel è in attesa di rilasciare le risorse dei thread.
IN ESECUZIONE	Esecuzione attiva su una CPU
PRONTO	Non in esecuzione su una CPU ma pronto per l'esecuzione
INTERROTTO	Sospeso (segnale SIGSTOP)
INVIO	In attesa della ricezione di un messaggio da parte di un server
RICEVERE	In attesa che un client invii un messaggio
REPLY	In attesa della risposta di un server a un messaggio
PILA	In attesa dell'allocazione di altro stack
WAITPAGE	In attesa che Gestione processi risolva un errore di pagina

SIGSUSPEND	In attesa di un segnale
SIGWAITINFO	In attesa di un segnale
NANOSLEEP	Sospensione per un periodo di tempo
MUTEX	In attesa di acquisire un MUTEX
CONDVAR	In attesa della segnalazione di una variabile condizionale
UNISCI	In attesa del completamento di un altro thread
INTR	In attesa di un interrupt
SEM	In attesa di acquisire un semaforo

Processi importanti e relative funzioni

Cisco IOS XR dispone di molti processi. Questi sono alcuni importanti con le loro funzioni spiegate qui.

Monitor di sistema WatchDog (WDSysmon)

Questo servizio consente di rilevare blocchi del processo e condizioni di memoria insufficiente. La memoria insufficiente può essere il risultato di una perdita di memoria o di altre circostanze estranee. Il blocco può essere il risultato di diverse condizioni, ad esempio deadlock di processi, loop infiniti, blocchi del kernel o errori di pianificazione. In qualsiasi ambiente multithread il sistema può trovarsi in uno stato noto come "deadlock" o semplicemente "deadlock". Un deadlock può verificarsi quando uno o più thread non possono continuare a causa di un conflitto di risorse. Ad esempio, il thread A può inviare un messaggio al thread B mentre il thread B invia contemporaneamente un messaggio al thread A. Entrambi i thread attendono l'uno sull'altro e possono trovarsi nello stato di invio bloccato ed entrambi attendono per sempre. Si tratta di un caso semplice che coinvolge due thread, ma se un server è responsabile di una risorsa utilizzata da molti thread è bloccato su un altro thread, i molti thread che richiedono l'accesso a quella risorsa possono essere inviati bloccati in attesa sul server.

I deadlock possono verificarsi tra pochi thread, ma possono influire anche su altri thread. I deadlock vengono evitati da una buona progettazione dei programmi, ma indipendentemente da quanto magnificamente un programma è progettato e scritto. A volte una particolare sequenza di eventi dipendenti dai dati con intervalli specifici può causare un deadlock. Gli deadlock non sono sempre deterministici e generalmente sono molto difficili da riprodurre. WDSysmon ha molti thread con uno che viene eseguito alla massima priorità supportata da Neutrino, 63. L'esecuzione alla priorità 63 garantisce che il thread ottenga il tempo CPU in un ambiente di pianificazione preventiva basato sulla priorità. WDSysmon lavora con la capacità di sorveglianza hardware e controlla i processi software che cercano condizioni di blocco. Quando vengono rilevate tali condizioni, WDSysmon raccoglie ulteriori informazioni sulla condizione, può eseguire il coredump del processo o del kernel, scrivere nei syslog, eseguire script e terminare i processi con deadlock. A seconda della gravità del problema, può avviare uno switch del processore di routing per mantenere il funzionamento del sistema.

```
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
```

Il processo WDSysmon ha nove thread. Quattro vengono eseguiti con la priorità 10, gli altri quattro con le priorità 11, 19, 20 e 63. Quando un processo viene progettato, il programmatore considera attentamente la priorità che ogni thread all'interno del processo deve essere assegnato. Come descritto in precedenza, l'utilità di pianificazione è basata sulla priorità, il che significa che un thread con priorità più alta ha sempre la priorità più bassa. La priorità 63 è la massima priorità alla quale un thread può essere eseguito, in questo caso il thread 7. Il thread 7 è il thread di controllo, il thread che tiene traccia dei log della CPU. Deve essere eseguito con una priorità più alta rispetto agli altri thread controllati. In caso contrario, potrebbe non avere alcuna possibilità di essere eseguito, il che gli impedisce di eseguire i passaggi per i quali è stato progettato.

Netio

In Cisco IOS, esiste il concetto di switching rapido e switching di processo. La commutazione rapida utilizza il codice CEF e si verifica al momento dell'interrupt. La commutazione di processo utilizza ip_input, che è il codice di commutazione IP, ed è un processo pianificato. Sulle piattaforme più avanzate, la commutazione CEF viene eseguita nell'hardware e ip_input viene pianificato sulla CPU. L'equivalente di ip_input in Cisco IOS XR è Netio.

```
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
```

GSP (Group Services Process)

C'è bisogno di comunicazione in qualsiasi supercomputer con diverse migliaia di nodi che ciascuno esegue la propria istanza del kernel. In Internet, la comunicazione uno-a-molti viene effettuata in modo efficiente tramite protocolli multicasting. GSP è il protocollo multicasting interno utilizzato per l'IPC all'interno di CRS-1. GSP fornisce una comunicazione di gruppo affidabile uno a molti, senza connessione con la semantica asincrona. Questo permette al GSP di scalare fino alle migliaia di nodi.

```
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 Bulk Content Downloader

Il protocollo BCDL viene utilizzato per consentire il multicast affidabile dei dati su vari nodi, ad esempio RP e MSC. Usa l'SPG come mezzo di trasporto. BCDL garantisce la consegna dei messaggi. All'interno di BCDL ci sono un agente, un produttore e un consumatore. L'agente è il processo che comunica con il produttore al fine di recuperare e inserire i dati nel buffer prima dei multicast per i consumatori. Il produttore è il processo che produce i dati che tutti desiderano, e il consumatore è il processo interessato a ricevere i dati forniti dal produttore. Il protocollo BCDL viene utilizzato durante gli aggiornamenti del software Cisco IOS XR.

LWM (Lightweight Messaging)

LWM è una forma di messaggistica creata da Cisco e progettata per creare un livello di astrazione tra le applicazioni che comunicano tra processi e Neutrino, con l'obiettivo di indipendenza del sistema operativo e del livello di trasporto. Se Cisco desidera cambiare il fornitore del sistema operativo da QNX a qualcun altro, un livello di astrazione tra le funzioni rudimentali del sistema operativo sottostante aiuta a rimuovere la dipendenza dal sistema operativo e facilita il trasferimento a un altro sistema operativo. LWM fornisce il recapito sincrono garantito dei messaggi, che, come il passaggio nativo del messaggio Neutrino, provoca il blocco del mittente fino a quando il destinatario non risponde.

LWM fornisce anche la distribuzione asincrona dei messaggi tramite impulsi a 40 bit. I messaggi asincroni vengono inviati in modo asincrono, ovvero il messaggio viene accodato e il mittente non viene bloccato, ma non vengono ricevuti dal server in modo asincrono, bensì quando il server esegue il polling del successivo messaggio disponibile. LWM è strutturato come client/server. Il server crea un canale che gli dà l'orecchio per ascoltare i messaggi e si siede in un momento in cui il loop riceve un messaggio in ascolto sul canale, che ha appena creato. All'arrivo, il messaggio sblocca e ottiene l'identificativo del client, che equivale all'ID di ricezione del messaggio ricevuto. Il server quindi esegue alcune elaborazioni e successivamente risponde con un messaggio all'identificativo del client.

Sul lato client esegue una connessione al messaggio. Viene passato un identificatore a cui si connette e quindi invia un messaggio e viene bloccato. Al termine dell'elaborazione, il server risponde e il client viene sbloccato. Questo è praticamente lo stesso del passaggio del messaggio nativo Neutrino, quindi lo strato di astrazione è molto sottile.

LWM è progettato con un numero minimo di chiamate di sistema e di switch di contesto per prestazioni elevate ed è il metodo preferito di IPC nell'ambiente Cisco IOS XR.

Envmon

Al livello più fondamentale, il sistema di monitoraggio ambientale è responsabile dell'avviso quando i parametri fisici, ad esempio temperatura, tensione, velocità della ventola e così via, non rientrano nei limiti operativi e arresta l'hardware che si avvicina ai livelli critici in cui l'hardware potrebbe essere danneggiato. Controlla periodicamente ogni sensore hardware disponibile, confronta il valore misurato con le soglie specifiche della scheda e genera gli allarmi necessari per eseguire questa operazione. Processo persistente, avviato all'inizializzazione del sistema, che

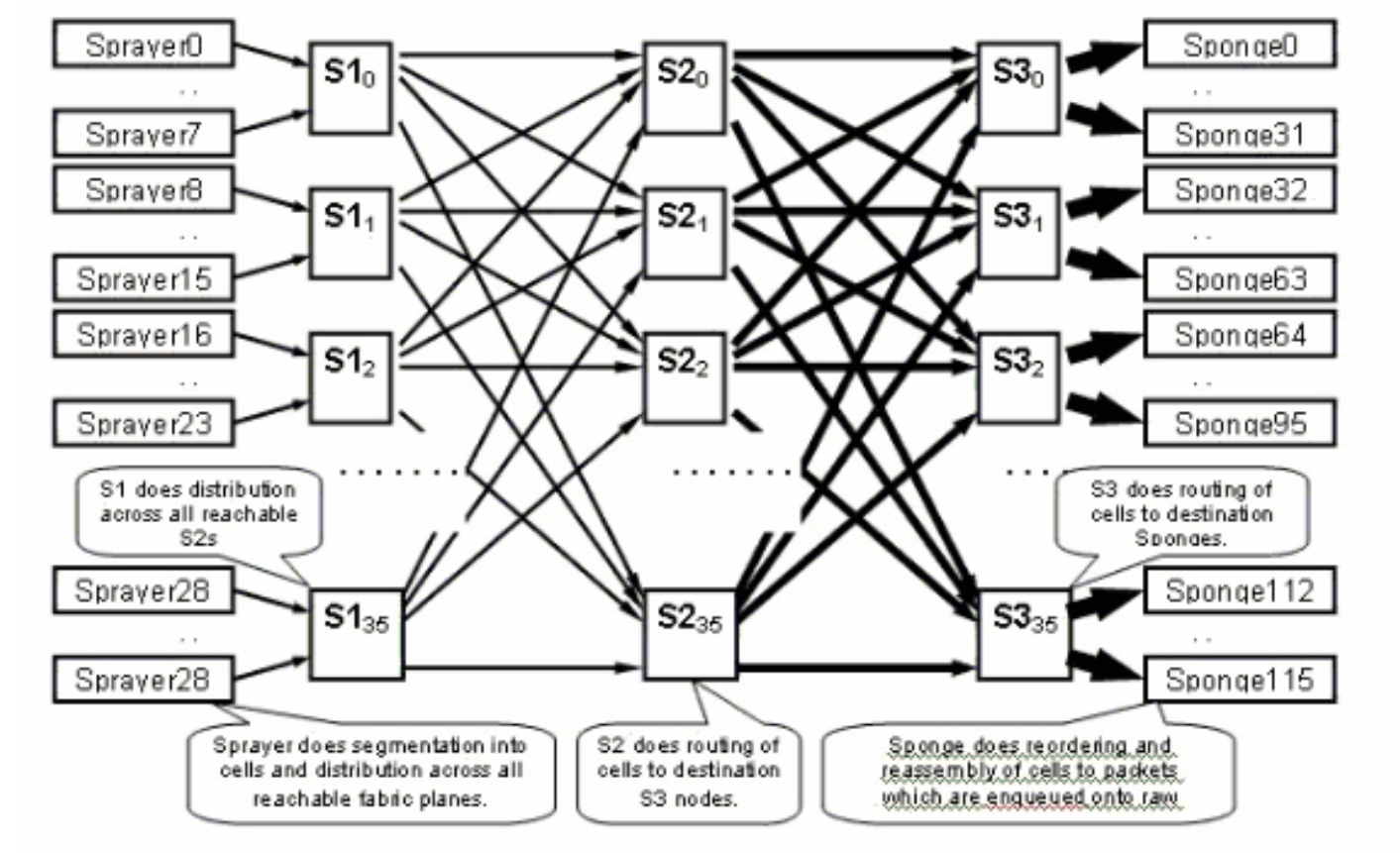
esegue periodicamente il polling di tutti i sensori hardware, ad esempio tensione, temperatura e velocità della ventola, nello chassis e fornisce questi dati ai client di gestione esterni. Inoltre, il processo periodico confronta le letture dei sensori con le soglie di allarme e pubblica gli avvisi ambientali nel database di sistema per l'azione successiva da parte di Fault Manager. Se le letture del sensore sono pericolosamente fuori portata, il processo di monitoraggio ambientale potrebbe provocare l'arresto della scheda.

CRS-1 Fabric Introduction

- Fabric multistage: topologia a 3 stadi Benes
- Routing dinamico all'interno del fabric per ridurre al minimo la congestione
- Basato su celle: Celle da 136 byte, payload dati da 120 byte
- Controllo del flusso per migliorare l'isolamento del traffico e ridurre al minimo i requisiti di buffering nel fabric
- Accelerazione da fase a fase
- Due cast di traffico supportati (Unicast e Multicast)
- Due priorità di traffico supportate per cast (alta e bassa)
- Supporto per gruppi multicast (FGID) fabric 1 MB
- Tolleranza ai guasti a costi contenuti: Ridondanza N+1 o N+k utilizzando piani fabric anziché 1+1 a un costo notevolmente aumentato

Quando si esegue in modalità a chassis singolo, gli elementi di base S1, S2 e S3 si trovano sulle stesse schede fabric. Questa scheda è comunemente indicata anche come **scheda S123**. In una configurazione multi-chassis, l'S2 è separato e si trova sullo chassis Fabric Card (FCC). Questa configurazione richiede due schede fabric per formare un piano, una scheda S2 e una scheda S13. Ogni MSC si connette a otto piani fabric per fornire ridondanza in modo che se si allentano uno o più piani, il fabric continua a trasmettere il traffico anche se il traffico aggregato, che può passare attraverso il fabric, è inferiore. Il CRS può ancora funzionare a velocità lineare per la maggior parte delle dimensioni di pacchetto con solo sette aerei. La contropressione viene esercitata sul tessuto su un aereo pari e dispari. Si consiglia di non eseguire un sistema con meno di due piani, su un piano pari e dispari. Una configurazione con meno di due piani non è supportata.

Il piano del fabric



Il diagramma precedente rappresenta un piano. Il diagramma va moltiplicato per otto. Ciò significa che l'asic dell'irroratore (ingressq) di un LC si collega a 8 S1 (1 S1 per piano). La S1 in ogni piano di tessuto si collega a 8 spruzzatori:

- 8 LC superiori dello chassis
- gli 8 LC inferiori

Esistono 16 S1 per chassis LC a 16 slot: 8 per i LC superiori (1 per piano) + 8 per i LC inferiori.

Su uno chassis a 16 slot singolo, una scheda fabric S123 è dotata di 2 S1, 2 S2 e 4 S3. Fa parte del calcolo della velocità del fabric. Il traffico è il doppio e può uscire dalla struttura in modo da consentire l'ingresso del traffico. Ci sono anche attualmente due spugne (fabricq) per LC rispetto a 1 spray. Ciò consente di eseguire il buffering sul LC in uscita quando più LC in entrata sovraccaricano un LC in uscita. Il LC in uscita è in grado di assorbire quella larghezza di banda extra dal tessuto.

Monitoraggio fabric

Disponibilità e connettività del piano:

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

Verificare se i piani stanno ricevendo/trasmettendo celle e se alcuni errori stanno aumentando:

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

Gli acronimi nel comando precedente:

- CE - Errore correggibile

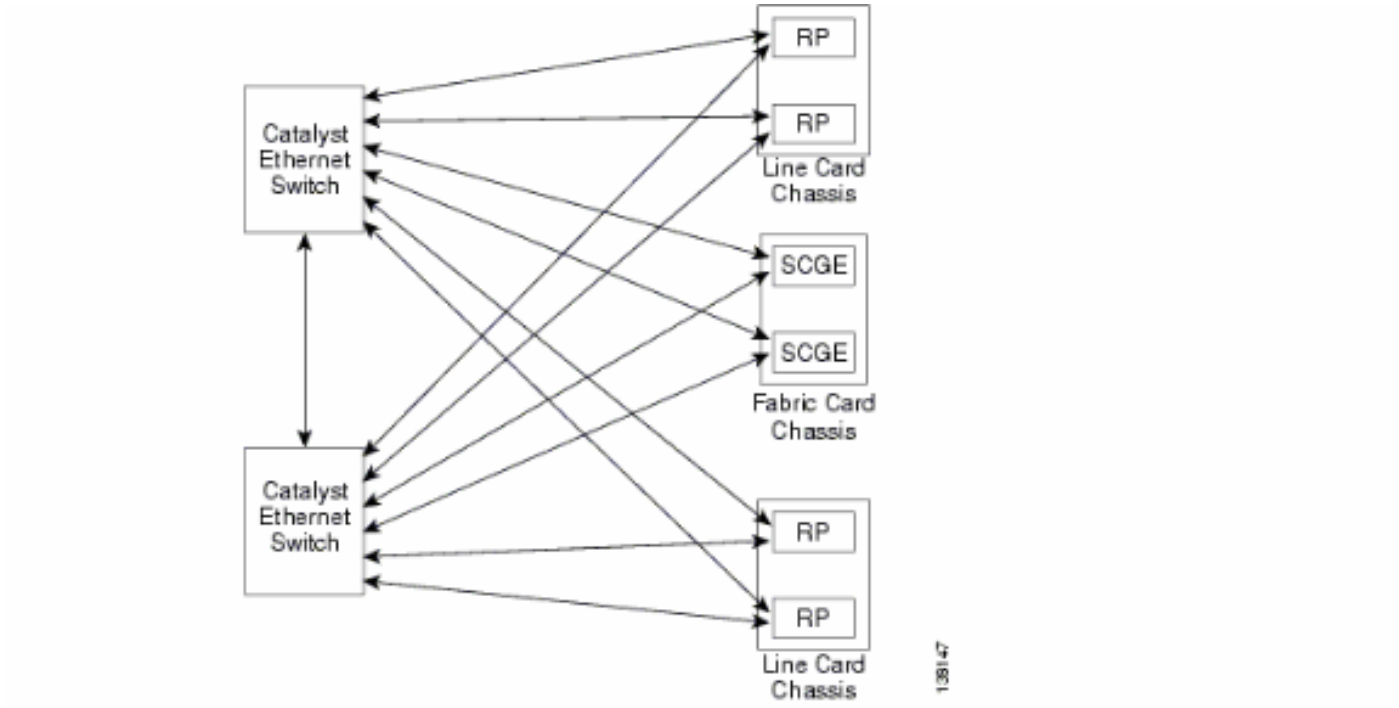
- UCE - Errore non correggibile
- PE - Errore di parità

Non preoccuparsi se notano alcuni errori, in quanto ciò può accadere all'avvio. I campi non devono aumentare in fase di esecuzione. Se lo sono, può essere un'indicazione di un problema nel tessuto. Per ottenere un'analisi stratificata degli errori per piano fabric, eseguire questo comando:

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

Panoramica del Control Plane

La connettività del control plane tra lo chassis della scheda di linea e lo chassis del fabric avviene attualmente tramite porte Gigabit Ethernet sulle schede RP (LCC) e SCGE (FCC). L'interconnessione tra le porte è fornita tramite una coppia di switch Catalyst 6500 che può essere connessa tramite due o più porte Gigabit Ethernet.



Configurazione Catalyst 6500

Questa è la configurazione consigliata per gli switch Catalyst utilizzati per il control plane multitchassis:

- Su tutte le porte viene utilizzata una singola VLAN.
- Tutte le porte vengono eseguite in modalità di accesso (senza trunking).
- Spanning-tree 802.1w/s è usato per la prevenzione dei loop.
- Per la connessione incrociata dei due switch vengono usati due o più collegamenti e il protocollo STP viene usato per impedire la formazione di loop. Il channeling non è consigliato.
- Le porte che si connettono a CRS-1 RP e SCGE utilizzano la modalità pre-standard poiché IOS-XR non supporta gli standard basati su 802.1s.

- Il protocollo UDLD deve essere abilitato sulle porte che si connettono tra gli switch e tra gli switch e il protocollo RP/SCGE.
- il protocollo UDLD è abilitato per impostazione predefinita sul CRS-1.

Per ulteriori informazioni su come configurare Catalyst 6500 in un sistema Multishelf, fare riferimento a Avvio del software Cisco IOS XR su un sistema Multishelf.

Gestione del Control Plane a chassis multipli

Lo chassis Catalyst 6504-E, che fornisce la connettività del control plane per il sistema multi-chassis, è configurato per i seguenti servizi di gestione:

- Gestione in-band tramite la porta gigabit 1/2, che si connette a uno switch LAN a ogni PoP. L'accesso è consentito solo per un piccolo intervallo di subnet e protocolli.
- L'ora NTP viene usata per impostare l'ora del sistema.
- Il syslog viene eseguito sugli host standard.
- Èpossibile abilitare il polling e le trap SNMP per le funzioni critiche.

Nota: non apportare modifiche al Catalyst in funzione. È consigliabile eseguire test preliminari su qualsiasi modifica pianificata e tale operazione deve essere eseguita durante un intervento di manutenzione.

Ecco un esempio della configurazione di gestione.

```
#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 e Monlib

Cisco monlib è un programma eseguibile memorizzato sul dispositivo e caricato nella RAM per l'esecuzione da ROMMON. ROMMON utilizza monlib per accedere ai file sul dispositivo. Le versioni di ROMMON possono essere aggiornate e devono essere eseguite seguendo la raccomandazione del supporto tecnico Cisco. L'ultima versione di ROMMON è la 1.40.

Istruzioni per l'aggiornamento

Attenersi alla seguente procedura:

1. Scaricare i file binari di ROMMON da Cisco CRS-1 ROMMON (solo utenti registrati).
2. Decomprimere il file TAR e copiare i 6 file BIN nella directory principale CRS di Disk0.
   ```
   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. Utilizzare il **comando show diag | inc ROM|NODE|PLIM** per visualizzare la versione rommon corrente.
   ```
   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. Per aggiornare ROMMON, accedere in modalità ADMIN e usare il comando **upgrade rommon a all disk0**.

```
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. Uscire dalla modalità ADMIN e immettere **show log | incluso "OK, ROMMON A"** e verificare che tutti i nodi siano stati aggiornati correttamente. Se si verifica un errore in uno dei nodi, tornare al passaggio 4 e riprogrammare.

```
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. Per aggiornare ROMMON, accedere in modalità ADMIN e usare il comando **upgrade rommon b all disk0**.

```
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. Uscire dalla modalità ADMIN e immettere **show log | incluso "OK, ROMMON B"** e verificare che tutti i nodi siano stati aggiornati correttamente. Se si verifica un errore in uno dei nodi, tornare al passaggio 4 e riprogrammare.

```
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. Il comando **upgrade** masterizza una sezione riservata speciale di bootflash con il nuovo ROMMON. Tuttavia, il nuovo ROMMON rimane inattivo finché la scheda non viene ricaricata. Quando si ricarica la scheda, il nuovo ROMMON è attivo. A tale scopo, reimpostare ogni nodo uno alla volta o reimpostare semplicemente l'intero router.

```
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. Utilizzare il **comando show diag | inc ROM|NODE|PLIM** per controllare la versione corrente di ROMMON.

```
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]
```

Nota: su CRS-8 e chassis Fabric, ROMMON imposta anche la velocità della ventola sulla velocità predefinita di 4000 RPM.

Panoramica su PLIM e MSC

Questo rappresenta il flusso di pacchetto sul router CRS-1, e questi termini vengono usati in modo intercambiabile:

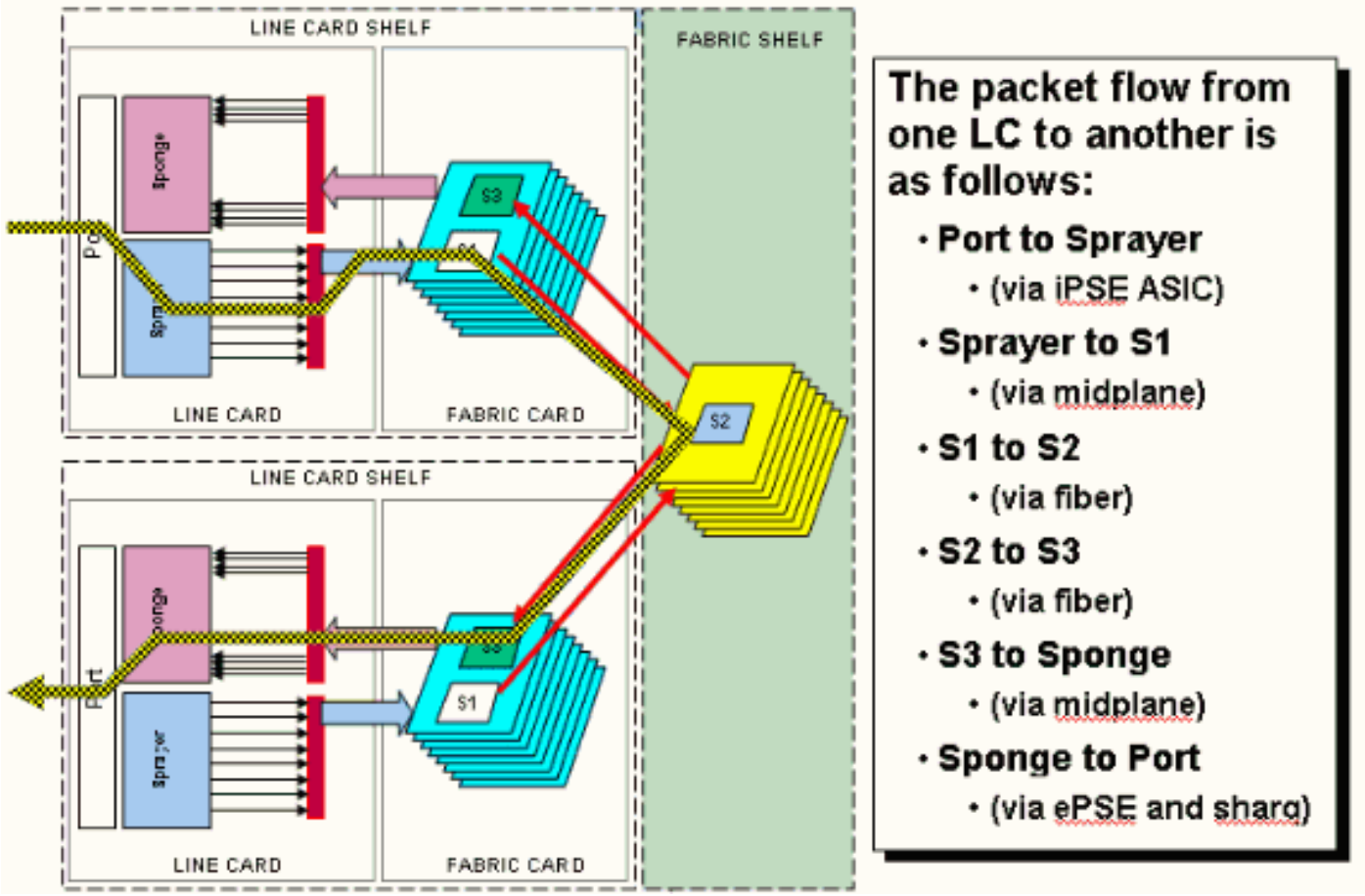
IngressQ ASIC è anche chiamato ASIC Sprayer.

FabricQ ASIC è anche denominato Sponge ASIC.

EgressQ ASIC è anche chiamato Sharq ASIC.

SPP è anche denominato PSE (Packet Switch Engine) ASIC.

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

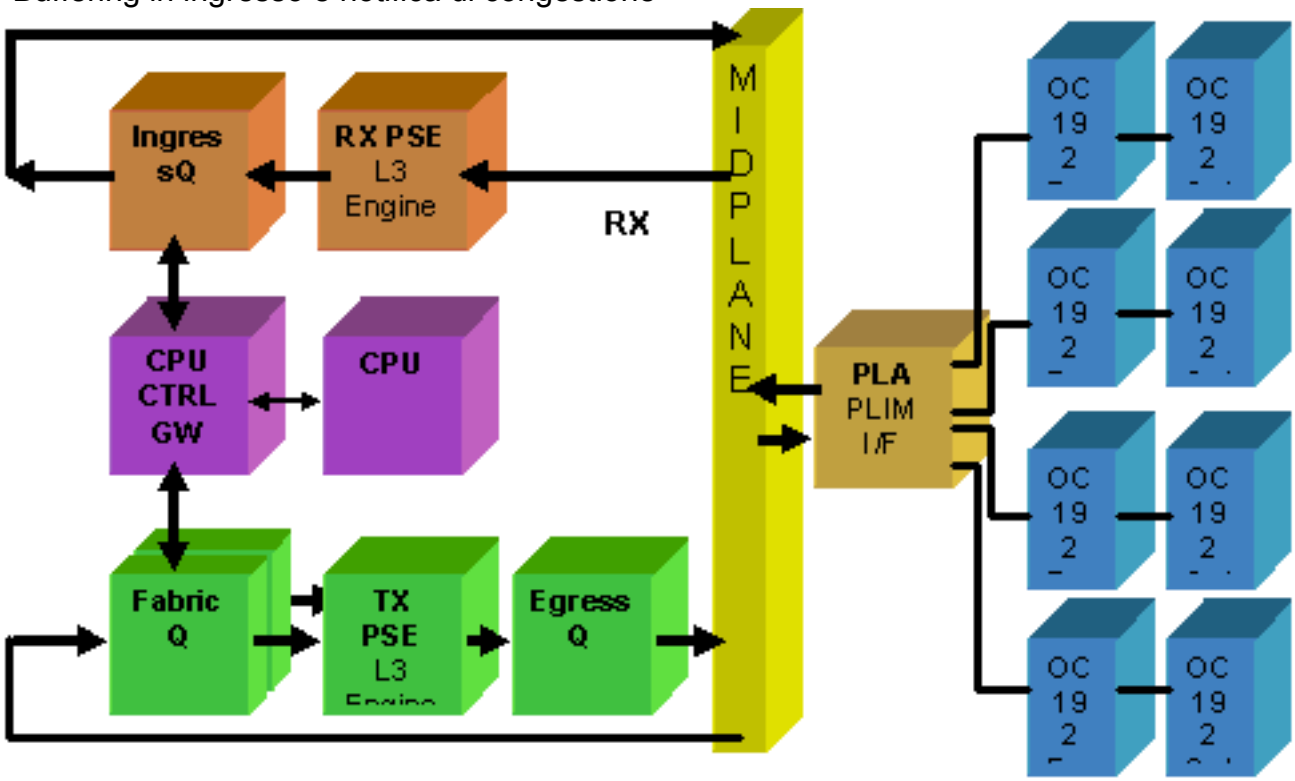


I pacchetti vengono ricevuti sul modulo PLIM (Physical Layer Interface Module).

Il PLIM contiene le interfacce fisiche per l'MSC con cui viene accoppiato. Il PLIM e MSC sono schede separate collegate tramite il backplane dello chassis. Di conseguenza, i tipi di interfaccia per un particolare MSC sono definiti dal tipo di PLIM con cui è accoppiato. A seconda del tipo di PLIM, la scheda contiene una serie di ASIC che forniscono il supporto fisico e la struttura delle interfacce. Lo scopo degli ASIC PLIM è fornire l'interfaccia tra MSC e le connessioni fisiche. Termina la fibra, esegue la conversione da luce a elettricità, termina il frame multimediale SDH/Sonet/Ethernet/HDLC/PPP, controlla il CRC, aggiunge alcune informazioni di controllo chiamate Buffer Header e inoltra i bit che rimangono sul MSC. Il PLIM non genera/trasferisce i pacchetti keepalive HDLC o PPP. Questi vengono gestiti dalla CPU dell'MSC.

Il PLIM fornisce anche le seguenti funzioni:

- Filtraggio MAC per 1/10 Gigabit Ethernet
- Accounting MAC in ingresso/uscita per 1/10 Gigabit Ethernet
- Filtro VLAN per 1/10 Gigabit Ethernet
- Accounting VLAN per 1/10 Gigabit Ethernet
- Buffering in ingresso e notifica di congestione



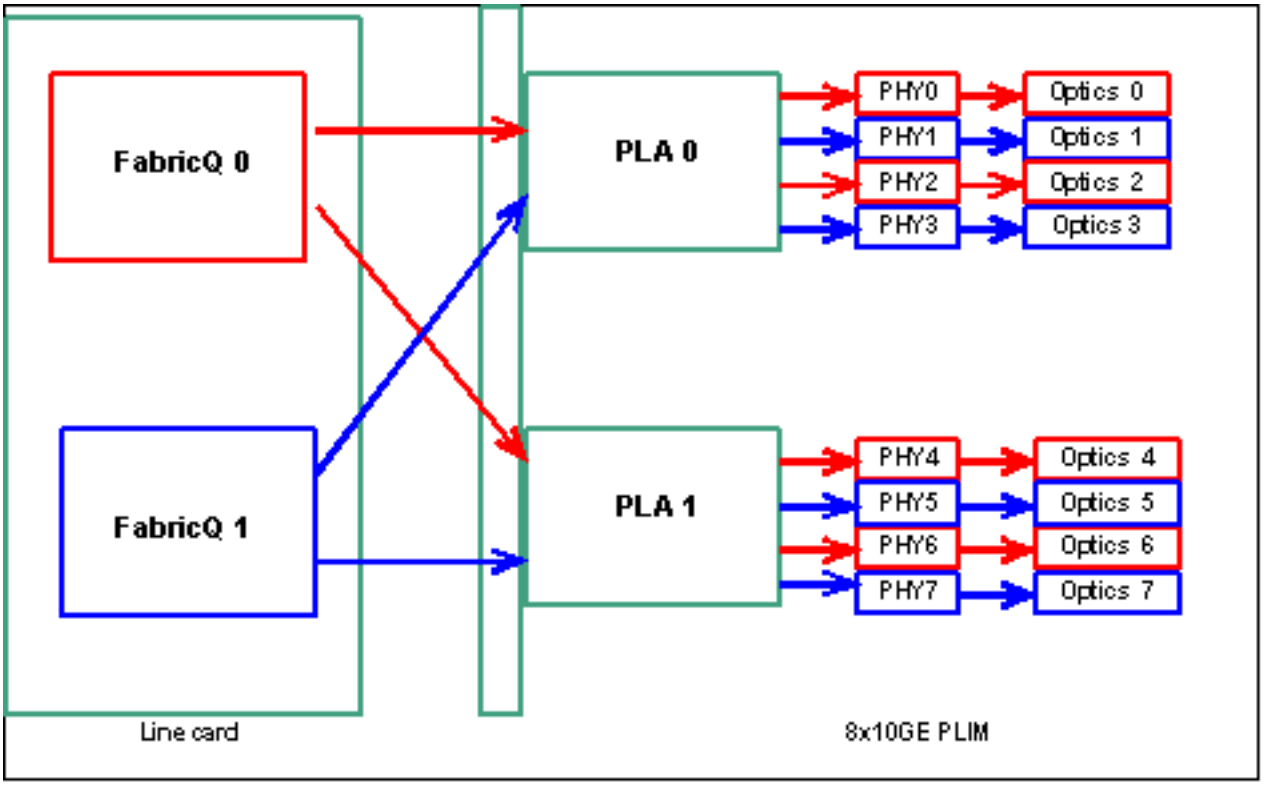
Sottoscrizione in eccesso PLIM

10GE PLIM

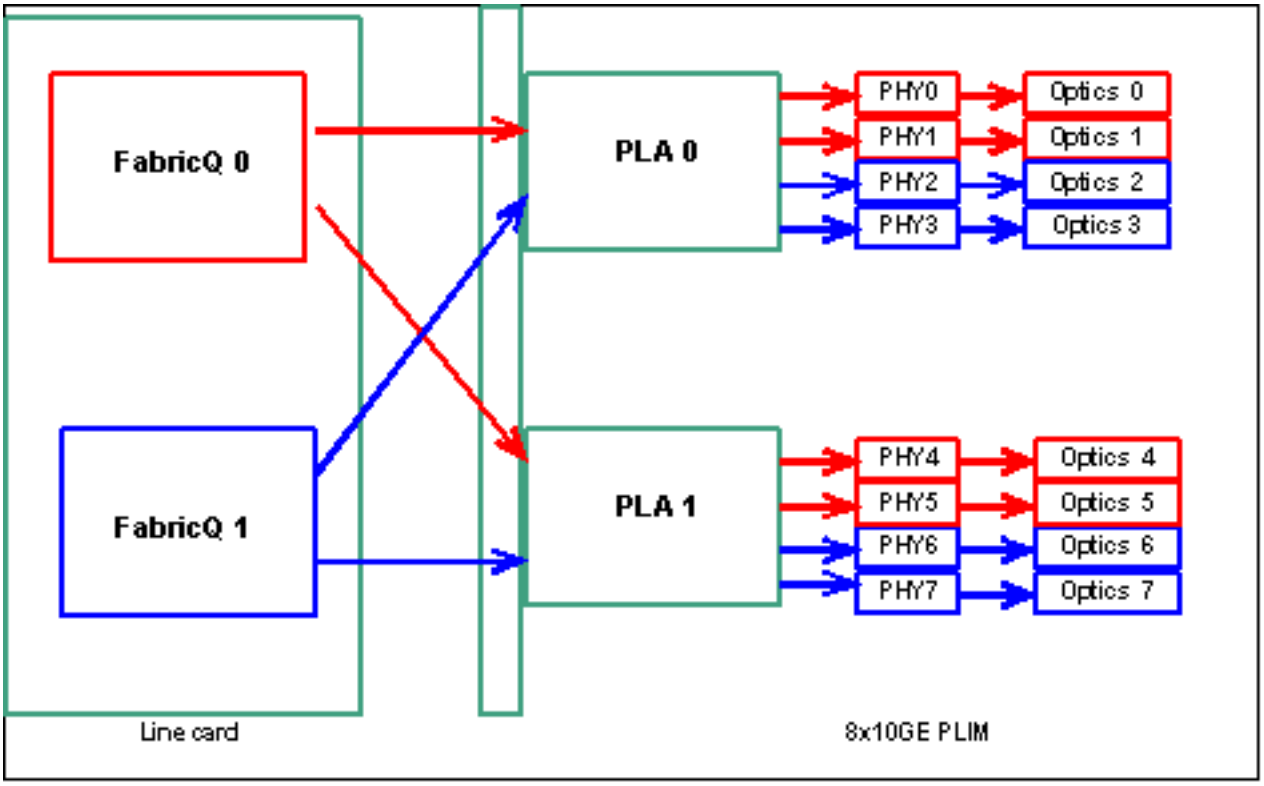
Il PLIM 8 X 10G consente di terminare circa 80 Gb/s di traffico, mentre la capacità di inoltro del MSC è al massimo di 40 Gb/s. Se tutte le porte disponibili sul PLIM sono popolate, si verifica una sovrascrittura e la modellazione QoS diventa estremamente importante per garantire che il traffico premium non venga inavvertitamente interrotto. Per alcuni, la sottoscrizione eccessiva non è un'opzione e deve essere evitata. A tale scopo, è necessario utilizzare solo quattro delle otto porte. Inoltre, è necessario assicurarsi che la larghezza di banda ottimale all'interno di MSC e PLIM sia disponibile per ciascuna delle quattro porte.

Nota: la mappatura delle porte viene modificata a partire dalla release 3.2.2. Vedere questi diagrammi.

Mappatura delle porte fino alla release 3.2.1



Mappatura delle porte a partire dalla release 3.2.2



Come accennato in precedenza, le porte fisiche sono servite da uno dei due ASIC FabricQ. L'assegnazione delle porte all'ASIC è definita in modo statico e non può essere modificata. Inoltre, il PLIM 8 X 10G dispone di due ASIC PLA. Il primo servizio PLA porta da 0 a 3, il secondo servizi da 4 a 7. La capacità di larghezza di banda di un singolo PLA sul PLIM 8 X 10G è di circa 24 Gbps. La capacità di switching di un singolo ASIC FabricQ è di circa 62 Mp/s.

Se si popolano le porte da 0 a 3 o da 4 a 7, la capacità della larghezza di banda del PLA (24 Gbps) viene condivisa tra tutte e quattro le porte che limitano il throughput complessivo. Se si popolano le porte 0,2,4 e 6 (fino alla 3.2.1) o 0,1,4 e 5 (3.2.2 in avanti) poiché tutte queste porte sono servite dall'ASIC FabricQ, la cui capacità di switching è di 62 Mp/s, ancora una volta, il che limita la capacità di throughput.

Per ottenere prestazioni ottimali, è consigliabile utilizzare le porte in modo da ottenere la massima efficienza sia dei PLA che degli ASIC FabricQ.

SIP-800/SPA

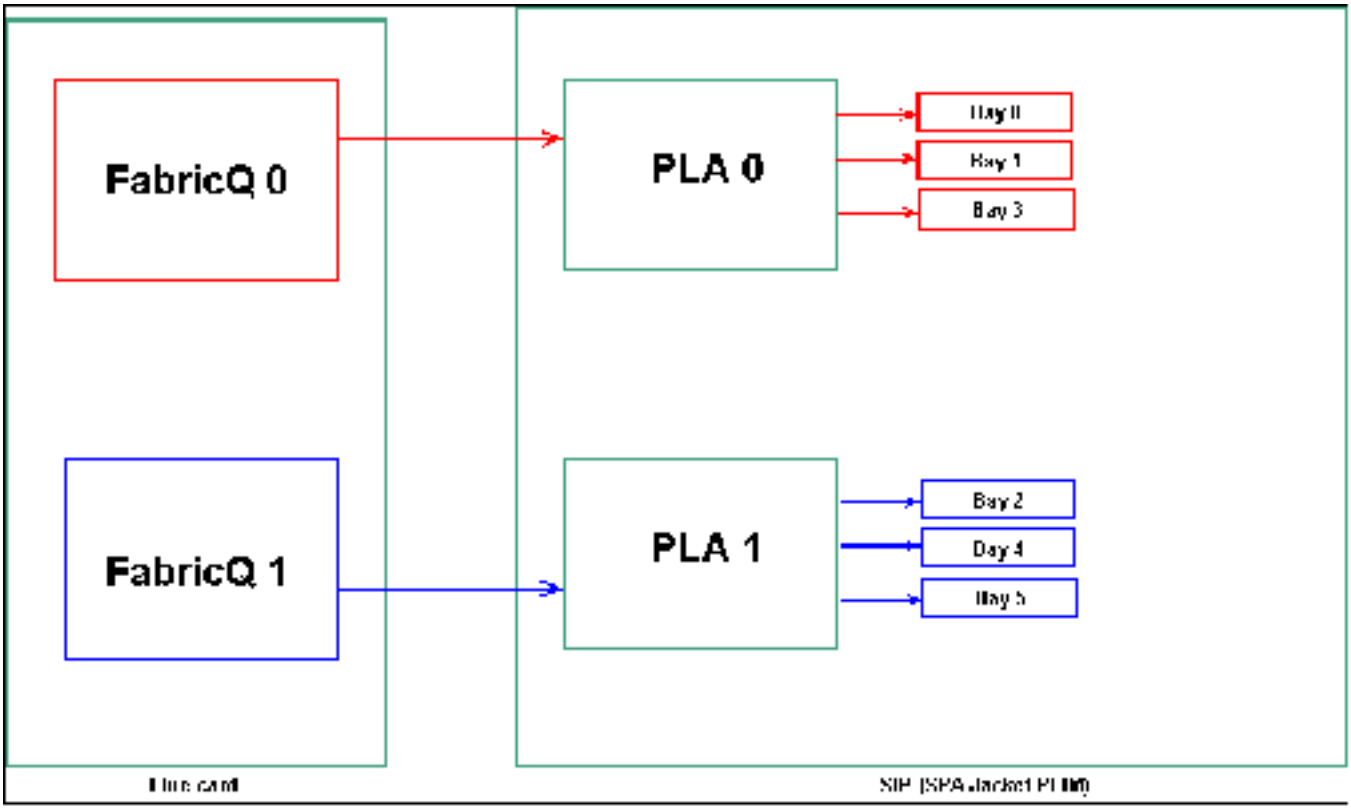
Il SIP-800 PLIM consente di funzionare con schede di interfaccia modulari note come SPA (Service Port Adapter). Il SIP-800 fornisce 6 alloggiamenti SPA con una capacità di interfaccia teorica di 60 Gbps. La capacità di inoltro dell'MSC è al massimo di 40 Gb/s. Se tutti gli alloggiamenti del SIP-800 dovessero essere popolati, a seconda del tipo di SPA, potrebbe verificarsi una sovrassegnazione e la modellazione QoS diventerebbe estremamente importante per garantire che il traffico premium non venga inavvertitamente scartato.

Nota: l'oversubscription non è supportato con le interfacce POS. Tuttavia, il posizionamento del

POS SPA da 10 Gb deve essere appropriato per garantire la corretta capacità di throughput fornita. L'SPA Ethernet da 10 Gb è supportata solo in IOS-XR versione 3.4. Questa SPA offre funzionalità di oversubscription.

Per alcuni, la sottoscrizione eccessiva non è un'opzione e deve essere evitata. A tale scopo, utilizzare solo quattro dei sei alloggiamenti. Inoltre, è necessario prestare attenzione per garantire che la larghezza di banda ottimale all'interno delle porte MSC e PLIM sia disponibile per ognuna delle quattro porte.

Mappatura dell'alloggiamento SPA



Come accennato in precedenza, le porte fisiche sono servite da uno dei due ASIC FabricQ. L'assegnazione delle porte all'ASIC è definita in modo statico e non può essere modificata. Inoltre, il SIP-800 PLIM ha due ASIC PLA. Le porte dei primi servizi PLA sono 0,1 e 3, mentre le porte dei secondi servizi sono 2, 4 e 5.

La capacità di larghezza di banda di un singolo PLA sul SIP-800 PLIM è di circa 24 Gbps. La capacità di switching di un singolo ASIC FabricQ è di circa 62 Mp/s.

Se si popolano le porte 0, 1 e 3 o le porte 2, 4 e 5, la capacità della larghezza di banda del PLA (24 Gbps) viene condivisa tra tutte e tre le porte che limitano il throughput complessivo. Poiché ogni FabricQ gestisce questi gruppi di porte, la velocità massima del pacchetto del gruppo di porte è di 62 Mp/s. È preferibile utilizzare le porte in modo da ottenere la massima efficienza degli ACL per ottenere la massima larghezza di banda.

Posizionamento consigliato:

	N. alloggiamento SPA	N. alloggiamento SPA	N. alloggiamento SPA	N. alloggiamento SPA
Opzione 1	0	1	4	5
Opzione 2	1	2	3	4

Se si desidera popolare la scheda con più di quattro SPA, si consiglia di completare una delle opzioni elencate in precedenza, che consente di distribuire le interfacce tra i due gruppi di porte (0,1 & 3 & 2,4 & 5). I successivi moduli SPA devono essere posizionati in una delle porte aperte dei gruppi di porte 0,1 & 3 & 2,4 & 5.

DWDM XENPACK

A partire dalla release 3.2.2, è possibile installare i DWDM XENPACK e fornire moduli ottici **regolabili**. I requisiti di raffreddamento di tali moduli XENPACK richiedono l'esistenza di uno slot vuoto tra i moduli installati. Inoltre, se è installato un singolo modulo DWDM XENPACK, è possibile utilizzare un massimo di quattro porte, anche se i moduli XENPACK non sono dispositivi DWDM. Questo ha quindi un impatto diretto sulla mappatura da FabricQ a PLA a Port. È necessario prestare attenzione a tale requisito e nella tabella che segue è indicato.

Posizionamento consigliato:

	N. porta ottica	N. porta ottica	N. porta ottica	N. porta ottica
Opzione 1 o DWDM XENPACK	0	2	5	7
Opzione 2	1	3	4	6

Per un'installazione 3.2.2 o successiva o 3.3, evitare la modifica del mapping FabricQ. Un modello di posizionamento più semplice può quindi essere utilizzato sia per i moduli regolari che per i moduli DWDM XENPACK.

	N. porta ottica	N. porta ottica	N. porta ottica	N. porta ottica
Opzione 1	0	2	4	6
Opzione 2	1	3	5	7

Se si desidera popolare la scheda con più di quattro porte XENPACK non DWDM, si consiglia di completare una delle opzioni elencate, che consente di distribuire i moduli di interfaccia ottica tra i due gruppi di porte (0-3 e 4-7). Quindi, i moduli di interfaccia ottica successivi devono essere posizionati in una delle porte aperte nei gruppi di porte 0-3 o 4-7. Se si utilizza il gruppo di porte 0-3 per il modulo di interfaccia ottica 5, i moduli di interfaccia ottica 6 devono essere inseriti nel gruppo di porte 4-7.

Per ulteriori informazioni, fare riferimento a DWDM XENPAK Module.

Gestione della configurazione

La configurazione in IOS-XR viene eseguita tramite una configurazione in due fasi, immessa dall'utente nella prima fase. Questa è la fase in cui solo la sintassi di configurazione viene controllata dalla CLI. La configurazione immessa in questa fase è nota solo al processo dell'agente di configurazione, ad esempio CLI/XML. La configurazione non viene verificata perché non viene scritta sul server sysdb. In questa fase, l'applicazione back-end non riceve alcuna notifica e non può accedere alla configurazione né esserne a conoscenza.

Nella seconda fase, la configurazione viene esplicitamente sottoposta a commit dall'utente. In questa fase la configurazione viene scritta sul server sysdb, le applicazioni back-end verificano le configurazioni e le notifiche generate da sysdb. È possibile interrompere una sessione di configurazione prima di eseguire il commit della configurazione immessa nella prima fase. Pertanto, non è sicuro presumere che tutte le configurazioni immesse nella prima fase vengano sempre eseguite nella seconda fase.

Inoltre, il funzionamento e/o la configurazione corrente del router possono essere modificati da più utenti durante la prima e la seconda fase. Pertanto, qualsiasi test del router che esegue la configurazione e/o lo stato operativo nella prima fase potrebbe non essere valido nella seconda fase in cui la configurazione è stata effettivamente eseguita.

File system di configurazione

Il file system di configurazione (CFS, Configuration File System) è un gruppo di file e directory utilizzati per memorizzare la configurazione del router. Il file CFS è memorizzato nella directory disk0:/config/, che è il supporto predefinito utilizzato nell'RP. I file e le directory in CFS sono interni al router e non devono mai essere modificati o rimossi dall'utente. Ciò può causare la perdita o il danneggiamento della configurazione e incidere sul servizio.

Il CFS viene selezionato per lo standby-RP dopo ogni commit. In questo modo è possibile preservare il file di configurazione del router dopo un failover.

Durante l'avvio del router, l'ultima configurazione attiva viene applicata dal database di commit della configurazione archiviato in CFS. non è necessario che l'utente salvi manualmente la configurazione attiva dopo ogni commit di configurazione, in quanto questa operazione viene eseguita automaticamente dal router.

Non è consigliabile apportare modifiche alla configurazione durante l'applicazione della configurazione durante l'avvio. Se l'applicazione di configurazione non è completa, quando si accede al router viene visualizzato questo messaggio:

Processo di configurazione del sistema

Èin corso il caricamento della configurazione di avvio del dispositivo. L'operazione può richiedere alcuni minuti. L'utente viene informato del completamento. Non tentare di riconfigurare il dispositivo fino al completamento del processo. In alcuni rari casi, potrebbe essere opportuno ripristinare la configurazione del router da un file di configurazione ASCII fornito dall'utente anziché ripristinare l'ultima configurazione attiva da CFS.

Per forzare l'applicazione di un file di configurazione, è possibile:

```
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>
```

Quando si ripristina la configurazione del router, uno o più elementi di configurazione potrebbero non avere effetto. Tutte le configurazioni non riuscite vengono salvate nel file CFS e mantenute fino al successivo caricamento.

È possibile sfogliare la configurazione non riuscita, correggere gli errori e riapplicare la configurazione.

Di seguito vengono riportati alcuni suggerimenti per risolvere i problemi di configurazione durante l'avvio del router.

In IOX è possibile classificare una configurazione come non riuscita per tre motivi:

1. Errori di sintassi: il parser genera errori di sintassi che in genere indicano un'incompatibilità con i comandi CLI. Correggere gli errori di sintassi e riapplicare la configurazione.
2. Errori semantici - Gli errori semantici vengono generati dai componenti back-end quando Gestione configurazione ripristina la configurazione durante l'avvio del router. È importante notare che cfgmgr non è responsabile dell'accettazione della configurazione come parte della configurazione in esecuzione. Cfgmgr è semplicemente un **intermediario** e segnala solo gli errori semantici generati dai componenti back-end. Spetta a ciascun proprietario del componente back-end analizzare la causa dell'errore e determinarne la causa. Gli utenti possono eseguire i comandi **description <CLI>** dalla modalità di configurazione per trovare facilmente il proprietario del verificatore del componente back-end. Ad esempio, se il **router bgp 217** mostra una configurazione non riuscita, il comando **description** mostra che il verificatore componente è il componente ipv4-bgp.

```
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. Errori di applicazione: la configurazione è stata verificata e accettata come parte della configurazione in esecuzione, ma per qualche motivo il componente back-end non è in grado di aggiornare il proprio stato operativo. La configurazione viene mostrata sia nella configurazione in esecuzione, poiché è stata verificata correttamente, sia come configurazione non riuscita a causa dell'errore operativo del back-end. per individuare il proprietario dell'applicazione del componente, è possibile eseguire nuovamente il comando **description** sulla CLI che non è stato possibile applicare.Completare questi passaggi per sfogliare e riapplicare la configurazione non riuscita durante gli operatori di avvio:Per R3.2 gli operatori possono utilizzare questa procedura per riapplicare la configurazione non riuscita:Gli operatori possono usare il comando **show configuration failed startup** per sfogliare la configurazione errata salvata durante l'avvio del router.Gli operatori devono eseguire il comando **show configuration failed startup noerror | file myfailed.cfg** per salvare in un file la configurazione di avvio non riuscito.Gli operatori devono passare alla modalità di **configurazione** e utilizzare i comandi **load/commit** per riapplicare la configurazione non riuscita:

```
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
```

Per le immagini R3.3 gli operatori possono utilizzare questa procedura aggiornata:Gli operatori devono utilizzare il comando **show configuration failed startup** e il comando **load configuration failed startup** per individuare e riapplicare eventuali configurazioni non riuscite.

```
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
```

Dumper kernel

Per impostazione predefinita, IOS-XR scrive un dump di core sul disco rigido in caso di arresto anomalo di un processo, ma non in caso di arresto anomalo del kernel stesso. Si noti che per un sistema a chassis multiplo questa funzionalità è attualmente supportata solo per lo chassis con scheda di linea 0. L'altro chassis è supportato in una versione futura del software.

Si consiglia di abilitare i dump del kernel sia per gli RP che per gli MSC utilizzando questa configurazione sia nella configurazione standard che in quella amministrativa:

```
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
```

Configurazione dump kernel

Ciò si verifica in caso di arresto anomalo del kernel:

1. Un RP si blocca e un dump viene scritto sul disco rigido di quel RP nella directory radice del disco.
2. In caso di arresto anomalo di un MSC, viene scritto un dump sul disco rigido di RP0 nella directory principale del disco.

Ciò non ha alcun impatto sui tempi di failover RP poiché per i protocolli di routing è configurato l'inoltro non-stop (NSF). Può essere necessario qualche minuto in più perché l'RP o la scheda di linea bloccata diventi nuovamente disponibile dopo un guasto mentre scrive il core.

Di seguito è riportato un esempio di aggiunta di questa configurazione sia alla configurazione standard che a quella in modalità di amministrazione. La configurazione della modalità di amministrazione richiede l'utilizzo di DRP.

Questo output mostra un esempio di configurazione del dump del kernel:

```
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)#
```

Sicurezza

LPT

Local Packet Transport Services (LPTS) gestisce i pacchetti destinati localmente. LPTS è costituito da vari componenti diversi.

1. Il processo principale è denominato processo di arbitraggio della porta. Ascolta le richieste socket provenienti da diversi processi di protocollo, ad esempio BGP, IS-IS e tiene traccia di tutte le informazioni di binding per tali processi. Ad esempio, se un processo BGP resta in ascolto al socket numero 179, l'amministratore ottiene tali informazioni dai processi BGP e quindi assegna un binding a tale processo in un IFIB.
2. L'IFIB è un altro componente del processo LPTS. Consente di mantenere una directory in cui un processo è in ascolto di un binding di porta specifico. L'IFIB viene generato dalla procedura Port Arbitrator e viene conservato con l'arbitro di porta. Vengono quindi generati più sottoinsiemi di queste informazioni.Il primo sottoinsieme è la sezione a dell'IFIB. Questa slice può essere associata al protocollo IPv4 e così via. Le slice vengono quindi inviate ai gestori di flusso appropriati, che utilizzano la slice IFIB per inoltrare il pacchetto al processo appropriato.Il secondo sottoinsieme è un pre-IFIB, che consente al LC di inoltrare il pacchetto al processo appropriato se esiste un solo processo o a un flow manager appropriato.
3. I gestori del flusso aiutano a distribuire ulteriormente i pacchetti se la ricerca non è banale, ad esempio, se si utilizzano più processi per BGP. Ciascun gestore di flusso dispone di una o più slice dell'IFIB e inoltra correttamente i pacchetti ai processi appropriati associati alla slice dell'IFIB.
4. Se non è definita una voce per la porta di destinazione, è possibile eliminarla o inoltrarla al gestore dei flussi. Un pacchetto viene inoltrato senza alcuna porta associata se esiste un criterio associato per la porta. Il gestore del flusso consente quindi di generare una nuova voce di sessione.

Come viene inoltrato un pacchetto interno?

Esistono due tipi di flussi, i flussi di layer 2 (HDLC, PPP) e i flussi e i flussi di routing ICMP/PING di layer 4.

1. Layer 2 HDLC/PPP: questi pacchetti sono identificati dall'identificatore del protocollo e vengono inviati direttamente alle code della CPU nello spray. I pacchetti del protocollo di layer 2 hanno alta priorità e vengono quindi prelevati dalla CPU (tramite Squid) ed elaborati. Pertanto, i pacchetti keepalive per il layer 2 ricevono risposta diretta tramite la scheda LC tramite la CPU. In questo modo si evita di dover ricorrere al punto di ripristino per ottenere risposte e si interagisce con il tema della gestione di interfacce distribuite.
2. I pacchetti ICMP (layer 4) vengono ricevuti nel VLAN e inviati tramite ricerca tramite IFBI alle code della CPU sullo spray. Questi pacchetti vengono quindi inviati alla CPU (tramite Squid) ed elaborati. La risposta viene quindi inviata attraverso le code di uscita dello spray per

essere inoltrata attraverso il fabric. In questo caso, anche un'altra applicazione necessita delle informazioni (replicate tramite la struttura). Una volta attraversato il fabric, il pacchetto è destinato al corretto LC in uscita e attraverso la spugna e la coda di controllo appropriati.

3. I flussi di routing vengono cercati nell'IFIB e quindi inviati alle code di data shaping di output (8000 code), una delle quali è riservata ai pacchetti di controllo. Si tratta di una coda senza forma che viene semplicemente servita ogni volta che è piena. - priorità alta. Il pacchetto viene quindi inviato attraverso la struttura in code ad alta priorità in un set di code CPU sullo Sponge (simili alle code Squid sullo Sprayer), e quindi viene elaborato dal processo appropriato, dal gestore di flusso o dal processo effettivo. La risposta viene inviata tramite la spugna della scheda di linea in uscita e quindi tramite la scheda di linea. La spugna LC in uscita dispone di una coda speciale riservata alla gestione dei pacchetti di controllo. Le code nello Sponge sono suddivise in pacchetti ad alta priorità, pacchetti di controllo e pacchetti a bassa priorità, per porta di uscita.
4. Il Navigatore struttura di prodotto dispone di un set di policer configurati per limitare la velocità, il layer 4, il layer 2 e instradare pacchetti. Questi sono preimpostati e modificati per essere configurabili dall'utente in un secondo momento.

Uno dei problemi più comuni dei pacchetti LPT è che vengono scartati quando si tenta di eseguire il ping sul router. I policer LPTS limitano in genere la velocità di questi pacchetti. Ciò avviene per confermare:

```
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

I pacchetti IP sono intrinsecamente non sicuri. IPsec è un metodo utilizzato per proteggere i pacchetti IP. CRS-1 IPsec è implementato nel percorso di inoltro software, pertanto la sessione IPsec viene terminata sull'RP/DRP. È supportato un numero totale di 500 sessioni IPSec per CRS-1. Il numero dipende dalla velocità della CPU e dalle risorse allocate. Non vi sono limiti software a questo, solo il traffico originato localmente e terminato localmente su RP sono idonei per la gestione di IPSec. Per il tipo di traffico è possibile utilizzare la modalità trasporto IPsec o la modalità tunnel, anche se la prima è preferibile a causa di un minore sovraccarico nell'elaborazione IPsec.

R3.3.0 supporta la crittografia di BGP e OSPFv3 su IPsec.

Per ulteriori informazioni su come implementare IPsec, consultare la guida alla configurazione

della sicurezza del sistema Cisco IOS XR.

Nota: IPsec richiede la funzione crypto pie, ad esempio hfr-k9sec-p.pie-3.3.1.

Fuori banda

Accesso da console e AUX

I CRS-1 RP/SC dispongono di una console e di una porta AUX per la gestione fuori banda, nonché di una porta Ethernet per la gestione fuori banda tramite IP.

La console e la porta AUX di ciascun RP/SCGE, due per chassis, possono essere collegate a un console server. Ciò significa che il sistema a chassis singolo richiede quattro porte console e che i sistemi a chassis multiplo richiedono 12 porte più altre due porte per i Supervisor Engine su Catalyst 6504-E.

La connessione della porta AUX è importante in quanto fornisce l'accesso al kernel IOS-XR e può consentire il ripristino del sistema quando ciò non è possibile tramite la porta della console. L'accesso tramite la porta AUX è disponibile solo per gli utenti definiti localmente sul sistema e solo quando l'utente ha accesso a livello di sistema radice o supporto cisco. Inoltre, l'utente deve avere una password **segreta** definita.

Accesso terminale virtuale

Èpossibile usare Telnet e Secure Shell (SSH) per raggiungere il CRS-1 tramite le porte vty. Per impostazione predefinita, entrambi sono disattivati e l'utente deve attivarli esplicitamente.

Nota: IPsec richiede la funzione crypto pie, ad esempio hfr-k9sec-p.pie-3.3.1.

Per abilitare SSH, generare prima le chiavi RSA e DSA come mostrato nell'esempio:

```
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
```

Informazioni correlate

- Supporto router
- Documentazione e supporto tecnico – Cisco Systems