

CRS-1 en IOS XR operationele beste praktijken

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Cisco IOS XR-Overzicht](#)

[Processen en draden](#)

[Processtaten en Threat-staten](#)

[Synchronous Message Passing](#)

[Verbonden proces- en processtaten](#)

[Belangrijke processen en hun functies](#)

[Netio](#)

[PROCES VOOR groepsservices \(GSP\)](#)

[BCDL Bulk contentdownloader](#)

[Lichtgewicht Messaging \(LWM\)](#)

[Envmon](#)

[CRS-1 fabric-inleiding](#)

[Het fabric-vlak](#)

[Fabricbewaking](#)

[Overzicht van besturingsplane](#)

[Catalyst 6500-configuratie](#)

[Beheer van besturingsplane voor meerdere chassis](#)

[ROMMON en Monlib](#)

[upgrade-instructies](#)

[PLIM en MSC - Overzicht](#)

[PLIM-overabonnement](#)

[Configuratie-beheer](#)

[Security](#)

[LPTS](#)

[Hoe wordt een intern pakket doorgestuurd?](#)

[Uit de band](#)

[Gerelateerde informatie](#)

[Inleiding](#)

Dit document helpt u bij het begrijpen van de volgende documenten:

- Processen en draden
- CRS-1 fabric-enabled
- besturingsplane
- Rommon en Monlib
- Physical Layer-interfacemodule (PLIM) en modulaire servicekaart (MSC)
- Configuratie-beheer
- Security
- Uit de band
- Simple Network Management Protocol (SNMP)

Voorwaarden

Vereisten

Cisco raadt u aan om kennis te hebben van Cisco IOS[®] XR.

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- Cisco IOS XR-software
- CRS-1

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

Conventies

Raadpleeg [Cisco Technical Tips Conventions \(Conventies voor technische tips van Cisco\) voor meer informatie over documentconventies.](#)

Cisco IOS XR-Overzicht

Cisco IOS XR is ontworpen om te schalen. Het deelvenster is een Microkernel-architectuur en biedt dus alleen essentiële services, zoals procesbeheer, schema's, signalen en timers. Alle andere diensten zoals bestandssystemen, chauffeurs, protocolstapels en toepassingen worden beschouwd als resource managers en draaien in geheugen beschermde gebruikersruimte. Deze andere services kunnen tijdens de looptijd worden toegevoegd of verwijderd, afhankelijk van het ontwerp van het programma. De voetafdruk van Microkernel is slechts 12 kb. Het Microsoft en het onderliggende besturingssysteem komen van QNX Software Systems, dat wordt Neutrino genoemd. QNX is gespecialiseerd in realtime besturingssysteemontwerp. De microkernel is preventief en de planner is op prioriteit gebaseerd. Dit waarborgt dat context-switching tussen processen zeer snel is en dat de hoogste prioriteit dithreads altijd toegang tot CPU hebben wanneer nodig. Dit zijn een aantal voordelen waarvan Cisco IOS XR gebruik maakt. Maar het grootste voordeel is het erfelijke ontwerp van Inter Processing Communications binnen de kern van de besturingssysteem.

Neutrino is een besturingssysteem voor het doorgeven van berichten en berichten zijn de basisinstrumenten voor de communicatie tussen alle verbindingen. Wanneer een bepaalde Server een dienst wil verlenen, creëert het een kanaal voor het uitwisselen van berichten. Clients hechten zich aan het kanaal van de servers door deze rechtstreeks in kaart te brengen naar de relevante bestandsindeling om de service te kunnen gebruiken. Alle communicatie tussen client en server is volgens hetzelfde mechanisme. Dit is een enorm voordeel voor een supercomputer, wat CRS-1 is. Overweeg deze wanneer een lokale gelezen handeling op een standaard UNIX-kern wordt uitgevoerd:

- Software onderbreekt in de kern.
- Kernel stuurt het bestandssysteem in.
- Gegevens worden ontvangen.

Denk eens na in het geval op afstand:

- Software onderbreekt in de kern.
- Kernel stuurt NFS uit.
- NFS roept de netwerkcomponent aan.
- Remote wordt verzonden naar de netwerkcomponent.
- NFS wordt gebeld.
- Kernel zendt het bestandssysteem uit.

De semantiek voor het lokale lezen en het externe lezen zijn niet hetzelfde. Argumenten en parameters voor het vergrendelen van bestanden en het instellen van rechten zijn anders.

Neem het lokale QNX-geval:

- Software onderbreekt in de kern.
- Kernel voert bericht uit dat het bestandssysteem binnengaat.

Neem het niet-lokale geval in overweging:

- Software onderbreekt in de kern.
- Kernel gaat naar QNET, het IPC-transportmechanisme.
- QNET gaat naar de kern.
- Kernel zendt het bestandssysteem uit.

Alle semantiek die betrekking hebben op argument-passeren en bestands systeemparemeters zijn identiek. Alles is losgekoppeld op de IPC-interface waardoor de client en de server volledig gescheiden kunnen worden. Dit betekent dat elk proces op elk moment kan worden uitgevoerd. Als een bepaalde routeprocessor te druk bezig is met het indienen van verzoeken, kunt u die services gemakkelijk verplaatsen naar een andere CPU die op een DRP draait. Een supercomputer die verschillende services op verschillende CPU's uitvoert, verspreid over meerdere knooppunten die gemakkelijk met een ander knooppunt kunnen communiceren. De infrastructuur is aanwezig om de mogelijkheid te bieden om te schalen. Cisco heeft dit voordeel gebruikt en geschreven extra software die in de basisoperaties van het bericht-passeerkanaal past dat de CRS-router toestaat om naar duizenden knooppunten te schalen, waar een knooppunt, in dit geval een CPU, een exemplaar van het besturingssysteem uitvoert, of het nu een routeproces (RP), een gedistribueerde routeprocessor (DRP), een modulaire serviceskaart (MSC) of een Switch Processor (SP) is.

[Processen en draden](#)

Binnen de grenzen van Cisco IOS XR, is een proces een beschermd gebied van geheugen dat

één of meer draden bevat. Vanuit het perspectief van programmeurs doen de draden het werk, en elk voltooit een logisch uitvoeringspad om een specifieke taak uit te voeren. Het geheugen dat de draden nodig hebben tijdens de stroom van uitvoering is eigendom van het proces dat zij binnen uitvoeren, beschermd tegen andere procesthreads. Een thread is een uitvoeringseenheid met een uitvoeringscontext die een stack en registers bevat. Een proces is een groep draden die een virtuele adresruimte delen, hoewel een proces één thread kan bevatten maar vaak meer bevat. Als een andere thread in een ander proces probeert om naar het geheugen te schrijven in uw proces, wordt het betreffende proces gedood. Als er meer dan één draad is die binnen uw proces werkt, dan heeft die draad toegang tot hetzelfde geheugen binnen uw proces, en kan diens volgorde de gegevens van een andere draad overschrijven. Voltooi de stappen in een procedure om synchronisatie te behouden met resources om deze thread in hetzelfde proces te voorkomen.

Een thread gebruikt een object dat een wederzijdse uitsluiting (MUTEX) wordt genoemd om wederzijdse uitsluiting van diensten te waarborgen. De thread die MUTEX heeft, is de thread die naar een bepaald gebied van het geheugen kan schrijven als voorbeeld. Andere draden die de MUTEX niet hebben, kunnen dat niet. Er zijn ook andere mechanismen om de synchronisatie van hulpmiddelen te verzekeren, en dit zijn Semaphores, Voorwaardelijke Variabelen of Condvvars, Barriers, en Sleepons. Deze worden hier niet besproken, maar ze bieden synchronisatiediensten als onderdeel van hun taken. Als u de principes die hier aan Cisco IOS worden besproken vergelijkt, dan is Cisco IOS één proces dat vele draden exploiteert, met alle draden die toegang tot de zelfde geheugenruimte hebben. Maar, Cisco IOS roept deze draden processen.

Processtaten en Threatstaten

Binnen Cisco IOS XR zijn er servers die de services en klanten leveren die de services gebruiken. Een bepaald proces kan een aantal draden hebben die dezelfde dienst leveren. Een ander proces kan een aantal klanten hebben die op elk moment een specifieke dienst nodig kunnen hebben. Toegang tot de servers is niet altijd beschikbaar en als een klant om toegang tot een dienst vraagt, zit hij daar en wacht tot de server gratis is. In dit geval zou de cliënt geblokkeerd zijn. Dit wordt een blokkerend model van de clientserver genoemd. De client is mogelijk geblokkeerd omdat hij wacht op een resource als MUTEX, of omdat de server nog niet heeft geantwoord.

Geef een **opdracht** voor het **tonen van proces** op om de status van de draden in het ospf-proces te controleren:

```
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 SHARED MEM MAIN MEM
      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

Merk op dat ospf-proces een baan-ID (JID) krijgt, die 250 is. Dit verandert nooit op een actieve router en in het algemeen op een bepaalde versie van Cisco IOS XR. Binnen het spf-proces zijn er vijf draden elk met hun eigen Draad-ID (TID). Hierop staat de stapelruimte voor elke thread, de prioriteit van elke thread en de staat ervan.

Synchronous Message Passing

Een eerder vermeld QNX-systeem is een besturingssysteem dat berichten doorgeeft. Het is een synchroon systeem voor het passeren van berichten. Veel van de besturingssysteemproblemen worden weerspiegeld in het synchrone berichtenverkeer. Het is niet gezegd dat synchroon bericht passeren problemen veroorzaakt, maar eerder wordt het probleemsymptoom gereflecteerd in het synchrone bericht passeren. Omdat het synchroon is, is informatie over de levenscyclus of de staat gemakkelijk toegankelijk voor de CRS-1-exploitant, die de probleemoplossing ondersteunt. Het bericht met een levenscyclus lijkt hierop:

- Een server maakt een berichtenkanaal.
- Een client sluit aan op het kanaal van een server (analoog aan positie open).
- Een client stuurt een bericht naar een server (MsgSend) en wacht op een antwoord en blokkeert.
- De server ontvangt (MsgOntvang) een bericht van een cliënt, verwerkt het bericht, en antwoordt aan de cliënt.
- De client opent en verwerkt het antwoord van de server.

Dit blokkerende client-server model is het synchrone bericht-passeren. Dit betekent dat de cliënt een bericht en blokken stuurt. De server ontvangt het bericht, verwerkt het, antwoordt terug naar de client en ontgrendelt de client. Dit zijn de specifieke details:

- Server wacht in RECEIVE status.
- De client stuurt een bericht naar de server en wordt BLOCKED.
- De server ontvangt het bericht en ontgrendelt, indien het wachten in de ontvangen status is.
- Clientverplaatsingen naar de REPLY-status.
- Server beweegt naar de RUNNING status.
- Server verwerkt het bericht.
- De server reageert op de client.
- Clientontgrendeling.

Geef de **opdracht** van het **showproces uit** om te zien in welke staat de client en servers zijn.

```
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

```

Verbonden proces- en processtaten

Geef de **verstoopt** opdracht van het **showproces uit** om te zien hoe het verstoopt proces verloopt.

```
RP/0/RP1/CPU0:CWD CRS#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#

Door gesynchroniseerd bericht-passeren kunt u gemakkelijk de levenscyclus van interprocescommunicatie tussen de verschillende draden volgen. Op elk moment kan een thread in een specifieke status worden uitgevoerd. Een geblokkeerde staat kan een symptoom van een probleem zijn. Dit betekent niet dat als een thread in geblokkeerde toestand is, er een probleem is, zodat u het **proces-geblokkeerde** opdracht niet **uitvoert** en een case opent met Cisco Technical Support. Vergrendelde draden zijn ook heel normaal.

Let op de vorige uitvoer. Als je kijkt naar de eerste draad in de lijst, let op, het is de ksh, en zijn antwoord is geblokkeerd op devc-conaux. De klant, de ksh in dit geval, stuurde een bericht naar het devc-conaux proces, de server, die devc-conaux is, houdt een ksh-antwoord geblokkeerd tot het antwoordt. Ksh is de UNIX schaal die iemand gebruikt op de console of AUX poort. Ksh wacht op input van de console, en als er geen is omdat de operator niet typt, dan blijft de kraan geblokkeerd tot het moment waarop het enige input verwerkt. Na de verwerking keert ksh terug om verstopt te antwoorden op devc-conaux.

Dit is normaal en illustreert geen probleem. Het punt is dat geblokkeerde draden normaal zijn, en het hangt af van de XR-versie, het type systeem dat u hebt, wat u hebt ingesteld en wie doet wat dat de uitvoer van het **proces** van **show verstopt** opdracht verandert. Het gebruik van de opdracht **proces geblokkeerd** tonen is een goede manier om te beginnen met het oplossen van OS type problemen. Als er een probleem is, bijvoorbeeld de CPU is hoog, gebruikt u de vorige opdracht om te zien of er iets buiten het normale bereik valt.

Begrijp wat normaal is voor uw functionerende router. Dit biedt een basislijn voor u om als vergelijking te gebruiken wanneer u de levenscycli van het proces van probleemoplossing gebruikt.

Op elk moment kan een thread in een bepaalde staat zijn. Deze tabel bevat een lijst van de staten:

Indien de staat:	De drempel is:
DOOD	Dood. Het Kernel wacht op het vrijgeven van de dieptes.
STRIJKEN	Actief uitvoeren op een CPU
KLAAR	Geen CPU's uitvoeren, maar deze kunnen ook worden uitgevoerd
GESTOPT	Opgesteld (SIGSTOP-signaal)
VERZENDEN	Wachten op een server om een bericht te ontvangen
ONTVANGEN	Wachten op een client om een bericht te verzenden
ANTWOORDEN	Wachten op een server om een

	bericht te beantwoorden
STACKEN	Wachten op meer toe te wijzen stack
WACHTEN	Wacht tot de procesmanager een pagina-fout heeft opgelost
SIGSUSPEND	Wachten op een signaal
SIGWAITINFO	Wachten op een signaal
NANOSLEEP	Slapen gedurende een periode
MUTEX	Wachten op de verwerving van een MUTEX
CONDVAR	Wachten op een voorwaardelijke variabele die moet worden aangegeven
SAMENVOEGEN	Wachten op de voltooiing van een andere draad
INTR	Wachten op een onderbreking
SEM	Wachten op de verwerving van een semafoor

Belangrijke processen en hun functies

Cisco IOS XR heeft veel processen. Dit zijn een paar belangrijke punten met hun taken die hier zijn toegelicht.

WatchDog System Monitor (WDSysmon)

Dit is een service die geleverd wordt voor het detecteren van proceshangers en geheugenstoornissen. Een geheugenlek of een andere vreemde omstandigheid kan leiden tot een laag geheugen. Een hang kan het resultaat zijn van een aantal voorwaarden zoals procespatches, oneindige lussen, kernelsluitingen of planningsfouten. In elke multithreaded omgeving kan het systeem zich in een toestand bevinden die bekend staat als een impasse of simpelweg een impasse. Een impasse kan zich voordoen wanneer een of meer threads niet kunnen worden voortgezet vanwege een gebrek aan bronnen. thread A kan bijvoorbeeld een bericht naar thread B sturen terwijl tegelijkertijd thread B een bericht naar thread A verstuurt. Beide draden wachten op elkaar en worden geblokkeerd door versturen van de tekst en beide draden wachten voorgoed. Dit is een simpel geval dat twee draden impliceert, maar als een server verantwoordelijk is voor een resource dat door veel threads wordt gebruikt, wordt geblokkeerd op een andere thread, dan kunnen de vele draden die om toegang tot dat resource vragen geblokkeerd worden verzonden door het wachten op de server.

Er kunnen zich tussen een paar draden vertragingen voordoen, maar deze kunnen ook andere factoren beïnvloeden. Tijdsloten worden vermeden door een goed programmaontwerp, maar ongeacht hoe prachtig een programma is ontworpen en geschreven. Soms kan een bepaalde opeenvolging van gebeurtenissen die data afhankelijk van specifieke tijden zijn een impasse veroorzaken. Deadlocksparran zijn niet altijd deterministisch en zijn over het algemeen zeer moeilijk te reproduceren. WDSysmon heeft vele draden met één die op de hoogste prioriteit draait die Neutrino ondersteunt, 63. Running op prioriteit 63 waarborgt dat thread CPU-tijd krijgt in een op prioriteit gebaseerde, preventieve planningsomgeving. WDSysmon werkt met de mogelijkheid van een hardwarewaakhond en kijkt naar de softwareprocessen die op hang worden gecontroleerd. Wanneer dergelijke omstandigheden worden gedetecteerd, verzamelt WDSysmon

verdere informatie rond de voorwaarde, kan het proces of de kern coredump, uitschrijven naar syslogs, scripts en de dodelijke processen doden. Afhankelijk van hoe drastisch het probleem is, kan het een routeprocessorswitch opstarten om de werking van het systeem te onderhouden.

```
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

Het proces WDSysmon heeft negen draden. Vier draaien op prioriteit 10, de andere vier op 11, 19, 20 en 63. Wanneer een proces wordt ontworpen, beschouwt de programmeur zorgvuldig de prioriteit die elke draad in het proces moet krijgen. Zoals eerder besproken is de planner op prioriteit gebaseerd, wat betekent dat een hogere prioriteit altijd voorrang heeft op een lagere prioriteit. Prioriteit 63 is de hoogste prioriteit die een draad kan lopen, in dit geval thread 7. Thread 7 is de watcher draad, de draad die CPU-slangen volgt. Het moet met een hogere prioriteit draaien dan de andere draden die het anders ziet, heeft het misschien helemaal geen kans om te draaien, wat het verhindert van de stappen die het ontworpen heeft om te doen.

[Netio](#)

In Cisco IOS, is er het concept van snelle omschakeling en processwitching. Snelle switching gebruikt de CEF-code en treedt op bij onderbreking. Processwitching gebruikt ip_input, de IP switching code, en is een gepland proces. Op hogere eindplatforms wordt CEF-switching in hardware toegepast en ip_input is gepland op de CPU. De equivalent van ip_input in Cisco IOS XR is 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

PROCES VOOR groepsservices (GSP)

Er is behoefte aan communicatie in elke supercomputer met enkele duizenden knooppunten die elk hun eigen exemplaar van de kern runnen. In het internet wordt één op vele communicatie efficiënt uitgevoerd via multicastprotocollen. SAP is het interne multicastprotocol dat voor IPC binnen CRS-1 wordt gebruikt. SAP biedt één aan vele betrouwbare groepscommunicatie die zonder verbinding is met asynchrone semantiek. Hierdoor kan het SAP worden geschaald naar de duizend knooppunten.

```

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 SHARED MEM 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 contentdownloader](#)

BCDL wordt gebruikt om multicast gegevens naar diverse knooppunten zoals RP's en MSC's betrouwbaar te verzenden. Het gebruikt SAP als het onderliggende vervoer. BCDL garandeert de levering van berichten. Binnen het BCDL is er een agent, een producent en een consument. De agent is het proces dat met de producent communiceert om de gegevens voor zijn multicast aan de consumenten op te halen en te bufferen. De producent is het proces dat de data produceert die iedereen wil, en de consument is het proces dat geïnteresseerd is in het ontvangen van de gegevens van de producent. BCDL wordt gebruikt tijdens Cisco IOS XR-softwareupgrades.

[Lichtgewicht Messaging \(LWM\)](#)

LWM is een door Cisco gemaakte vorm van overseinen die ontworpen werd om een laag van abstractie tussen de toepassingen te creëren die met elkaar en met Neutrino communiceren, met het doel als onafhankelijkheid van het besturingssysteem en de transportlaag. Als Cisco de verkoper van het besturingssysteem van QNX naar iemand anders wil wijzigen, helpt een laag van abstractie tussen de rudimentaire functies van het onderliggende besturingssysteem de afhankelijkheid van het besturingssysteem en de ondersteuning bij het poorten naar een ander besturingssysteem te verwijderen. LWM biedt synchrone gegarandeerde berichtlevering, die net als het doorgeven van het native Neutrino-bericht veroorzaakt dat de zender blokkeert totdat de ontvanger antwoordt.

LWM biedt ook asynchrone berichtlevering via 40 bit pulsen. Asynchrone berichten worden asynchrone verzonden, wat betekent dat het bericht in de wachtrij staat en de zender niet blokkeert, maar asynchrone ontvangen wordt door de server, maar wanneer de server opiniepeilingen voor het volgende beschikbare bericht geeft. LWM is gestructureerd als client/server. De server creëert een kanaal dat het een oor geeft om in te luisteren voor berichten en in te zitten terwijl de loop een bericht ontvangt dat luistert op het kanaal, dat het net creëerde. Wanneer een bericht aankomt, ontslaat het de blokkering en krijgt het een client-identificator, wat in feite hetzelfde is als de ontvangstID uit het ontvangen bericht. De server voert dan wat verwerking uit en geeft later een bericht antwoord naar de client identifier.

Aan de kant van de cliënt doet het een bericht verbinden. Het krijgt een identificatiemiddel over waaraan het verbindt en dan een bericht verzenden en wordt geblokkeerd. Wanneer de server klaar is met verwerken, reageert deze en wordt de client vrijgegeven. Dit is nagenoeg hetzelfde als het passeren van Neutrinos native message, dus de abstractielaag is zeer dun.

LWM wordt ontworpen met een minimum aantal systeemaanroepen en contextswiches voor hoge prestaties en is de geprefereerde methode van IPC in de Cisco IOS XR-omgeving.

Envmon

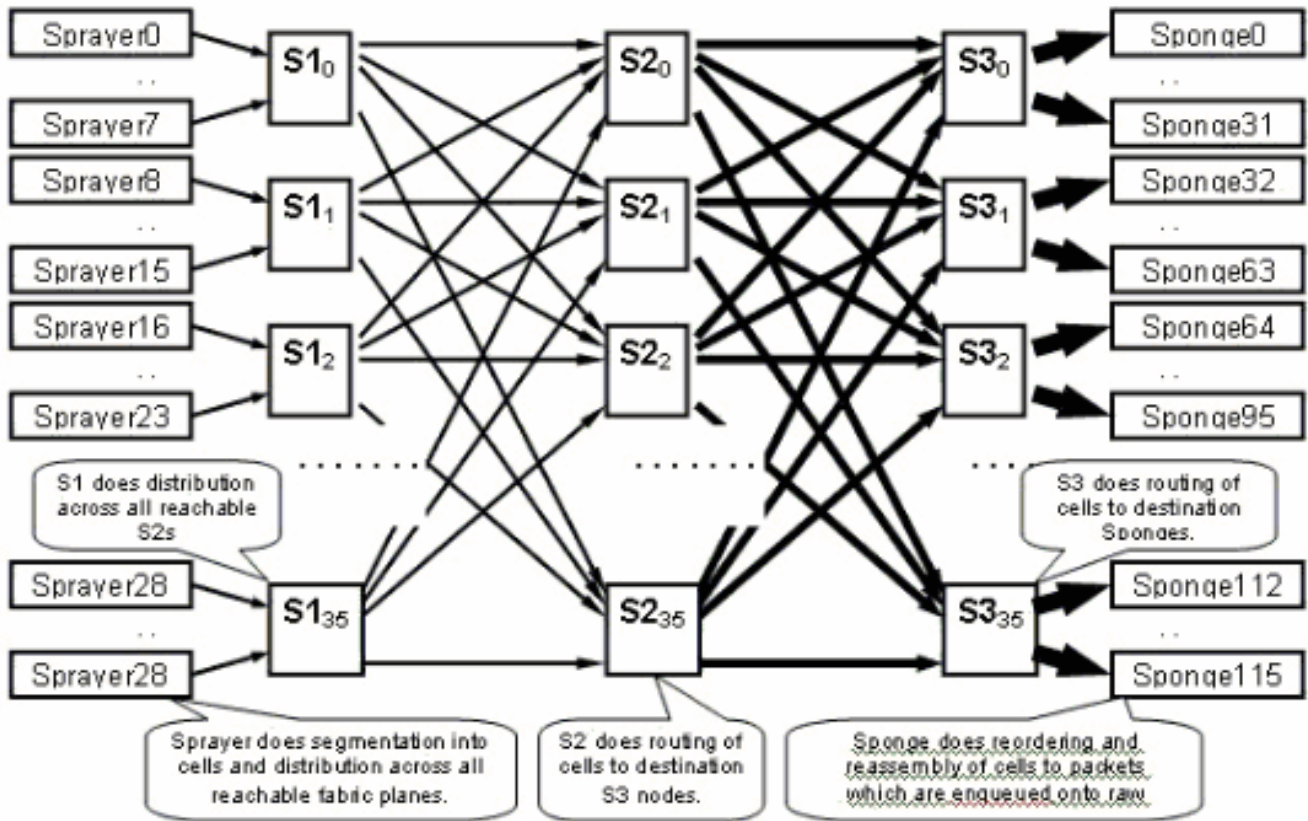
Op het meest fundamentele niveau is het milieumonitoringsysteem verantwoordelijk voor de waarschuwing wanneer fysieke parameters, bijvoorbeeld temperatuur, spanning, ventilatorsnelheid enzovoort, buiten het operationele bereik vallen en de hardware uitzetten die kritische niveaus benadert waar de hardware beschadigd kan worden. Het controleert periodiek elke beschikbare hardwaresensor, vergelijkt de gemeten waarde met kaartspecifieke drempels en verhoogt zo nodig alarm om deze taak te kunnen vervullen. Een aanhoudend proces, gestart bij systeeminicialisatie, dat periodiek alle hardwaresensoren, bijvoorbeeld voltage, temperatuur en ventilatorsnelheid, in het chassis peilt en deze gegevens aan externe beheerklanten doorgeeft. Bovendien worden in het periodieke proces de waarden van de sensor vergeleken met de alarmdrempels en worden milieuwaarschuwingen in de systeemdatabse gepubliceerd voor verdere actie door de foutmanager. Als de sensorlezingen gevaarlijk buiten bereik zijn, kan het milieubewakingsproces ertoe leiden dat de kaart wordt afgesloten.

CRS-1 fabric-inleiding

- Multiflex Fabric-3 streeksen topologie
- Dynamische routing binnen de stof om congestie te minimaliseren
- Op cellen gebaseerd: 136 bytcellen, 120 byte gegevenslading
- Flow-controle om de verkeersisolatie te verbeteren en de buffervereisten in de stof te minimaliseren
- Niveau naar Niveau Snelheid
- Twee kosten van ondersteund verkeer (Unicast & multicast)
- Twee verkeersprioriteiten per cast (hoog en laag)
- Ondersteuning voor 1M fabric multicast groepen (FGID's)
- Kosteneffectieve fouttolerantie: N+1 of N+k redundantie met behulp van fabric-vliegtuigen in plaats van 1+1, tegen sterk verhoogde kosten

Wanneer u in een enkele chassismodus draait, bevinden de S1-, S2- en S3-apparaten zich op dezelfde weefselkaarten. Deze kaart wordt ook gewoonlijk **S123 kaart** genoemd. Bij een configuratie met meerdere chassis wordt de S2 gescheiden en gebruikt op het Fabric Card Chassis (FCC). Voor deze configuratie zijn twee stofkaarten nodig, een S2-kaart en een S13-kaart. Elke MSC sluit zich aan op acht wasvliegtuigen om redundantie te bieden, zodat wanneer u een of meer vlakken los maakt, uw stof nog steeds door verkeer komt hoewel het totale verkeer, dat door de stof kan gaan, lager is. Het CRS kan nog steeds op linerate werken voor de meeste pakketformaten met slechts zeven vliegtuigen. De rugdruk wordt over het stof verzonden over een vreemd en zelfs vlak. Het wordt niet aanbevolen een systeem met minder dan twee vlakken in een vreemd en zelfs vlak te laten lopen. Minder dan twee vliegtuigen is geen ondersteunde configuratie.

Het fabric-vlak



Het vorige schema geeft één vlak weer. Je moet dat diagram met acht vermenigvuldigen. Dit betekent dat de verstuiver (ingressq), die is gebaseerd op een LC, zich verbindt met 8 S1s (1 S1 per vliegtuig). De S1 in elk wasvlak wordt aangesloten op 8 verstuivers:

- de acht bovenste LC's van het chassis
- de onderste LC's van 8

Er zijn 16 S1s per LC-chassis met 16 sleuven: 8 voor de bovenste LC's (1 per vlak) + 8 voor de onderste LC's.

Op één chassis met 16 sleuven heeft een S123-fabric-kaart 2 S1s, 2 S2 en 4 S3. Dat maakt deel uit van de stof versnelde berekening. Er is twee keer zoveel verkeer, dat de stof kan verlaten als het verkeer kan binnendringen. Op dit moment zijn er ook twee spons (fabricq) per LC vergeleken met 1 spuit. Dit maakt het mogelijk om zich op de grap LC te bufferen wanneer meer dan één LC ingedrukt wordt en de bovengrens van de C wordt overbelast. De graafkamer LC kan die extra bandbreedte uit het weefsel absorberen.

Fabricbewaking

Beschikbaarheid en connectiviteit van het vliegtuig:

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

Controleer of velden cellen ontvangen/verzenden en sommige fouten nemen toe:

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

De tekens in de vorige opdracht:

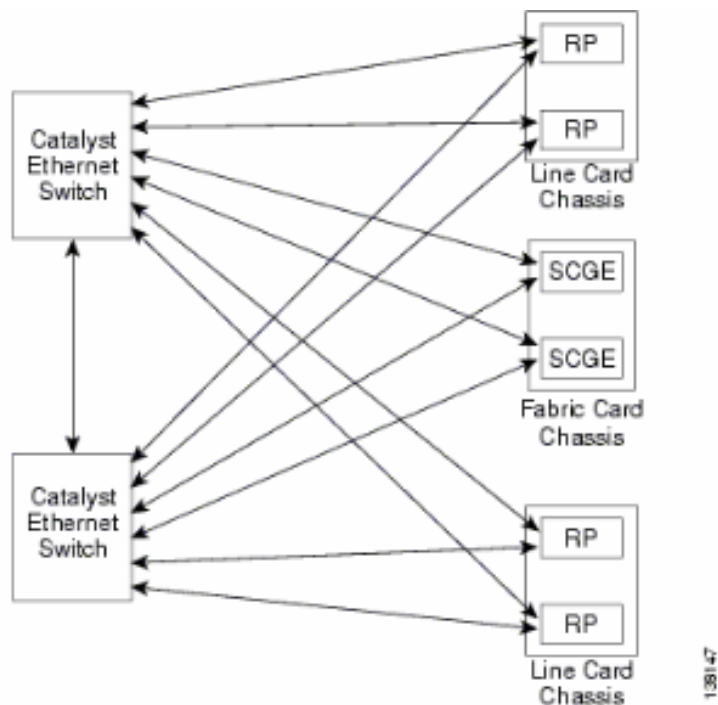
- CE—corrieerbare fout
- UCE-niet-corrigeerbare fout
- PE—paringsfout

Maak je geen zorgen als er een paar fouten worden opgemerkt, omdat dit bij het opstarten kan gebeuren. De velden moeten niet tijdens het werken zijn toegenomen. Als ze dit wel zijn, dan kan dit duiden op een probleem in het weefsel. Geef deze opdracht uit om een defect aan de fouten per materiaal te verkrijgen:

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

Overzicht van besturingsplane

De regelvliegtuigverbinding tussen het chassis van de lijnkaart en het fabric chassis is momenteel via Gigabit Ethernet poorten op de RPs (LCC) en SCGE (FCC). De onderlinge verbinding tussen de poorten wordt verschaft via een paar Catalyst 6500 switches, die kunnen worden aangesloten via twee of meer Gigabit Ethernet-poorten.



Catalyst 6500-configuratie

Deze configuratie wordt aanbevolen voor Catalyst-switches die worden gebruikt voor het besturingsplane met meerdere chassis:

- Eén VLAN wordt op alle poorten gebruikt.
- Alle poorten worden in de toegangsmodus (geen trunking) uitgevoerd.
- Spanning-boom 802.1w/s wordt gebruikt voor luspreventie.
- Er worden twee of meer koppelingen gebruikt om de twee switches onderling te verbinden en STP wordt gebruikt voor luspreventie. Kanteling wordt niet aanbevolen.
- poorten die aansluiten op CRS-1 RP en SCGE gebruiken pre-standaard modus omdat IOS-

XR de op standaarden gebaseerde 802.1s niet ondersteunt.

- UDLD moet worden ingeschakeld op de poorten die tussen de switches en tussen de switches en de RP/SCGE aansluiten.
- UDLD is standaard ingeschakeld op het CRS-1.

Raadpleeg [de Cisco IOS XR-software op een systeem met meerdere schappen](#) voor meer informatie over het configureren van een Catalyst 6500 in een systeem met meerdere schappen.

Beheer van besturingsplane voor meerdere chassis

Catalyst 6504-E chassis, dat de besturingssysteemconnectiviteit voor het systeem met meerdere chassis biedt, is voor deze beheerservices geconfigureerd:

- In-band beheer via poort Gigabit 1/2, dat aangesloten is op een LAN switch bij elke PoP. Toegang is alleen toegestaan voor een klein aantal subnetten en protocollen.
- NTP wordt gebruikt om de systeemtijd in te stellen.
- Syslogging wordt uitgevoerd naar de standaard hosts.
- SNMP-opiniepeiling en -trap kunnen worden ingeschakeld voor belangrijke functies.

Opmerking: De in bedrijf zijnde Catalyst mag niet worden gewijzigd. Voorafgaande tests dienen te worden uitgevoerd op elke geplande wijziging en het wordt ten zeerste aanbevolen dit tijdens een onderhoudsvenster te doen.

Dit is een voorbeeld van de beheerconfiguratie:

```
#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 en Monlib

Cisco monlib is een uitvoerbaar programma dat op het apparaat wordt opgeslagen en in RAM wordt geladen voor uitvoering door ROMMON. ROMMON gebruikt monlib om de bestanden op het apparaat te benaderen. ROMMON-versies kunnen worden bijgewerkt en dienen dit te doen onder aanbeveling van Cisco Technical Support. De laatste ROMMON-versie is 1.40.

upgrade-instructies

Voer de volgende stappen uit:

1. Download de ROMMON binaire woordenboeken van [Cisco CRS-1 ROMMON](#) (alleen [geregistreerde](#) klanten).
2. Pak het TAR-bestand uit en kopieer de 6 BIN-bestanden naar de CRS-basismap van 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. Gebruik de **showdiag | INC-ROM|NODE|PLIM**-opdracht om de huidige versie van het Romeinse programma te zien.

```
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 : 40C192-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 : 160C48-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. Ga naar de ADMIN-modus en gebruik de **upgrade-optie op de volledige schijf 0-opdracht om de ROMMON te upgraden.**

```
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. Afsluiten ADMIN Mode en **show logbestand** invoeren | inc "OK, ROMMON A" en zorg ervoor dat alle knooppunten met succes zijn bijgewerkt. Als een van de knooppunten faalt, ga dan terug naar stap 4 en herprogramma.

```
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. Ga naar de ADMIN-modus en gebruik de **upgrade-module door de opdracht all disk0 om de ROMMON te upgraden.**

```
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. Afsluiten ADMIN Mode en **show logbestand** invoeren | inc "OK, ROMMON B" en zorg ervoor dat alle knooppunten met succes zijn bijgewerkt. Als een van de knooppunten faalt, ga dan terug naar stap 4 en herprogramma.

```
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. De upgradeopdracht brandt gewoon een speciale gereserveerde sectie van flitsers met de nieuwe ROMMON. Maar de nieuwe ROMMON blijft inactief tot de kaart opnieuw wordt geladen. Dus als je de kaart opnieuw laadt, is de nieuwe ROMMON actief. Zet elk knooppunt één voor één terug of stel de gehele router opnieuw in om dit te doen.

Reload Router:

```
RP/0/RP0/CPU0:ROUTER#hw-module node 0/RP0/CPU0 or 0/RP1/CPU0 reload (depends on which one 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 one 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. Gebruik de **showdiag | INC-ROM|NODE|PLIM**-opdracht om de huidige ROMMON-versie te controleren.

```
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 : 40C192-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 : 160C48-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]
```

Opmerking: ROMMON stelt op CRS-8 en fabric-chassis ook de ventilatorsnelheid in op de standaardsnelheid van 4000 toeren/ minuut.

PLIM en MSC - Overzicht

Dit vertegenwoordigt de pakketstroom op de router CRS-1 en deze termen worden uitwisselbaar gebruikt:

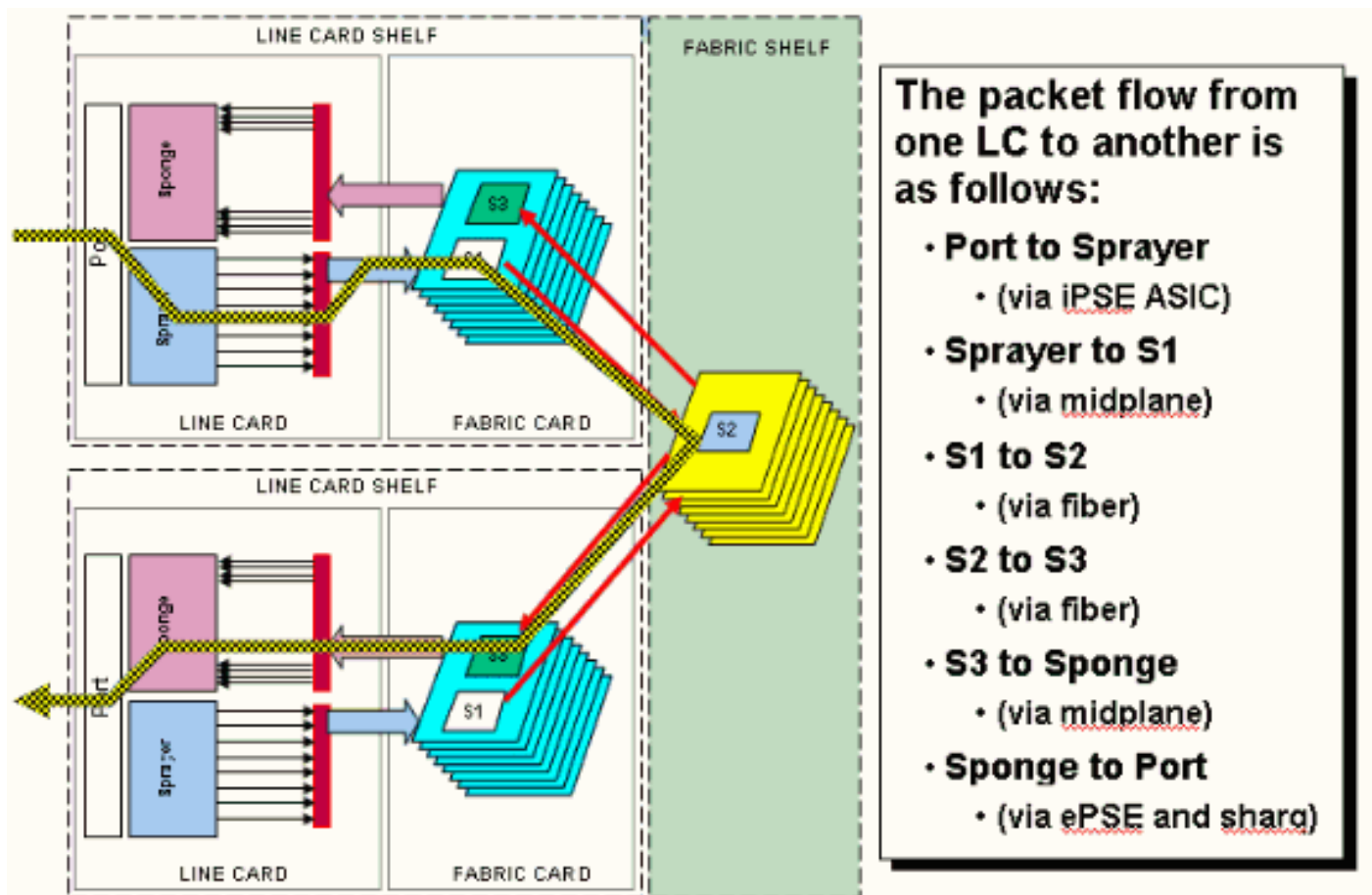
IngressQ ASIC wordt ook Sprayer ASIC genoemd.

FabricQ ASIC wordt ook de spons ASIC genoemd.

EgressQ ASIC wordt ook de Sharq ASIC genoemd.

SPP wordt ook de PSE (Packet Switch Engine) ASIC genoemd.

RX PLIM > RX SPP > Ingress Q > Fabric Q > TX SPP > Uitgangspriet Q > TX PLIM (spons)
(spons) (Sharq)

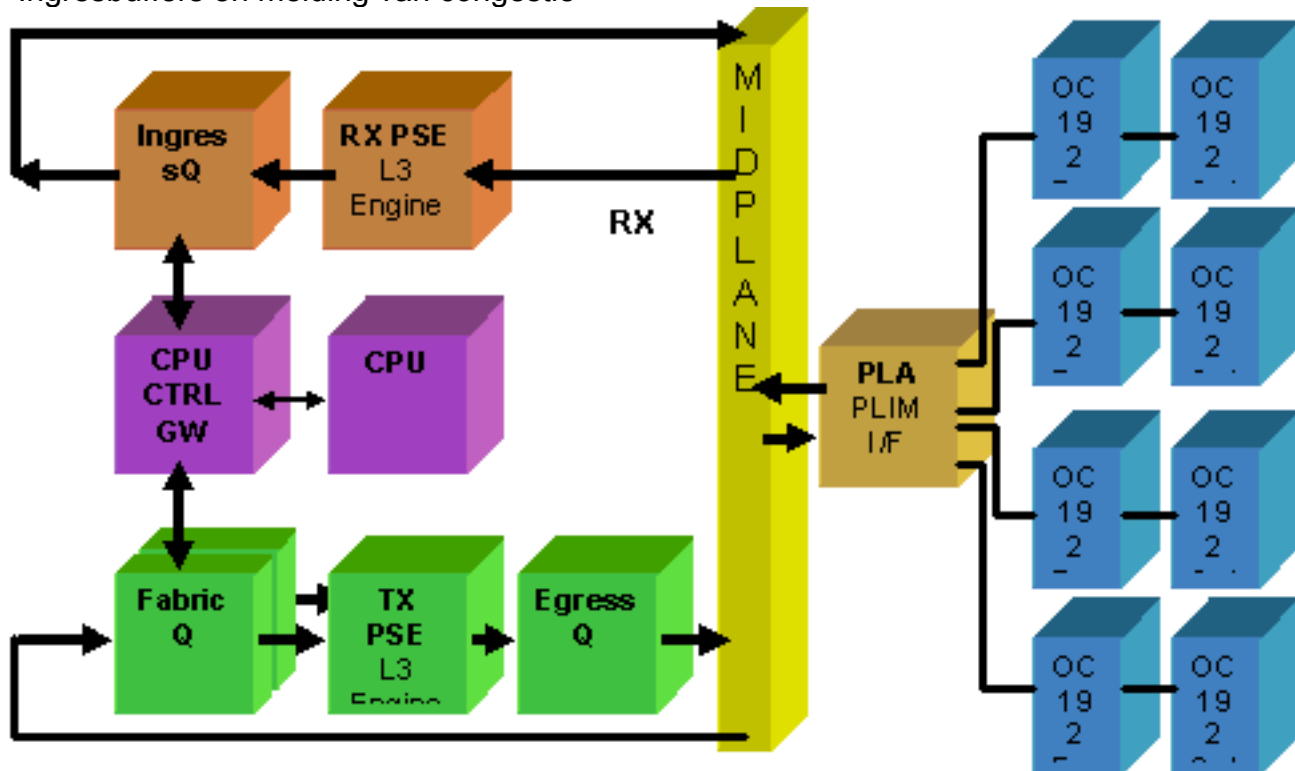


Packets worden ontvangen op de Physical Layer Interface Module (PLIM).

Het PLIM bevat de fysieke interfaces voor de MSC waarmee het wordt gemeten. De PLIM en MSC zijn aparte kaarten die via de chassisbackplane worden aangesloten. Als resultaat hiervan worden de interfacetypen voor een bepaalde MSC gedefinieerd door het type van het PLIM dat het met heeft gemaakt. Afhankelijk van het type PLIM bevat de kaart een aantal ASIC's die de fysieke media en de vormgeving voor de interfaces bieden. Het doel van de PLIM ASIC's is de interface tussen de MSC en de fysieke verbindingen te verschaffen. Het beëindigt de vezel, doet het licht aan elektrische conversie, beëindigt de media die vormgeving SDH/SONET/Ethernet/HDLC/PPP zijn, controleert de CRC, voegt wat controle informatie toe die de bufferknop wordt genoemd en zendt de bits die op de MSC blijven door. Het PLIM bronst/zinkt de HDLC- of PPP-keepalives niet. Deze worden door CPU op de MSC verwerkt.

Het PLIM biedt ook de volgende functies:

- MAC-filtering voor 1/10 Gigabit Ethernet
- Ingress/ress MAC-accounting voor 1/10 Gigabit Ethernet-module
- VLAN-filtering voor 1/10 Gigabit Ethernet
- VLAN-accounting voor 1/10 Gigabit Ethernet
- Ingresbuffers en melding van congestie



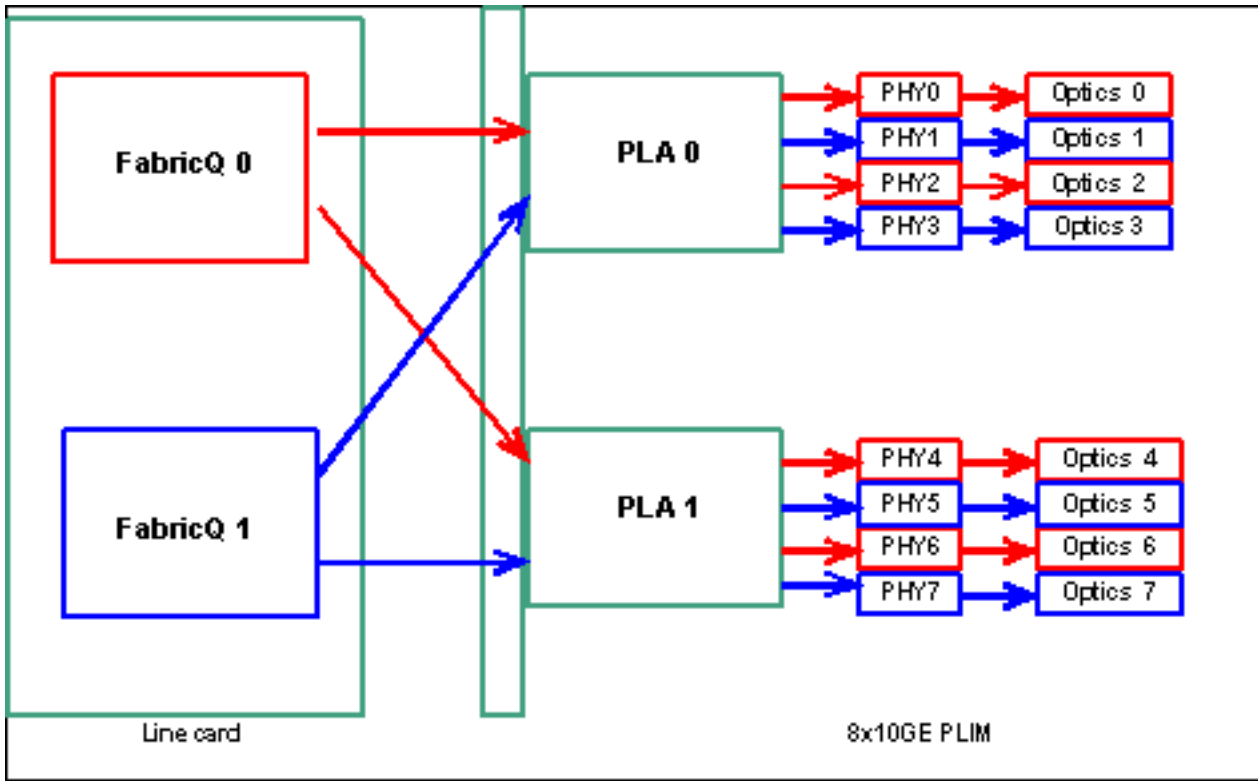
PLIM-overabonnement

10 GE PLIM

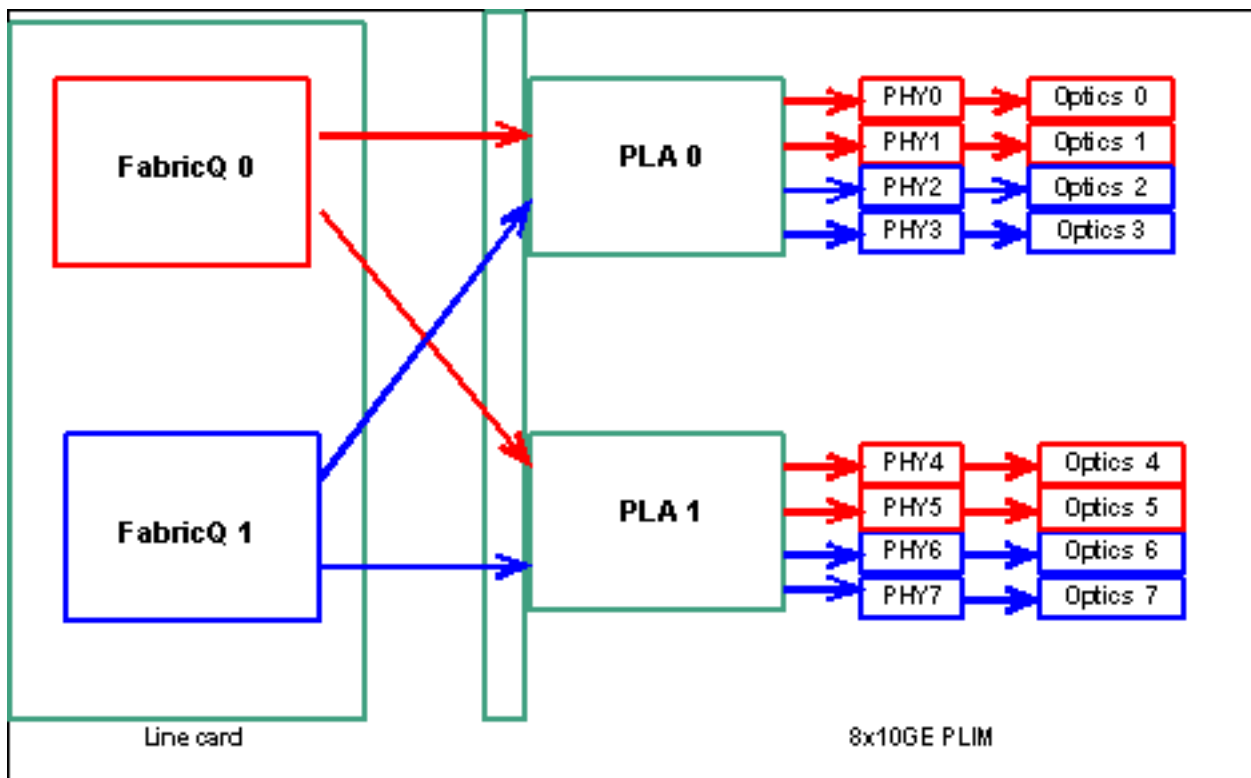
De 8 x 10G PLIM biedt de mogelijkheid om ongeveer 80 Gbps verkeer te beëindigen terwijl de verzendcapaciteit van de MSC maximaal 40 Gbps is. Als alle poorten die op de PLIM beschikbaar zijn, bevolkt zijn, vindt er overabonnement plaats en wordt de QoS-modellering extreem belangrijk om te voorkomen dat het premieverkeer onopzettelijk wordt ingetrokken. Voor sommigen is overinschrijving geen optie en moet dat vermeden worden. Hiervoor moeten slechts vier van de acht havens worden gebruikt. Daarnaast moet ervoor worden gezorgd dat de optimale bandbreedte binnen de MSC en PLIM beschikbaar is voor elk van de vier poorten.

Opmerking: de poortkarakterisering verandert vanaf release 3.2.2. Zie deze diagrammen.

Port mapping tot release 3.2.1



Poortbewaking vanaf release 3.2.2



Zoals eerder vermeld, worden de fysieke poorten onderhouden door één van de twee FabricQ ASIC's. De toewijzing van havens aan het ASIC is statistisch gedefinieerd en kan niet worden gewijzigd. Bovendien heeft de 8 x 10G PLIM twee PLA ASIC's. De eerste PLA-servicepoorten 0 tot 3, de tweede services 4 tot 7. De bandbreedte-capaciteit van één PLA op de 8 x 10G PLIM is ongeveer 24 Gbps. De switchcapaciteit van één FabricQ ASIC is ongeveer 62 Mpps.

Als u poort 0 tot 3 of poorten 4 tot 7 vult, wordt de bandbreedtecapaciteit van de PLA (24 Gbps) gedeeld tussen alle vier de poorten die de totale doorvoersnelheid beperken. Als u poorten 0,2,4 & 6 (tot 3.2.1) of 0,1,4 & 5 (3.2.2 vanaf) vult, omdat al deze poorten worden onderhouden door één FabricQ ASIC, waarvan de switchcapaciteit 62 Mpps is, opnieuw, wat de doorvoercapaciteit beperkt.

Het is het beste om de havens op een manier te gebruiken die de hoogste efficiëntie van zowel de PLA's als de FabricQ ASIC's verkrijgt om optimale prestaties te bereiken.

[SIP-telefoon 8800/SPA](#)

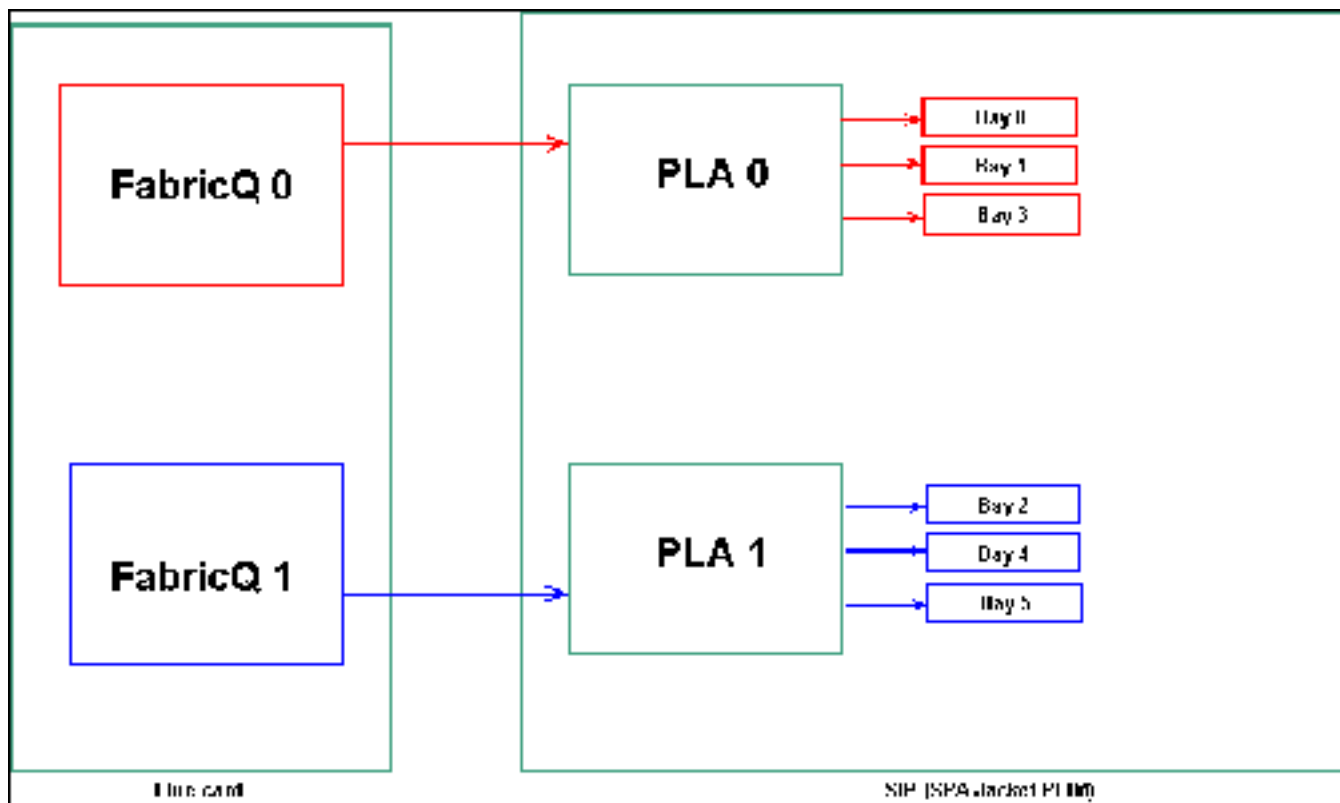
De SIP-800 PLIM biedt de mogelijkheid om met modulaire interfacekaarten te werken die bekend staan als Service Port Adapters (SPA's). SIP-800 biedt 6 SPA-bakken met een theoretische interfacecapaciteit van 60 Gbps. De transportcapaciteit van de MSC is maximaal 40 Gbps. Als alle bays van de SIP-800 zouden worden bevolkt, dan is het, afhankelijk van het SPA-type, mogelijk dat er te veel wordt geabonneerd en dat de QoS-modellen uiterst belangrijk worden om te voorkomen dat premieverkeer per ongeluk wordt ingetrokken.

Opmerking: overabonnement wordt niet ondersteund door POS-interfaces. Maar de plaatsing van

de 10 Gbps POS SPA moet geschikt zijn om de correcte doorvoercapaciteit te garanderen. De 10 Gbps Ethernet SPA wordt alleen ondersteund in IOS-XR release 3.4. Deze SPA biedt overabonnementsmogelijkheden.

Voor sommigen is overinschrijving geen optie en moet dat vermeden worden. Hiervoor hoeven slechts vier van de zes te worden gebruikt. Daarnaast moet ervoor worden gezorgd dat de optimale bandbreedte binnen de MSC en PLIM beschikbaar is voor elk van de vier poorten.

Toewijzing van SPA-borden



Zoals eerder vermeld, worden de fysieke poorten onderhouden door één van de twee FabricQ ASIC's. De toewijzing van havens aan het ASIC is statistisch gedefinieerd en kan niet worden gewijzigd. Bovendien heeft de SIP-800-PLIM twee PLA-ASIC's. De eerste PLA-dienstenpoorten 0,1 en 3, de tweede diensten 2, 4 en 5.

De bandbreedtecapaciteit van één enkele PLA op de SIP-800 PLIM is ongeveer 24 Gbps. De switchcapaciteit van één FabricQ ASIC is ongeveer 62 Mpps.

Als u poort 0,1 & 3 of poorten 2, 4 & 5 vult, wordt de bandbreedtecapaciteit van de PLA (24 Gbps) gedeeld tussen alle drie de poorten die de totale doorvoersnelheid beperken. Aangezien één FabricQ elk van de diensten die poortgroepen omvat, is het maximum pakkettafbeeldingsrate van de poortgroep 62 Mpps. Het is het beste om de havens te gebruiken op een manier die de hoogste efficiëntie van de PLA's verkrijgt om een optimale bandbreedte te bereiken.

Aanbevolen plaats:

	SPA900	SPA900	SPA900	SPA900
Optie 1	0	1	4	5
Optie 2	1	2	3	4

Indien u de kaart met meer dan vier SPA wilt bevolken, wordt aanbevolen één van de eerder genoemde opties te voltooien, die de interfaces tussen de twee poortgroepen (0,1 & 3 & 2,4 & 5) uitspreiden. U dient dan de volgende SPA-modules in een van de open poorten te plaatsen in de 0,1 & 3 & 2,4 & 5 poortgroepen.

[DWDM XENPACK's](#)

Vanaf release 3.2.2 kunnen DWDM XENPACK's worden geïnstalleerd en **tunable** glasvezelmodules worden geleverd. De koelvereisten van dergelijke XENPACK-modules vereisen dat er een blanco sleuf tussen de geïnstalleerde modules is. Als bovendien één DWDM XENPACK-module is geïnstalleerd, kunnen maximaal vier poorten worden gebruikt, zelfs als de XENPACK-modules geen DWDM-apparaten zijn. Dit heeft daarom een directe impact op FabricQ naar PLA om de poort in kaart te brengen. Er moet aandacht aan deze eis worden besteed en dit wordt in deze tabel besproken.

Aanbevolen plaats:

	Optische poort #	Optische poort #	Optische poort #	Optische poort #
Optie 1 of DWDM XENPACK	0	2	5	7
Optie 2	1	3	4	6

Vermijd de verandering van FabricQ mapping voor een installatie van 3.2.2 of hoger of 3.3. Een eenvoudiger plaatsingspatroon kan daarom worden gebruikt voor zowel de reguliere als de DWDM XENPACK-modules.

	Optische poort #	Optische poort #	Optische poort #	Optische poort #
Optie 1	0	2	4	6
Optie 2	1	3	5	7

Als u de kaart met meer dan vier niet-DWDM XENPACK-poorten wilt vullen, wordt aanbevolen één van de genoemde opties te voltooien, waarbij de Optische interfacemodules tussen de twee poortgroepen worden verdeeld (0-3 en 4-7). U moet dan de volgende Optische interfacemodules in een van de open havens in of de 0-3 of 4-7 poortgroepen plaatsen. Als u de 0-3 poortgroep voor Optische interfacemodule #5 gebruikt, moeten de Optische interfacemodules #6 in de 4-7 poortgroep worden geplaatst.

Raadpleeg de [DWDM XENPAK-modules](#) voor meer informatie.

[Configuratie-beheer](#)

De configuratie in IOS-XR wordt uitgevoerd door een configuratie in twee fasen. De configuratie wordt in het eerste stadium door de gebruiker ingevoerd. Dit is het stadium waarin alleen de syntaxis van de configuratie door de CLI wordt gecontroleerd. De configuratie die in dit stadium is ingevoerd, is alleen bekend bij het proces van de configuratieagent, bijvoorbeeld CLI/XML. De configuratie is niet geverifieerd omdat deze niet op de systeemserver is geschreven. De backend-toepassing wordt niet aangemeld en kan in deze fase geen toegang krijgen tot de configuratie of deze kennis ervan hebben.

In de tweede fase is de configuratie expliciet door de gebruiker vastgelegd. In deze fase wordt de configuratie geschreven op de systeemserver, de backend-toepassingen controleren of de configuraties en kennisgevingen door sysdb worden gegenereerd. U kunt een configuratiesessie afbreken voordat u de configuratie die in de eerste fase is ingevoerd, doorvoert. Het is dus niet veilig om aan te nemen dat alle configuratie die in stap één is ingevoerd altijd in fase twee is geëngageerd.

Daarnaast kunnen de bediening en/of de actieve configuratie van de router door meerdere gebruikers tijdens fase één en fase twee worden gewijzigd. Dus elke test van router die configuratie en/of operationele status in fase één draait zou niet geldig kunnen zijn in fase twee waar de configuratie daadwerkelijk is begaan.

[Configuratie-bestandssystemen](#)

Configuration File System (CFS) is een reeks bestanden en directories die worden gebruikt om de configuratie van de router op te slaan. CFS wordt opgeslagen onder de folder `disk0:/fig/`, de standaardmedia die op de RP worden gebruikt. Bestanden en mappen in CFS zijn intern voor de router en mogen nooit door de gebruiker worden gewijzigd of verwijderd. Dit kan resulteren in verlies of corruptie van de configuratie en de service beïnvloeden.

De CFS wordt gecontroleerd aan de standby-RP na elke verbintenis. Dit helpt het configuratiebestand van de router na een defect over te houden.

Tijdens het opstarten van de router, wordt de laatste actieve configuratie toegepast van de configuratie die in CFS opgeslagen moet worden. Het is niet nodig voor de gebruiker om de actieve configuratie handmatig op te slaan nadat elke configuratie zich heeft gecommitteerd, omdat dit automatisch door de router wordt gedaan.

Het is niet raadzaam de configuratie te wijzigen terwijl de configuratie wordt toegepast tijdens het opstarten. Als de configuratietoepassing niet voltooid is, ziet u dit bericht wanneer u op de router inlogt:

[Systeemconfiguratie](#)

De opstartconfiguratie van dit apparaat is momenteel het laden. Dit kan een paar minuten duren. U wordt op de hoogte gebracht na voltooiing. Probeer niet opnieuw te configureren het apparaat totdat dit proces is voltooid. In een paar zeldzame gevallen is het wellicht wenselijk om de routerconfiguratie te herstellen van een door een gebruiker opgegeven ASCII-configuratiebestand in plaats van de laatste actieve configuratie te herstellen van CFS.

U kunt de toepassing van een configuratiebestand forceren door:

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>
```

Tijdens het herstellen van de routerconfiguratie kunnen een of meer configuratieopties niet werken. Alle defecte configuratie wordt opgeslagen in de CFS en blijft behouden tot de volgende herlading.

U kunt door de mislukte configuratie bladeren, de fouten adresseren en de configuratie opnieuw toepassen.

Dit zijn enkele tips om de mislukte configuratie tijdens het opstarten van de router aan te pakken.

In IOX kan de configuratie om drie redenen worden geclassificeerd als defecte configuratie:

1. De parser genereert syntax fouten, die er doorgaans op wijzen dat er een oncompatibiliteit met de CLI-opdrachten is. U dient de syntaxisfouten te corrigeren en de configuratie opnieuw toe te passen.
2. Semantische fouten—semantische fouten worden gegenereerd door de backend componenten wanneer de configuratiemanager de configuratie tijdens het opstarten van de router herstelt. Het is belangrijk om op te merken dat cfgmgr niet verantwoordelijk is voor het waarborgen dat de configuratie wordt geaccepteerd als onderdeel van de configuratie. Cfgmgr is slechts een **middenman** en meldt alleen elke semantische mislukking die de backend-componenten genereren. Het is aan elke backend component-eigenaar om de oorzaak van de storing te analyseren en om de reden van de storing te bepalen. De gebruikers kunnen de **beschrijvende <CLI-opdrachten>** vanuit de configuratiemodus uitvoeren om zo eenvoudig de eigenaar van de backend-component-verificateur te vinden. Als de **router bgp 217** bijvoorbeeld als mislukte configuratie is weergegeven, toont de opdracht om te beschrijven dat de component verificateur de component ipv4-bgp component is.

```
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/mbil2000-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. Toepassen fouten-de configuratie is met succes geverifieerd en geaccepteerd als deel van de actieve configuratie maar de backend-component kan zijn operationele status niet om de een of andere reden bijwerken. De configuratie toont in beide actieve configuratie, aangezien het correct geverifieerd werd, en als mislukte configuratie wegens de backend operationele fout. De **beschrijvende** opdracht kan opnieuw worden uitgevoerd op de CLI die niet is toegepast om de component van toepassing te vinden eigenaar. Voltooi deze stappen om te bladeren en de mislukte configuratie opnieuw toe te passen tijdens opstartende ondernemingen: Voor R3.2 kunnen de exploitanten deze procedure gebruiken om de defecte configuratie opnieuw toe te passen: De exploitanten kunnen de **showconfiguratie mislukte opstartopdracht** gebruiken om door de mislukte configuratie te bladeren die tijdens routeropstarten is opgeslagen. De bedieners moeten de **showconfiguratie defecte opstartbeeld uitvoeren | bestand kan mislukt.cfg** opdracht om de opstartconfiguratie op te slaan, mislukte configuratie in een bestand. De exploitanten moeten naar de **configuratie-modus** gaan en opdrachten **laden/vastzetten** gebruiken om deze mislukte configuratie opnieuw toe te passen:

```
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
```

Voor R3.3-beeldexploitanten kunnen deze bijgewerkte procedure gebruiken: De exploitanten moeten de **showconfiguratie mislukte start** opdracht gebruiken en de **laadconfiguratie mislukte opstarten** opdracht gebruiken om door gefaalde configuratie te bladeren en opnieuw toe te passen.

```
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

```

Kernel Dumper

Standaard IOS-XR schrijft een kernstop naar de vaste schijf bij een procescrash, maar niet als de kern zelf crasht. Deze functionaliteit voor een meervoudig chassis wordt momenteel alleen ondersteund voor het lijnkaartchassis 0. Het andere chassis wordt ondersteund bij een toekomstige softwarerelease.

Aanbevolen wordt om kerneldumps voor zowel de RP's als de MSC's in staat te stellen met het gebruik van deze configuratie in zowel de standaard- als de admin-mode-configuraties:

```

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

```

Configuratie van moederpomp

Dit komt bij een ongeluk voor de rechter:

1. Een RP crashes en een stortplaats worden op de harddisk geschreven in die RP in de root folder van de disk.
2. Als een MSC crasht, wordt een dummy geschreven aan de harddisk van RP0 in de root folder van de schijf.

Dit heeft geen impact op RP failover tijden aangezien non-stop expanderen (NSF) voor de routingprotocollen wordt gevormd. Het kan een paar extra minuten duren voor de crashed RP of de lijnkaart weer beschikbaar komt na een crash terwijl het de kern schrijft.

Hier wordt een voorbeeld weergegeven van de toevoeging van deze configuratie aan zowel de standaard- als de admin-modus. Merk op dat voor de configuratie van de beheermodus de DRP's moeten worden gebruikt.

Deze uitvoer toont een voorbeeld van de configuratie van de pomp van 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)#
```

Security

LPTS

Local Packet Transport Services (LPTS) verwerkt lokaal bestemde pakketten. LPTS bestaat uit verschillende onderdelen.

1. Het belangrijkste proces wordt de "havenarbiter" genoemd. Het luistert naar socket verzoeken van verschillende protocolprocessen, bijvoorbeeld BGP, IS-IS, en houdt alle bindende informatie voor die processen bij. Bijvoorbeeld, als een BGP-proces op socket nummer 179 luistert, verkrijgt de PA die informatie van de BGP-processen, en wijst zij vervolgens een binding aan dat proces toe in een IFIB.
2. IFIB is een ander onderdeel van het LPTS-proces. Het helpt een directory te houden van waar een proces is dat luistert naar een specifieke havenbinding. Het IFIB wordt gegenereerd door het proces van de poortadapter en wordt bij de havenarbiter bewaard. Het genereert vervolgens meerdere subgroepen van deze informatie. De eerste subset is het segment van IFIB. Dit segment kan worden gekoppeld aan IPv4-protocol enzovoort. Splitsen worden vervolgens naar de juiste stroombeheerders gestuurd, die vervolgens het IFIB-segment gebruiken om het pakket naar het juiste proces door te sturen. De tweede subset is een pre-IFIB, zodat de LC het pakket naar het juiste proces kan doorsturen indien er slechts één proces bestaat of naar een juiste stroombeheerder.
3. Flow managers helpen de pakketten verder te distribueren als het omhoog kijken niet triviaal is, bijvoorbeeld, meerdere processen voor BGP. Elke stroombeheerder heeft een plakje of meerdere plakken van de IFIB en zendt pakketten naar behoren door naar de geëigende processen die verbonden zijn met de plak van IFIB.
4. Als een entry moet worden getypt voor de bestemmingspoort, dan kan deze worden geworpen of naar de stroommanager doorgestuurd. Een pakje wordt zonder bijbehorende poort doorgestuurd als er een gekoppeld beleid voor de poort is. De stroommanager helpt dan een nieuwe sessie te genereren.

Hoe wordt een intern pakket doorgestuurd?

Er zijn twee soorten stromen, Layer 2 (HDLC, PPP) en Layer 4 ICMP/PING-stromen en routingstromen.

1. Layer 2 HDLC/PPP-Deze pakketten worden geïdentificeerd door de protocolidentificatie en worden rechtstreeks naar de CPU-wachtrijen in de droger verzonden. Layer 2-protocolpakketten krijgen een hoge prioriteit en worden vervolgens door de CPU (via het netwerk) opgepikt en verwerkt. Daarom worden keepalives voor Layer 2 rechtstreeks via de LC via de CPU beantwoord. Dit vermijdt de noodzaak om naar de RP te gaan voor reacties en speelt in met het thema van gedistribueerd interfacebeheer.
2. ICMP (Layer 4)-pakketten worden in de LC ontvangen en worden via raadpleging door het

IFBI naar de CPU-wachtrijen op de droger verzonden. Deze pakketten worden vervolgens naar de CPU (via het netwerk) verzonden en verwerkt. Het antwoord wordt vervolgens door de wachtrijen van de spuitpers verzonden om door het weefsel te kunnen worden doorgestuurd. Voor het geval dat een andere toepassing ook de informatie nodig heeft (overgenomen door het weefsel). Eenmaal door de stof is het pakje bestemd voor de juiste opening van de stekker en voor de juiste spons- en controlewachtrij.

3. De routingstromen worden in IFIB opgezocht en vervolgens naar de uitvoervormingswachtrijen (8000 wachtrijen) verzonden waarvan er één is gereserveerd voor controlepakketten. Dit is een niet-gevormde rij en wordt simpelweg onderhouden telkens als het vol is. - hoge prioriteit. Het pakket wordt vervolgens door het weefsel verzonden in wachtrijen met hoge prioriteit naar een reeks CPU-wachtrijen in de spons (gelijk aan de wachtrijen bij het plein in de wasdroger), en dan verwerkt door het juiste proces, de stroombeheerder of het eigenlijke proces. Er wordt een antwoord teruggestuurd door de spons van de spons van de spar-lijnkaart en dan de lijnkaart. De spons van de rand LC heeft een speciale rij verlaten om controlepakketten te behandelen. De wachtrijen in de spons worden gesplitst in pakketten met hoge prioriteit, controle en lage prioriteit, per spits basis.
4. PSE heeft een reeks politiemensen die zijn ingesteld voor snelheidsbeperking op Layer 4, Layer 2 en het routeren van pakketten. Deze worden vooraf ingesteld en gewijzigd zodat ze op een latere datum door de gebruiker kunnen worden ingesteld.

Een van het meest algemene probleem met LPTS is pakketten die worden gedropt, wanneer u de router probeert te pingelen. De lpg-politie beperkt deze pakketten meestal. Dit is het geval om te bevestigen:

```
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	P00/5/1/0	0x3e	4/0
224.0.0.5 any	any	P00/5/1/1	0x3e	4/0

<further output elided>

IPsec

IP-pakketten zijn inherent onveilig. IPsec vormt een methode die wordt gebruikt om de IP-pakketten te beveiligen. CRS-1 IPsec wordt uitgevoerd in een software-verzendpad en daarom wordt de IPsec-sessie beëindigd op de RP/DRP. Een totaal aantal 500 IPsec-sessies per CRS-1 wordt ondersteund. Het nummer hangt af van de CPU-snelheid en de toegewezen middelen. Er is geen softwarebeperking aan dit, alleen lokaal-bron en lokaal-beëindigd verkeer op RP komen in aanmerking voor IPsec-behandeling. Ofwel kan de IPsec-transportmodus of de tunnelmodus worden gebruikt voor het type verkeer, maar de eerste is bij voorkeur veroorzaakt door minder overhead bij de verwerking van IPsec.

R3.3.0 ondersteunt de encryptie van BGP en OSPFv3 via IPsec.

Raadpleeg de [Cisco IOS XR-systeembeveiligingsgids](#) voor meer informatie over het implementeren van IPsec.

Opmerking: IPsec vereist crypto-taart, bijvoorbeeld hfr-k9sec-p.pie-3.3.1.

[Uit de band](#)

[Console en AUX-toegang](#)

CRS-1 RP/SC's hebben zowel een console- als een AUX-poort beschikbaar voor buitenbandbeheerdoeleinden, als een Ethernet-beheerpoort voor buitenband via IP.

De console en AUX poort van elke RP/SCGE, twee per chassis, kunnen worden aangesloten op een console server. Dit betekent dat voor het ene chassis 4 console-poorten nodig zijn en voor de multi-chassis systemen 12 poorten plus twee extra poorten voor de Supervisor Engines op Catalyst 6504-E.

De AUX poortverbinding is belangrijk aangezien het toegang tot de IOS-XR kern verleent en systeemherstel kan toestaan wanneer dit niet mogelijk is via de console poort. Toegang via de AUX poort is alleen beschikbaar voor gebruikers die lokaal gedefinieerd zijn in het systeem, en alleen wanneer de gebruiker root-system of cisco-support level-toegang heeft. Bovendien moet de gebruiker een **geheim** wachtwoord hebben gedefinieerd.

[Virtuele toegang voor terminalgebruikers](#)

Telnet & Secure shell (SSH) kan worden gebruikt om CRS-1 via de vty poorten te bereiken. Standaard worden beide uitgeschakeld en de gebruiker moet ze expliciet inschakelen.

Opmerking: IPsec vereist crypto-taart, bijvoorbeeld hfr-k9sec-p.pie-3.3.1.

genereert eerst RSA- en DSA-toetsen zoals in dit voorbeeld om SSH in te schakelen:

```
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
```

[Gerelateerde informatie](#)

- [Ondersteuning van routers](#)
- [Technische ondersteuning en documentatie – Cisco Systems](#)