Prácticas recomendadas operativas de CRS-1 e IOS XR

Contenido

Introducción

Este documento le ayuda a entenderlos:

- Procesos y subprocesos
- Fabric CRS-1
- Plano de Control
- Rommon y Monlib
- Módulo de interfaz de capa física (PLIM) y tarjeta de servicio modular (MSC)
- Administración de la Configuración
- Security
- Fuera de banda
- ‘Protocolo de administración de red simple (SNMP)

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento de Cisco IOS® XR.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Software Cisco IOS XR
- CRS-1

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Convenciones

Consulte Convenciones de Consejos TécnicosCisco para obtener más información sobre las convenciones del documento.

Descripción general de Cisco IOS XR

Cisco IOS XR está diseñado para ampliarse. El kernel es una arquitectura de Microkernel por lo que proporciona solamente servicios esenciales como administración de procesos, programación, señales y temporizadores. Todos los demás servicios, como sistemas de archivos, controladores, pilas de protocolos y aplicaciones, se consideran administradores de recursos y se ejecutan en el espacio de usuario protegido por memoria. Estos otros servicios se pueden agregar o quitar en tiempo de ejecución, lo que depende del diseño del programa. La huella de Microkernel es de sólo 12 kb. El microkernel y el sistema operativo subyacente son de sistemas de software QNX, y se denomina Neutrino. QNX se especializa en el diseño del sistema operativo en tiempo real. El microkernel es preventivo y el programador se basa en la prioridad. Esto asegura que el cambio de contexto entre procesos sea muy rápido y que los subprocesos de mayor prioridad siempre tengan acceso a la CPU cuando sea necesario. Estas son algunas de las ventajas de las que Cisco IOS XR se beneficia. Sin embargo, el mayor beneficio es el diseño inherente de las comunicaciones entre procesos dentro del núcleo de los sistemas operativos.

Neutrino es un sistema operativo que pasa mensajes, y los mensajes son los medios básicos de las comunicaciones entre procesos entre todos los subprocesos. Cuando un servidor determinado desea proporcionar un servicio, crea un canal para el intercambio de mensajes. Los clientes se conectan al canal de servidores asignando directamente al descriptor de archivo pertinente para utilizar el servicio. Todas las comunicaciones entre el cliente y el servidor son por el mismo mecanismo. Esto es una gran ventaja para una supercomputadora, que es CRS-1. Tenga en cuenta lo siguiente cuando se realiza una operación de lectura local en un kernel UNIX estándar:

- El software se interrumpe en el kernel.
- El núcleo se envía al sistema de archivos.
- Se reciben los datos.

Considere estos datos en el caso remoto:

- El software se interrumpe en el kernel.
- El núcleo envía NFS.
- NFS llama al componente de red.
- Remoto envía el componente de red.
- NFS se llama.
- Kernel envía el sistema de archivos.

La semántica para la lectura local y la lectura remota no son iguales. Los argumentos y parámetros para el bloqueo de archivos y los permisos de configuración son diferentes.

Considere el caso local de QNX:

- El software se interrumpe en el kernel.
- Kernel realiza el envío de mensajes al sistema de archivos.

Consideremos el caso no local:

- El software se interrumpe en el kernel.
- El núcleo entra en QNET, que es el mecanismo de transporte IPC.
- QNET entra en el kernel.
- Kernel envía el sistema de archivos.

Toda la semántica que se refiere al paso de argumentos y los parámetros del sistema de archivos son idénticos. Todo se ha disociado en la interfaz IPC que permite que el cliente y el servidor estén completamente separados. Esto significa que cualquier proceso puede ejecutarse en cualquier lugar y momento. Si un procesador de ruta determinado está demasiado ocupado atendiendo las solicitudes, puede migrar fácilmente esos servicios a una CPU diferente que se ejecute en un DRP. Una supercomputadora que ejecuta diferentes servicios en diferentes CPU diseminadas por múltiples nodos que pueden comunicarse fácilmente con cualquier otro nodo. La infraestructura está en marcha para ofrecer la oportunidad de ampliación. Cisco ha utilizado esta ventaja y ha escrito software adicional que engancha las operaciones principales del mensaje que pasa el núcleo que permite que el router CRS se amplíe a miles de nodos, donde un nodo, en este caso una CPU, ejecuta una instancia del SO, ya sea un proceso de ruta (RP), un procesador de ruta distribuido (DRP), una tarjeta de servicios modular (MSC) o un procesador de switch (SP).

Procesos y subprocesos

Dentro de los límites de Cisco IOS XR, un proceso es un área protegida de memoria que contiene uno o más subprocesos. Desde la perspectiva de los programadores, los subprocesos realizan el trabajo y cada uno completa una ruta de ejecución lógica para realizar una tarea específica. La

memoria que requieren los subprocesos durante el flujo de ejecución pertenece al proceso en el que operan, protegido de cualquier otro subproceso de procesos. Un subproceso es una unidad de ejecución, con un contexto de ejecución que incluye una pila y registros. Un proceso es un grupo de subprocesos que comparten un espacio de dirección virtual, aunque un proceso puede contener un único subproceso pero con más frecuencia contiene más. Si otro subproceso de un proceso diferente intenta escribir en la memoria en su proceso, el proceso ofensivo es eliminado. Si hay más de un subproceso que funciona dentro del proceso, ese subproceso tiene acceso a la misma memoria dentro del proceso y, como resultado, puede sobrescribir los datos de otro subproceso. Complete los pasos de un procedimiento para mantener la sincronización con los recursos para evitar este subproceso dentro del mismo proceso.

Un subproceso utiliza un objeto denominado Mutual Exclusion (MUTEX) para garantizar la exclusión mutua de los servicios. El subproceso que tiene MUTEX es el subproceso que puede escribir en un área determinada de memoria como ejemplo. Otros subprocesos que no tienen el MUTEX no pueden. También hay otros mecanismos para asegurar la sincronización con los recursos, y estos son Semáforos, Variables Condicionales o Condvares, Barreras y Sleepons. Estos no se discuten aquí, pero proporcionan servicios de sincronización como parte de sus tareas. Si equipara los principios aquí descritos con Cisco IOS, Cisco IOS es un proceso único que opera muchos subprocesos, con todos los subprocesos que tienen acceso al mismo espacio de memoria. Sin embargo, Cisco IOS llama a estos procesos de subprocesos.

Estados de proceso y subproceso

Dentro de Cisco IOS XR hay servidores que proporcionan los servicios y clientes que los utilizan. Un proceso determinado puede tener varios subprocesos que proporcionan el mismo servicio. Otro proceso puede tener varios clientes que podrían requerir un servicio determinado en cualquier momento. El acceso a los servidores no siempre está disponible, y si un cliente solicita acceso a un servicio se sienta allí y espera a que el servidor sea gratuito. En este caso, se dice que el cliente está bloqueado. Esto se denomina modelo de servidor cliente de bloqueo. El cliente puede estar bloqueado porque espera un recurso como MUTEX, o debido al hecho de que el servidor aún no ha respondido.

Ejecute un comando **show process ospf** para verificar el estado de los subprocesos en el proceso 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
```

Tenga en cuenta que el proceso ospf recibe una ID de trabajo (JID), que es 250. Esto nunca cambia en un router en ejecución y generalmente en una versión particular de Cisco IOS XR. Dentro del proceso ospf hay cinco subprocesos cada uno con su propio ID de subproceso (TID). Se muestra el espacio de pila para cada subproceso, la prioridad de cada subproceso y su estado.

Paso de mensaje síncrono

Se ha mencionado anteriormente que QNX es un sistema operativo que transmite mensajes. En realidad, se trata de un sistema operativo de transmisión de mensajes síncronos. Muchos de los problemas del sistema operativo se reflejan en la mensajería sincrónica. No se dice que el paso sincrónico de mensajes cause cualquier problema, sino que el síntoma del problema se refleja en el paso sincrónico de mensajes. Debido a que es sincrónica, el operador de CRS-1 puede acceder fácilmente a la información de estado o ciclo de vida, lo que ayuda en el proceso de solución de problemas. El mensaje que pasa el ciclo de vida es similar a esto:

- Un servidor crea un canal de mensajes.
- Un cliente se conecta al canal de un servidor (análogo a posix open).
- Un cliente envía un mensaje a un servidor (MsgSend) y espera una respuesta y bloques.
- El servidor recibe (MsgReceive) un mensaje de un cliente, procesa el mensaje y responde al cliente.
- El cliente desbloquea y procesa la respuesta del servidor.

Este modelo de servidor-cliente de bloqueo es el mensaje sincrónico que pasa. Esto significa que el cliente envía un mensaje y bloquea. El servidor recibe el mensaje, lo procesa, responde al cliente y luego el cliente desbloquea. Estos son los detalles específicos:

- El servidor espera en el estado RECEIVE.
- El cliente envía un mensaje al servidor y se BLOQUEA.
- El servidor recibe el mensaje y se desbloquea si espera en estado de recepción.
- El cliente pasa al estado REPLY (RESPUESTA).
- El servidor pasa al estado EJECUCIÓN.
- El servidor procesa el mensaje.
- El servidor responde al cliente.
- El cliente se desbloquea.

Ejecute el comando **show process** para ver en qué estados están el cliente y los servidores.

```
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
```

Estados de proceso y proceso bloqueados

Ejecute el comando **show process locked** para ver qué proceso está en estado bloqueado.

```
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#
```

La transmisión sincronizada de mensajes permite realizar un seguimiento sencillo del ciclo de vida de la comunicación entre procesos entre los diferentes subprocesos. En cualquier momento, un subproceso puede estar en un estado específico. Un estado bloqueado puede ser síntoma de un problema. Esto no significa que si un subproceso está en estado bloqueado, entonces haya un problema, así que no ejecute el comando **show process locked** y abra un caso con el Soporte Técnico de Cisco. Los subprocesos bloqueados también son muy normales.

Observe el resultado anterior. Si observa el primer subproceso de la lista, tenga en cuenta que es el ksh y que su respuesta está bloqueada en devc-conaux. El cliente, el ksh en este caso, envió un mensaje al proceso devc-conaux, el servidor, que es devc-conaux, mantiene la respuesta ksh bloqueada hasta que responde. Ksh es el shell UNIX que alguien utiliza en la consola o en el puerto AUX. Ksh espera la entrada desde la consola, y si no hay ninguna porque el operador no está tecleando, entonces permanece bloqueado hasta tal momento que procesa alguna entrada. Después del procesamiento, ksh vuelve a responder bloqueada en devc-conaux.

Esto es normal y no ilustra un problema. El punto es que los subprocesos bloqueados son normales, y depende de la versión XR, el tipo de sistema que tenga, lo que haya configurado y quién haga lo que altera el resultado del comando **show process locked**. El uso del comando **show process locked** es una buena manera de comenzar a resolver problemas de tipo de sistema operativo. Si hay un problema, por ejemplo la CPU es alta, entonces use el comando anterior para ver si algo se ve fuera de lo normal.

Entienda lo que es normal para su router en funcionamiento. Esto proporciona una línea de base que puede utilizar como comparación cuando resuelva los ciclos de vida del proceso.

En cualquier momento, un subproceso puede estar en un estado determinado. Esta tabla proporciona una lista de los estados:

Si el Estado:	El hilo de discusión es:
MUERTO	Muerto. El núcleo está esperando para liberar los recursos de subprocesos.
EJECUTÁNDOSE	Ejecución activa en una CPU
LISTO	No se está ejecutando en una CPU pero está listo para ejecutarse
DETENIDO	Suspendido (señal SIGSTOP)
ENVIAR	Esperando que un servidor reciba un mensaje
RECIBIR	Esperando que un cliente envíe un mensaje
RESPUESTA	Esperando que un servidor responda a un mensaje

PILA	Esperando a que se asigne más pila
PÁGINA DE ESPERA	Esperando a que el administrador de procesos resuelva un error de página
SIGSUSPENDER	Esperando una señal
SIGWAITINFO	Esperando una señal
NANOSLEEP	Dormido durante un período
MUTEX	En espera de adquirir un MUTEX
CONDVAR	Esperando a que se señale una variable condicional
INCORPORARSE	Esperando la finalización de otro subproceso
INTR	Esperando una interrupción
SEM	En espera de adquirir un semáforo

Procesos importantes y sus funciones

Cisco IOS XR tiene muchos procesos. Estas son algunas de las funciones importantes que se explican aquí.

Monitor del sistema WatchDog (WDSysmon)

Este es un servicio proporcionado para la detección de bloqueos de procesos y condiciones de memoria baja. La memoria baja puede ocurrir como resultado de una pérdida de memoria o de alguna otra circunstancia extraña. Un bloqueo puede ser el resultado de una serie de condiciones tales como interbloqueos de procesos, loops infinitos, bloqueos de kernel o errores de programación. En cualquier entorno multiproceso, el sistema puede entrar en un estado conocido como condición de bloqueo o simplemente un punto muerto. Puede producirse un bloqueo cuando uno o más subprocesos no pueden continuar debido a la contención de recursos. Por ejemplo, el subproceso A puede enviar un mensaje al subproceso B mientras que simultáneamente el subproceso B envía un mensaje al subproceso A. Ambos subprocesos se esperan entre sí y pueden estar en estado de envío bloqueado, y ambos subprocesos esperan para siempre. Se trata de un caso simple que implica dos subprocesos, pero si un servidor es responsable de un recurso que utilizan muchos subprocesos se bloquea en otro subproceso, los muchos subprocesos que solicitan acceso a ese recurso se pueden enviar bloqueados en espera en el servidor.

Los bloqueos pueden ocurrir entre unos pocos subprocesos, pero pueden afectar a otros. Los retrasos se evitan mediante un buen diseño del programa, pero con independencia de la magnificencia con que se diseñe y escriba un programa. A veces, una secuencia determinada de eventos que dependen de los datos con horarios específicos puede provocar un punto muerto. Los bloqueos no siempre son determinísticos y generalmente son muy difíciles de reproducir. WDSysmon tiene muchos subprocesos con uno que se ejecuta con la prioridad más alta que admite Neutrino, 63. La ejecución en la prioridad 63 garantiza que el subproceso obtenga tiempo de CPU en un entorno de programación preventiva basado en la prioridad. WDSysmon funciona con la función de vigilancia de hardware y supervisa los procesos de software que buscan condiciones de bloqueo. Cuando se detectan estas condiciones, WDSysmon recopila más información sobre la condición, puede indicar vaciar el proceso o el núcleo, escribir en syslogs, ejecutar scripts y matar los procesos bloqueados. Dependiendo de lo drástico que sea el problema, puede iniciar un switch del procesador de ruta para mantener el funcionamiento 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
```

El proceso WDSysmon tiene nueve subprocesos. Cuatro de ellos se sitúan en la prioridad 10, los otros cuatro en los puntos 11, 19, 20 y 63. Cuando se diseña un proceso, el programador considera cuidadosamente la prioridad que debe darse a cada subproceso del proceso. Como se ha comentado anteriormente, el planificador se basa en la prioridad, lo que significa que un subproceso de prioridad más alta siempre precede a uno de prioridad más baja. La prioridad 63 es la prioridad más alta a la que puede ejecutarse un subproceso, que es el subproceso 7 en este caso. El subproceso 7 es el subproceso que controla, el que realiza un seguimiento de los acaparamientos de la CPU. Debe ejecutarse con una prioridad más alta que los demás subprocesos que observa, de lo contrario podría no tener la oportunidad de ejecutarse en absoluto, lo que le impide seguir los pasos para los que se diseñó.

Netio

En Cisco IOS, existe el concepto de fast switching y process switching. Fast Switching utiliza el código CEF y se produce en el momento de la interrupción. El switching de procesos utiliza ip_input, que es el código de switching IP, y es un proceso programado. En las plataformas de mayor capacidad, el switching CEF se realiza en hardware y ip_input se programa en la CPU. El equivalente de ip_input en Cisco IOS XR es 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
```

Proceso de servicios de grupo (GSP)

Hay una necesidad de comunicación en cualquier supercomputadora con varios miles de nodos que cada uno ejecuta su propia instancia del núcleo. En Internet, una a muchas comunicaciones se realizan de manera eficiente a través de protocolos de multidifusión. GSP es el protocolo de multidifusión interno que se utiliza para IPC dentro de CRS-1. GSP proporciona una a muchas comunicaciones de grupo confiables que no tienen conexión con semántica asíncrona. Esto permite que el SGP se amplíe a los miles de nodos.

```
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
```

Descargador de contenido masivo BCDL

BCDL se utiliza para multicast datos confiables a varios nodos como RPs y MSCs. Utiliza el SGP como transporte subyacente. BCDL garantiza **el** envío de mensajes. Dentro de la BCDL hay un agente, un productor y un consumidor. El agente es el proceso que se comunica con el productor para recuperar y almacenar en búfer los datos antes de su multidifusión a los consumidores. El productor es el proceso que produce los datos que todo el mundo desea y el consumidor es el proceso interesado en recibir los datos facilitados por el productor. BCDL se utiliza durante las actualizaciones del software Cisco IOS XR.

Mensajería ligera (LWM)

LWM es una forma de mensajería creada por Cisco que se diseñó para crear una capa de abstracción entre las aplicaciones que se comunican entre procesos y Neutrino, con el objetivo de ser independiente del sistema operativo y de la capa de transporte. Si Cisco desea cambiar el proveedor de sistemas operativos de QNX a otro usuario, una capa de abstracción entre las funciones rudimentarias del sistema operativo subyacente ayuda a eliminar la dependencia del sistema operativo y ayuda a migrar a otro sistema operativo. LWM proporciona entrega de mensajes garantizada sincrónica, lo que, al igual que la transmisión de mensajes neutrinos nativos, hace que el remitente se bloquee hasta que el receptor responda.

LWM también proporciona entrega de mensajes asincrónica a través de pulsos de 40 bits. Los mensajes asincrónicos se envían de forma asincrónica, lo que significa que el mensaje se coloca en cola y el remitente no se bloquea, pero no se recibe de forma asincrónica por el servidor, sino cuando el servidor sondea para el siguiente mensaje disponible. LWM está estructurado como cliente/servidor. El servidor crea un canal que le da una **oreja** para escuchar los mensajes y se sienta en un momento en que el loop hace que un mensaje reciba escucha en el canal, que acaba de crear. Cuando un mensaje llega desbloquea y obtiene un identificador de cliente, que es efectivamente lo mismo que el ID de recepción del mensaje recibido. A continuación, el servidor realiza algún proceso y luego envía un mensaje de respuesta al identificador del cliente.

En el lado del cliente hace una conexión de mensaje. Se pasa un identificador al que se conecta y luego se envía un mensaje y se bloquea. Cuando el servidor finaliza el procesamiento, responde y el cliente se desbloquea. Esto es prácticamente lo mismo que el paso de mensajes nativos de Neutrinos, por lo que la capa de abstracción es muy delgada.

LWM está diseñado con un número mínimo de llamadas del sistema y switches de contexto para lograr un alto rendimiento, y es el método preferido de IPC en el entorno Cisco IOS XR.

Envmon

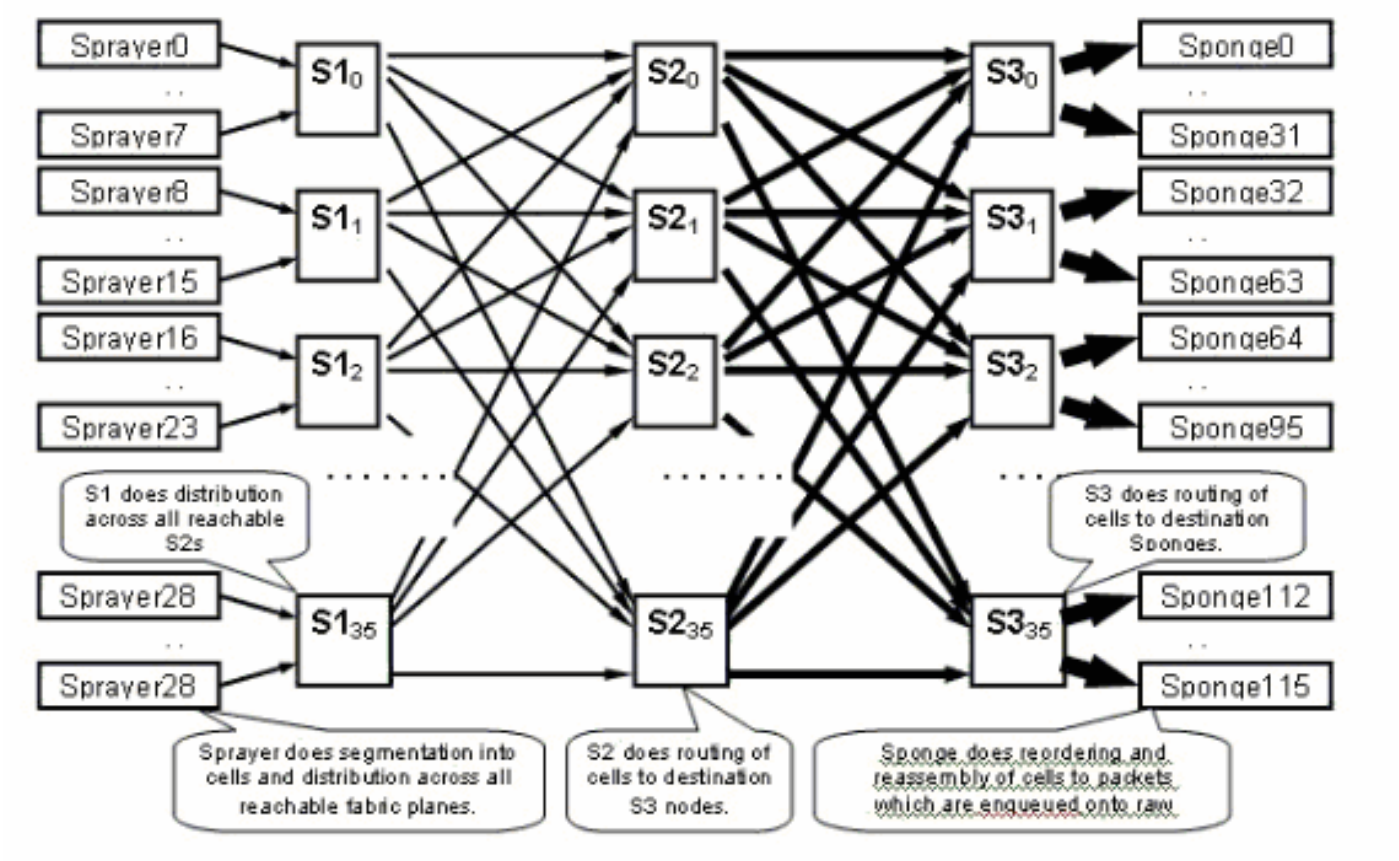
En el nivel más básico, el sistema de monitoreo ambiental es responsable de la advertencia cuando los parámetros físicos, por ejemplo la temperatura, el voltaje, la velocidad del ventilador, etc., caen fuera de los rangos de funcionamiento y cierran el hardware que se aproxima a niveles críticos donde el hardware podría resultar dañado. Supervisa periódicamente cada sensor de hardware disponible, compara el valor medido con los umbrales específicos de la tarjeta y provoca alarmas según sea necesario para realizar esta tarea. Un proceso persistente, iniciado en la inicialización del sistema, que sondea periódicamente todos los sensores de hardware, por ejemplo, el voltaje, la temperatura y la velocidad del ventilador, en el chasis y proporciona estos datos a los clientes de administración externos. Además, el proceso periódico compara las lecturas de los sensores con los umbrales de alarma y publica alertas ambientales en la base de datos del sistema para que el administrador de fallas tome las siguientes medidas. Si las lecturas de los sensores están peligrosamente fuera de alcance, el proceso de monitoreo ambiental podría hacer que la tarjeta se cierre.

Introducción al fabric de CRS-1

- Fabric en varias etapas: topología Benes en tres fases
- Routing dinámico dentro del fabric para minimizar la congestión
- Basado en celdas: celdas de 136 bytes, carga útil de datos de 120 bytes
- Control de flujo para mejorar el aislamiento del tráfico y minimizar los requisitos de almacenamiento en búfer en el fabric
- Entrega de velocidad de fase a fase
- Dos flujos de tráfico admitidos (unidifusión y multidifusión)
- Dos prioridades de tráfico admitidas por difusión (alta y baja)
- Compatibilidad con grupos de multidifusión de fabric (FGID) 1M
- Tolerancia a fallos rentable: Redundancia N+1 o N+k mediante planos de fabric, en lugar de 1+1 a un coste mucho mayor

Cuando se ejecuta en modo de chasis único, los ASIC S1, S2 y S3 se encuentran en las mismas tarjetas de fabric. Esta tarjeta también se conoce comúnmente como **tarjeta S123**. En una configuración de varios chasis, el S2 se separa y se encuentra en el chasis de tarjeta de fabric (FCC). Esta configuración requiere dos tarjetas de fabric para formar un plano, una tarjeta S2 y una tarjeta S13. Cada MSC se conecta a ocho planos de fabric para proporcionar redundancia de modo que, si pierde uno o más planos, el fabric todavía pasa el tráfico aunque el tráfico agregado, que puede pasar a través del fabric, sea inferior. El CRS todavía puede funcionar en línea para la mayoría de los tamaños de paquete con sólo siete planos. La contrapresión se envía sobre el fabric en un plano par e impar. No se recomienda ejecutar un sistema con menos de dos planos, en un plano par e impar. Cualquier valor inferior a dos planos no es una configuración admitida.

El plano de fabric



El diagrama anterior representa un plano. Debe multiplicar ese diagrama por ocho. Esto significa que el pulverizador (ingressq) asic de una LC se conecta a 8 S1s (1 S1 por plano). El S1 en cada plano de fabric se conecta a 8 pulverizadores:

- las 8 principales LC del chasis
- las 8 LC inferiores

Hay 16 S1s por chasis LC de 16 ranuras: 8 para las LC superiores (1 por plano) + 8 para las LC inferiores.

En un único chasis de 16 ranuras, una tarjeta de fabric S123 tiene 2 S1s, 2 S2 y 4 S3s. Esto forma parte del cálculo de velocidad del fabric. Hay el doble de tráfico, que puede salir del fabric de lo que puede entrar el tráfico. Actualmente también hay dos esponjas (fabricq) por LC en comparación con 1 pulverizador. Esto permite el almacenamiento en búfer en la LC de salida cuando más de una LC de ingreso sobrecarga una LC de salida. La LC de salida es capaz de absorber ese ancho de banda extra del entramado.

Supervisión de fabric

Conectividad y disponibilidad del plano:

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

Verifique si los planos están recibiendo/transmitiendo celdas y algunos errores están aumentando:

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

Acrónimos del comando anterior:

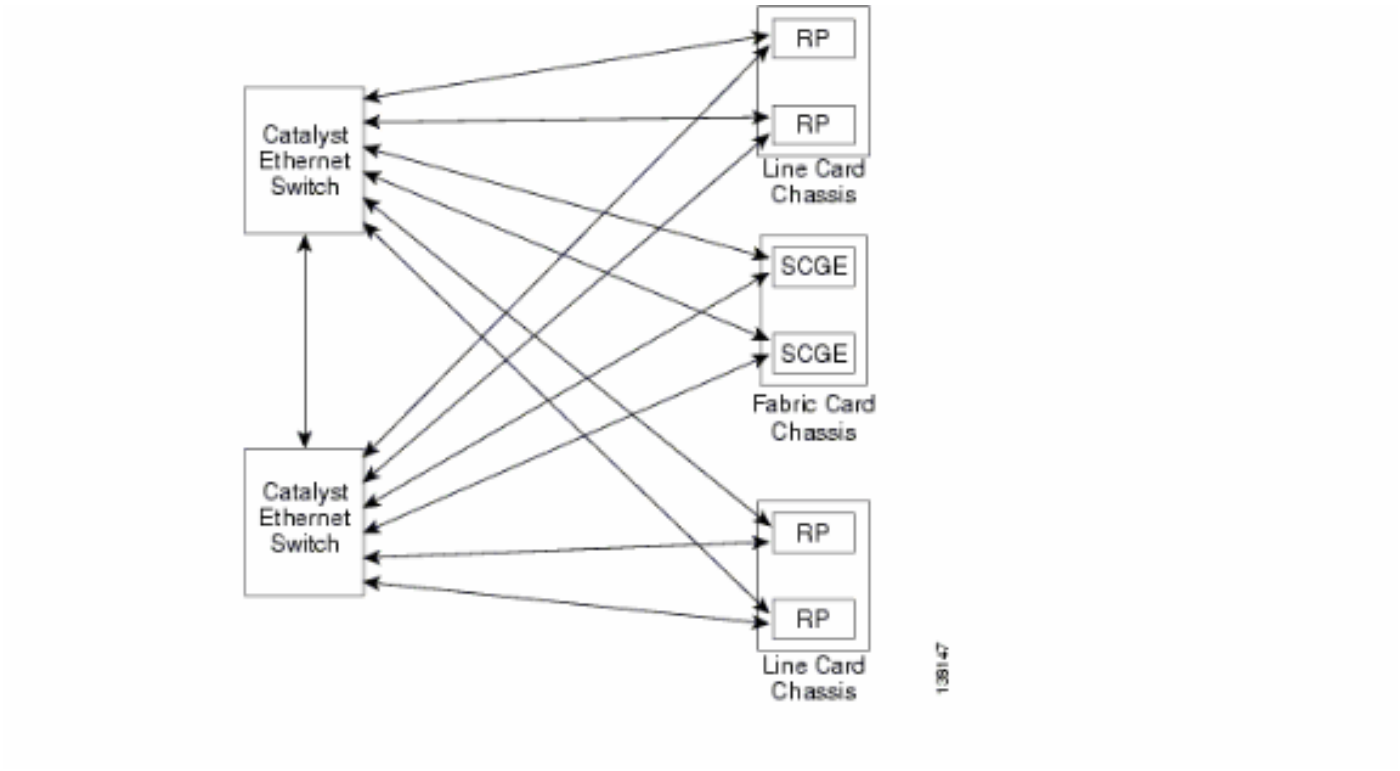
- CE: error corregible
- UCE: error incorregible
- PE: error de paridad

No se preocupe si observan algunos errores, ya que esto puede ocurrir en el inicio. Los campos no deben incrementarse en tiempo de ejecución. Si lo son, puede ser una indicación de un problema en el fabric. Ejecute este comando para obtener un desglose de los errores por plano de fabric:

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

Descripción general del plano de control

La conectividad del plano de control entre el chasis de tarjeta de línea y el chasis de fabric se realiza actualmente a través de puertos Gigabit Ethernet en los RP (LCC) y SCGE (FCC). La interconexión entre los puertos se proporciona a través de un par de switches Catalyst 6500, que se pueden conectar a través de dos o más puertos Gigabit Ethernet.



Configuración de Catalyst 6500

Esta es la configuración recomendada para los switches Catalyst utilizados para el plano de control de varios chasis:

- Se utiliza una sola VLAN en todos los puertos.
- Todos los puertos se ejecutan en modo de acceso (sin conexión troncal).
- El árbol de expansión 802.1w/s se utiliza para la prevención de loops.
- Se utilizan dos o más links para conectar los dos switches y el STP se utiliza para prevenir loops. No se recomienda el canal.
- Los puertos que se conectan al CRS-1 RP y SCGE utilizan el modo pre-estándar, ya que

IOS-XR no soporta los estándares 802.1s.
- El UDLD se debe habilitar en los puertos que se conectan entre los switches y entre los switches y el RP/SCGE.
- El UDLD se habilita de forma predeterminada en el CRS-1.

Consulte Cómo Subir el Cisco IOS XR Software en un Sistema Multicanal para obtener más información sobre cómo configurar un Catalyst 6500 en un Sistema Multicanal.

Gestión del plano de control de varios chasis

El chasis Catalyst 6504-E, que proporciona conectividad del plano de control para el sistema de varios chasis, se configura para estos servicios de administración:

- Administración en banda a través de puerto gigabit 1/2, que se conecta a un switch LAN en cada PoP. Sólo se permite el acceso a una pequeña gama de subredes y protocolos.
- NTP se utiliza para establecer la hora del sistema.
- El registro del sistema se realiza en los hosts estándar.
- El sondeo y las trampas SNMP se pueden habilitar para las funciones críticas.

Nota: No se deben realizar cambios en el Catalyst en funcionamiento. Las pruebas previas deben realizarse en cualquier cambio planificado y se recomienda encarecidamente que se realice durante una ventana de mantenimiento.

Este es un ejemplo de configuración de administración:

```
#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 y Monlib

Cisco monlib es un programa ejecutable que se almacena en el dispositivo y se carga en la RAM para su ejecución por ROMMON. ROMMON utiliza monlib para acceder a los archivos del dispositivo. Las versiones de ROMMON se pueden actualizar y se deben realizar bajo recomendación del Soporte Técnico de Cisco. La última versión de ROMMON es 1.40.

Instrucciones de actualización

Complete estos pasos:

1. Descargue los binarios ROMMON de Cisco CRS-1 ROMMON (sólo clientes registrados) .
2. Desempaquete el archivo TAR y copie los 6 archivos BIN en el directorio raíz CRS de 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. Utilice el comando **show diag | inc ROM|NODE|PLIM** para ver la versión actual de rommon.

```
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. Vaya al modo ADMIN y use el comando **upgrade rommon a all disk0** para actualizar el ROMMON.

```
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. Salga del modo ADMIN e ingrese **show log | inc "OK, ROMMON A"** y asegúrese de que todos los nodos se hayan actualizado correctamente. Si alguno de los nodos falla, vuelva al paso 4 y reprograme.

```
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. Vaya al modo ADMIN y use el comando **upgrade rommon b all disk0** para actualizar el ROMMON.

```
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. Salga del modo ADMIN e ingrese **show log | inc "OK, ROMMON B"** y asegúrese de que todos los nodos se hayan actualizado correctamente. Si alguno de los nodos falla, vuelva al paso 4 y reprograme.

```
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. El comando **upgrade** acaba de quemar una sección especial reservada de bootflash con el nuevo ROMMON. Pero el nuevo ROMMON permanece inactivo hasta que se recarga la tarjeta. Así que cuando recargue la tarjeta, el nuevo ROMMON estará activo. Reinicie cada nodo uno a la vez o simplemente reinicie todo el router para hacer esto.

```
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. Utilice el comando **show diag | inc ROM|NODE|PLIM** para verificar la versión ROMMON actual.

```
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: En CRS-8 y chasis de fabric, ROMMON también establece las velocidades del ventilador en la velocidad predeterminada de 4000 RPM.

Descripción general de PLIM y MSC

Esto representa el flujo de paquetes en el router CRS-1 y estos términos se utilizan indistintamente:

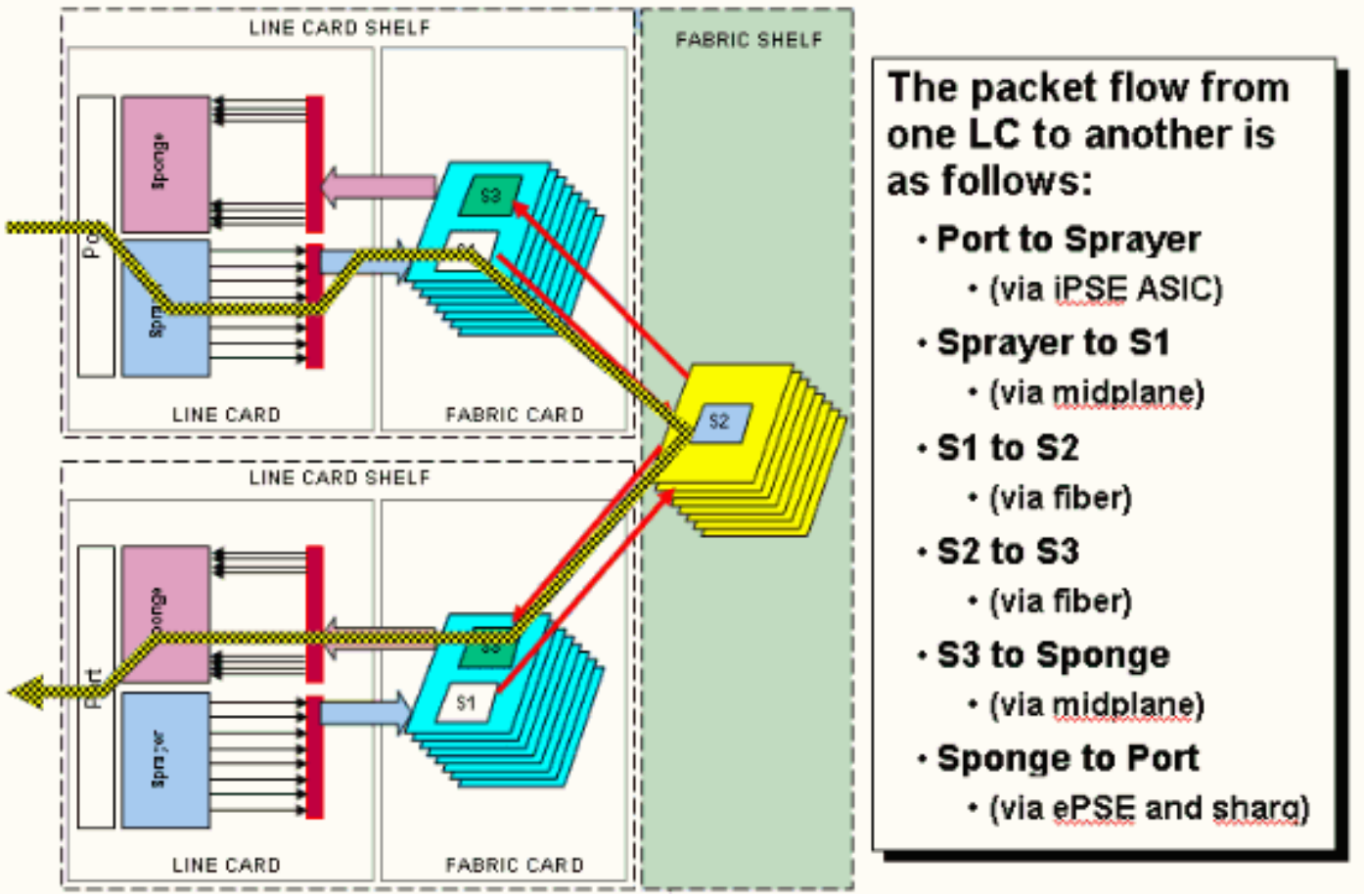
IngressQ ASIC también se denomina Sprayer ASIC.

FabricQ ASIC también se denomina ASIC de esponja.

El ASIC EgressQ también se denomina ASIC Sharq.

El SPP también se denomina ASIC (Packet Switch Engine).

Rx PLIM > Rx SPP > Cola de entrada > Fabric > Cola de fabric > Tx SPP > Cola de salida > Tx PLIM (Sprayer) (Esponja) (Sharq)

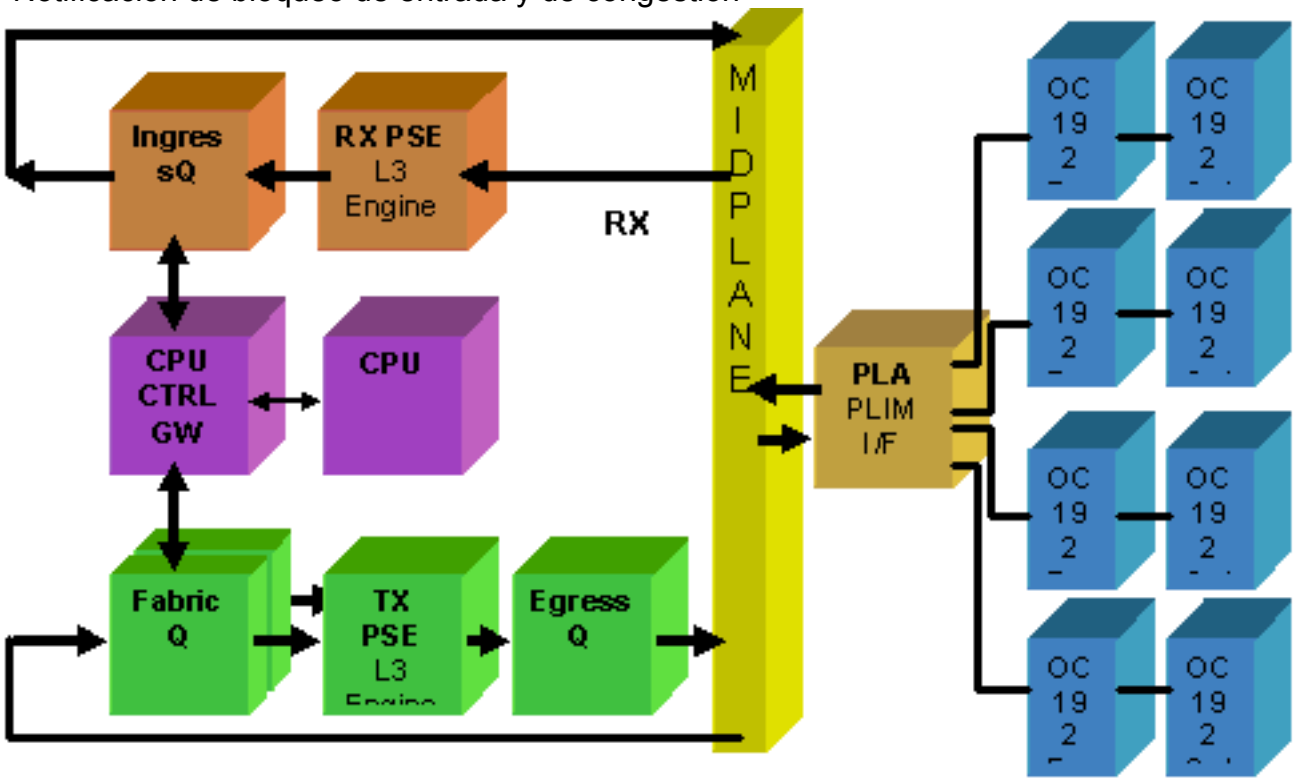


Los paquetes se reciben en el módulo de interfaz de capa física (PLIM).

El PLIM contiene las interfaces físicas para el MSC con el que se asocia. PLIM y MSC son tarjetas independientes conectadas a través de la placa de interconexiones del chasis. Como resultado, los tipos de interfaz para un MSC determinado se definen por el tipo de PLIM con el que apareó. Según el tipo de PLIM, la tarjeta contiene varios ASIC que proporcionan el medio físico y el entramado para las interfaces. El propósito de los ASIC PLIM es proporcionar la interfaz entre el MSC y las conexiones físicas. Termina la fibra, hace la luz a la conversión eléctrica, termina el entramado de medios siendo SDH/Sonet/Ethernet/HDLC/PPP, verifica el CRC, agrega cierta información de control llamada encabezado del búfer y reenvía los bits que permanecen en el MSC. El PLIM no origina/hunde las señales de mantenimiento HDLC o PPP. Estos son manejados por la CPU en el MSC.

El PLIM también proporciona estas funciones:

- Filtrado de MAC para 1/10 Gigabit Ethernet
- Contabilidad MAC de entrada/salida para 1/10 Gigabit Ethernet
- Filtrado de VLAN para 1/10 Gigabit Ethernet
- Contabilización de VLAN para 1/10 Gigabit Ethernet
- Notificación de bloqueo de entrada y de congestión



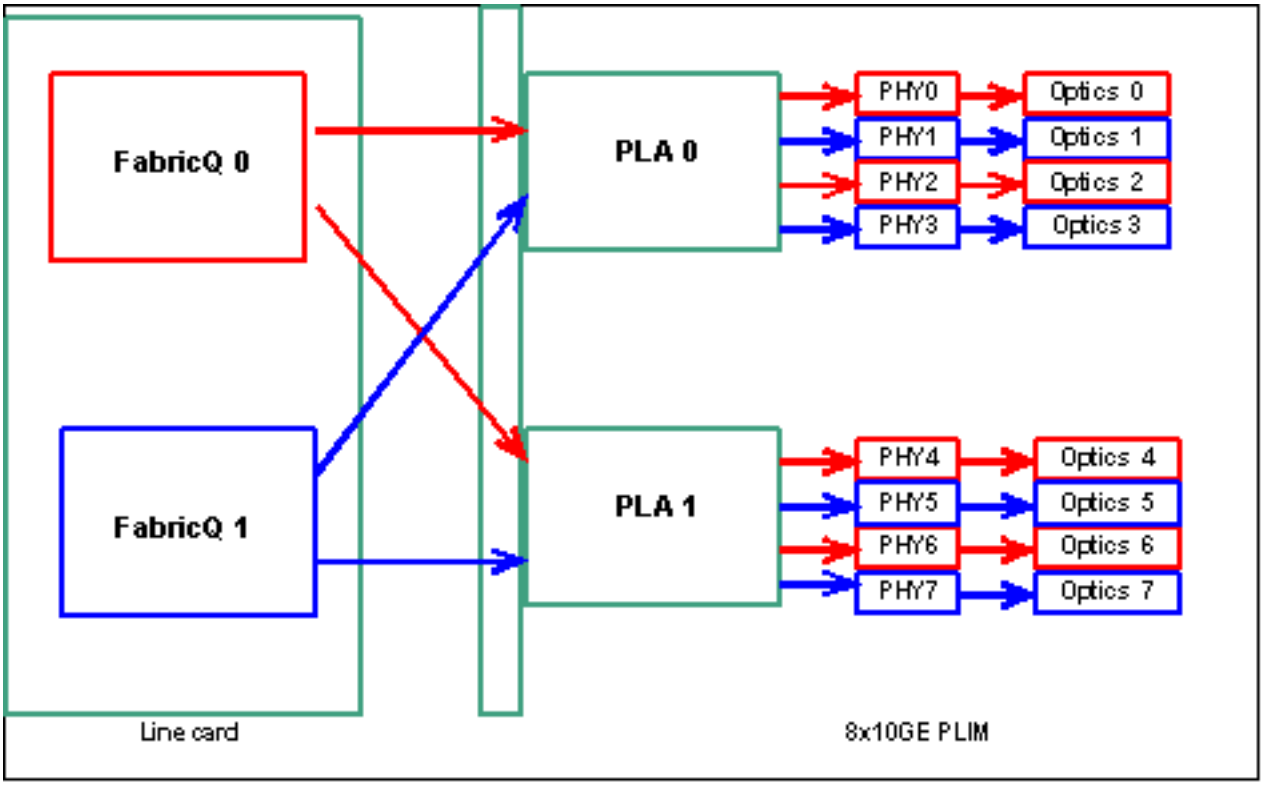
Sobresuscripción de PLIM

PLIM DE 10 GE

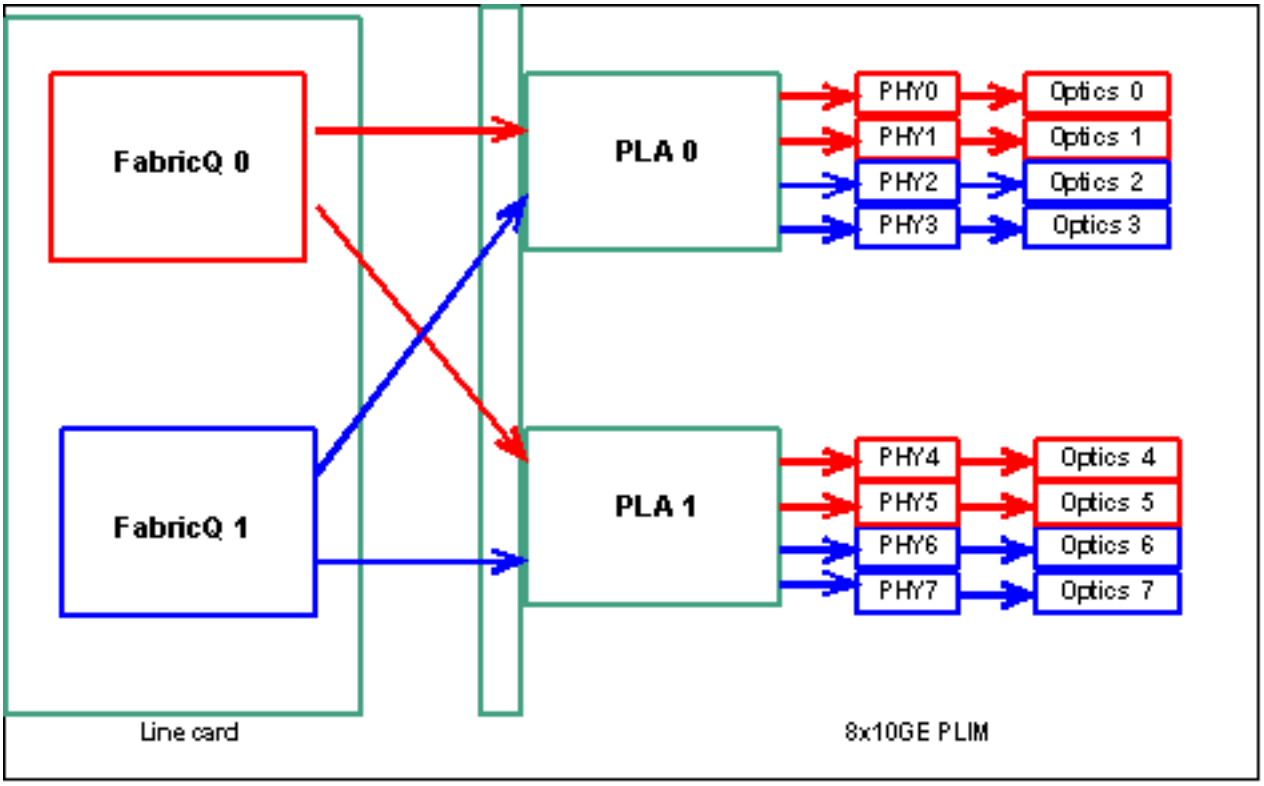
El PLIM 8 X 10 G ofrece la capacidad de finalizar aproximadamente 80 Gbps de tráfico mientras que la capacidad de reenvío de la MSC es un máximo de 40 Gbps. Si todos los puertos disponibles en el PLIM están llenos, se produce una sobresuscripción y el modelado de QoS se vuelve extremadamente importante para garantizar que el tráfico premium no se descarte inadvertidamente. Para algunos, la suscripción excesiva no es una opción y debe evitarse. Para hacer esto, sólo se deben utilizar cuatro de los ocho puertos. Además, se debe tener cuidado de asegurar que el ancho de banda óptimo dentro de MSC y PLIM esté disponible para cada uno de los cuatro puertos.

Nota: La asignación de puertos cambia de la versión 3.2.2 en adelante. Vea estos diagramas.

Asignación de puertos hasta la versión 3.2.1



Asignación de puertos desde la versión 3.2.2 en adelante



Como se mencionó anteriormente, uno de los dos ASIC de FabricQ presta servicio a los puertos físicos. La asignación de puertos al ASIC se define de forma estática y no se puede modificar. Además, el PLIM 8 X 10G tiene dos ASIC PLA. Los primeros puertos de servicios del ELP 0 a 3, los segundos servicios 4 a 7. La capacidad de ancho de banda de un único PLA en el PLIM de 8 x 10 G es de aproximadamente 24 Gbps. La capacidad de switching de un único FabricQ ASIC es de aproximadamente 62 Mpps.

Si rellena el puerto 0 a 3 o los puertos 4 a 7, la capacidad de ancho de banda del PLA (24 Gbps) se comparte entre los cuatro puertos que restringen el rendimiento total. Si rellena los puertos 0, 2, 4 y 6 (hasta 3.2.1) o 0, 1, 4 y 5 (a partir de 3.2.2), ya que todos estos puertos son atendidos por el único ASIC de FabricQ, cuya capacidad de switching es de 62 Mpps, de nuevo, lo que restringe la capacidad de rendimiento.

Es mejor utilizar los puertos de una manera que obtenga la mayor eficiencia tanto de los PLA como de los ASIC de FabricQ para lograr un rendimiento óptimo.

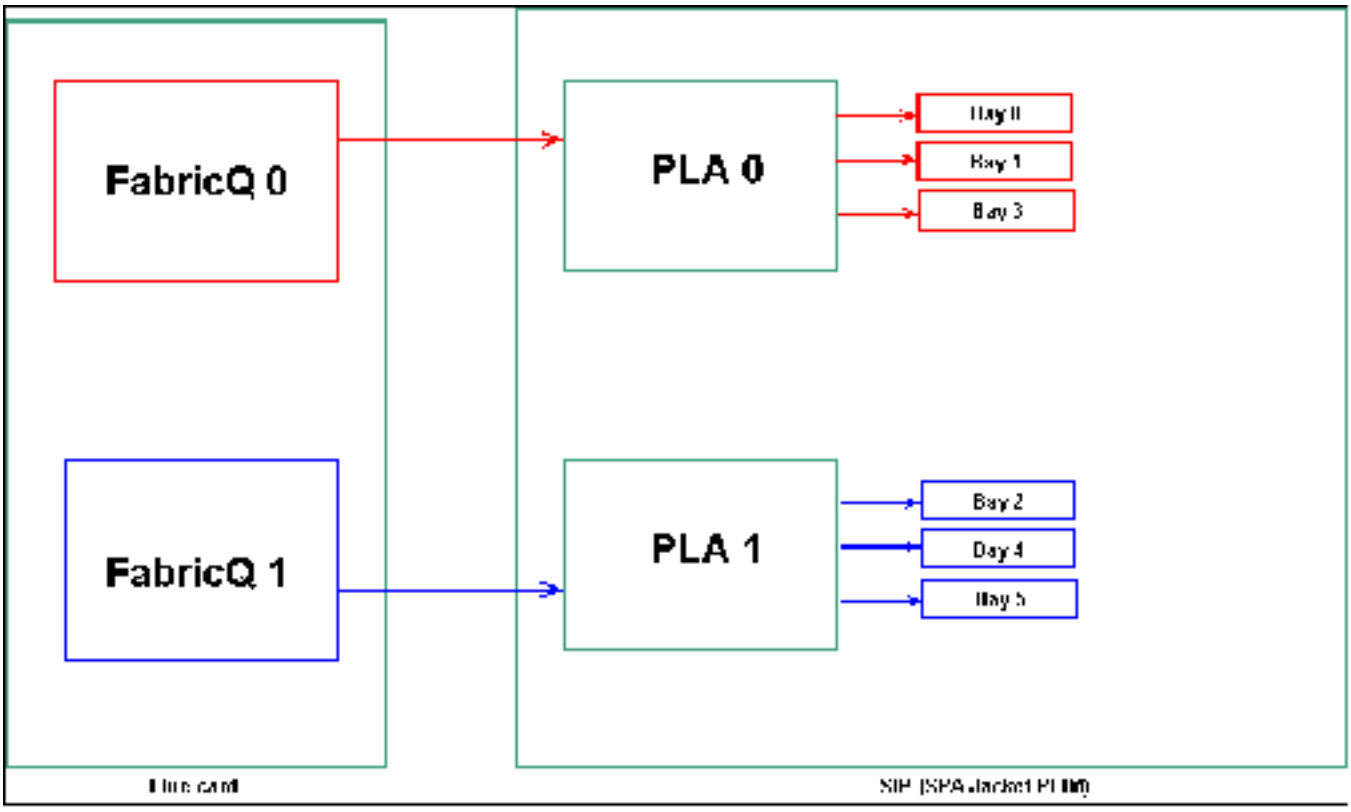
SIP-800/SPA

El PLIM SIP-800 ofrece la posibilidad de funcionar con tarjetas de interfaz modulares conocidas como adaptadores de puerto de servicio (SPA). El SIP-800 proporciona 6 bahías SPA con una capacidad de interfaz teórica de 60 Gbps. La capacidad de reenvío de la MSC es un máximo de 40 Gbps. Si se cumplimentaran todas las bahías del SIP-800, entonces, dependiendo del tipo de SPA, es posible que se produzca una sobresuscripción y que el modelado de QoS se vuelva extremadamente importante para garantizar que el tráfico premium no se descarte inadvertidamente.

Nota: La sobresuscripción no se soporta con las interfaces POS. Sin embargo, la ubicación del SPA POS de 10 Gb debe ser apropiada para garantizar la capacidad de rendimiento correcta. El SPA Ethernet de 10 Gb sólo se admite en la versión 3.4 del IOS-XR. Este SPA ofrece capacidades de sobresuscripción.

Para algunos, la suscripción excesiva no es una opción y debe evitarse. Sólo se deben utilizar cuatro de las seis bahías para hacer esto. Además, se debe tener cuidado para asegurar que el ancho de banda óptimo dentro de MSC y PLIM esté disponible para cada uno de los cuatro puertos.

[Asignación de bahía SPA]



Como se mencionó anteriormente, uno de los dos ASIC de FabricQ presta servicio a los puertos físicos. La asignación de puertos al ASIC se define de forma estática y no se puede modificar. Además, el PLIM SIP-800 tiene dos ASIC PLA. Los primeros puertos de servicios PLA 0,1 y 3, los segundos servicios 2, 4 y 5.

La capacidad de ancho de banda de un único PLA en el PLIM SIP-800 es de aproximadamente 24 Gbps. La capacidad de switching de un único FabricQ ASIC es de aproximadamente 62 Mpps.

Si rellena los puertos 0, 1 y 3 o los puertos 2, 4 y 5, la capacidad de ancho de banda del PLA (24 Gbps) se comparte entre los tres puertos que restringen el rendimiento total. Dado que una única FabricQ presta servicios a esos grupos de puertos, la velocidad máxima de paquetes del grupo de puertos es de 62 Mpps. Es mejor utilizar los puertos de una manera que obtenga la mayor eficiencia de los PLA para lograr un ancho de banda óptimo.

Ubicación sugerida:

	N.º de bahía de SPA	N.º de bahía de SPA	N.º de bahía de SPA	N.º de bahía de SPA
Opción 1	0	1	4	5
Opción 2	1	2	3	4

Si desea rellenar la tarjeta con más de cuatro SPA, la recomendación es completar una de las opciones previamente enumeradas, que distribuye las interfaces entre los dos grupos de puertos (0,1 y 3, y 2,4 y 5). A continuación, debe colocar los siguientes módulos SPA en uno de los puertos abiertos en los grupos de puertos 0, 1, 3, 2, 4 y 5.

DWDM XENPACK

A partir de la versión 3.2.2, los DWDM XENPACK se pueden instalar y proporcionar módulos **ajustables de** óptica. Los requisitos de refrigeración de estos módulos XENPACK requieren que haya una ranura en blanco entre los módulos instalados. Además, si se instala un solo módulo DWDM XENPACK, se puede utilizar un máximo de cuatro puertos, incluso si los módulos XENPACK no son dispositivos DWDM. Esto, por lo tanto, tiene un impacto directo en la asignación de FabricQ a PLA a puerto. Hay que prestar atención a este requisito y se examina en este cuadro.

Ubicación sugerida:

	Número de puerto óptico	Número de puerto óptico	Número de puerto óptico	Número de puerto óptico
Opción 1 o DWDM XENPACK	0	2	5	7
Opción 2	1	3	4	6

Para una instalación 3.2.2 o posterior o 3.3, evite el cambio de asignación de FabricQ. Por lo tanto, se puede utilizar un patrón de ubicación más sencillo tanto para los módulos XENPACK DWDM como para los módulos XENPACK DWDM.

	Número de puerto óptico	Número de puerto óptico	Número de puerto óptico	Número de puerto óptico
Opción 1	0	2	4	6
Opción 2	1	3	5	7

Si desea rellenar la tarjeta con más de cuatro puertos XENPACK que no sean DWDM, la recomendación es completar una de las opciones enumeradas, que propaga los módulos de

interfaz óptica entre los dos grupos de puertos (0-3 y 4-7). A continuación, debe colocar los siguientes módulos de interfaz óptica en uno de los puertos abiertos en los grupos de puertos 0-3 o 4-7. Si utiliza el grupo de puertos 0-3 para el módulo de interfaz óptica n.º 5, los módulos de interfaz óptica n.º 6 deben ubicarse en el grupo de 4-7 puertos.

Consulte Módulos DWDM XENPAK para obtener más detalles.

Administración de la Configuración

La configuración en IOS-XR se realiza a través de una configuración de dos etapas; el usuario ingresa la configuración en la primera etapa. Esta es la etapa en la que la CLI sólo verifica la sintaxis de la configuración. La configuración ingresada en esta etapa sólo es conocida por el proceso del agente de configuración, por ejemplo, CLI/XML. La configuración no se verifica porque no se escribe en el servidor sysdb. La aplicación backend no se notifica y no puede acceder a la configuración o tener conocimiento alguno sobre ella en esta etapa.

En la segunda etapa, el usuario se compromete explícitamente a la configuración. En esta etapa, la configuración se escribe en el servidor sysdb, las aplicaciones backend verifican las configuraciones y las notificaciones que genera sysdb. Puede anular una sesión de configuración antes de confirmar la configuración introducida en la primera etapa. Por lo tanto, no es seguro asumir que toda la configuración ingresada en la etapa uno siempre se compromete en la etapa dos.

Además, el funcionamiento y/o la configuración en ejecución del router pueden ser modificados por varios usuarios durante la etapa uno y la etapa dos. Por lo tanto, cualquier prueba del router que ejecute la configuración y/o el estado operativo en la etapa uno podría no ser válida en la etapa dos donde la configuración se compromete realmente.

Sistemas de archivos de configuración

Configuration File System (CFS) es un conjunto de archivos y directorios utilizados para almacenar la configuración del router. CFS se almacena en el directorio disk0:/config/, que es el medio predeterminado utilizado en el RP. Los archivos y directorios en CFS son internos del router y el usuario nunca los debe modificar ni eliminar. Esto puede provocar la pérdida o corrupción de la configuración y afectar al servicio.

El CFS se verifica en el standby-RP después de cada confirmación. Esto ayuda a preservar el archivo de configuración del router después de una conmutación por error.

Durante el arranque del router, la última configuración activa se aplica desde la base de datos de confirmación de configuración almacenada en CFS. No es necesario que el usuario guarde manualmente la configuración activa después de cada confirmación de configuración, ya que el router lo hace automáticamente.

No se recomienda realizar cambios en la configuración mientras se aplica la configuración durante el inicio. Si la aplicación de configuración no está completa, verá este mensaje cuando inicie sesión en el router:

Proceso de configuración del sistema

La configuración de inicio de este dispositivo se está cargando actualmente. Esto puede tardar

unos minutos. Se le notificará al finalizar. No intente volver a configurar el dispositivo hasta que se complete este proceso. En algunos casos excepcionales, podría ser deseable restaurar la configuración del router desde un archivo de configuración ASCII proporcionado por el usuario en lugar de restaurar la última configuración activa de CFS.

Puede forzar la aplicación de un archivo de configuración mediante:

```
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>
```

Mientras restaura la configuración del router, es posible que uno o más elementos de configuración no surtan efecto. Toda la configuración fallida se guarda en el CFS y se mantiene hasta la siguiente recarga.

Puede examinar la configuración fallida, solucionar los errores y volver a aplicar la configuración.

Estos son algunos consejos para resolver la configuración fallida durante el inicio del router.

En IOX, la configuración se puede clasificar como configuración fallida por tres razones:

1. Errores de sintaxis: el analizador genera errores de sintaxis, que normalmente indican que hay una incompatibilidad con los comandos CLI. Debe corregir los errores de sintaxis y volver a aplicar la configuración.
2. Errores semánticos: los componentes backend generan errores semánticos cuando el administrador de configuración restaura la configuración durante el inicio del router. Es importante tener en cuenta que cfgmgr no es responsable de garantizar que la configuración se acepte como parte de la configuración en ejecución. Cfgmgr no es más que un intermediario y sólo informa de cualquier falla semántica que generen los componentes backend. Depende de cada propietario del componente backend analizar la razón de la falla y determinar la razón de la falla. Los usuarios pueden ejecutar la descripción <comandos CLI> desde el modo de configuración para encontrar fácilmente el propietario del verificador del componente backend. Por ejemplo, si el router bgp 217 aparece como configuración fallida, el comando describe muestra que el verificador de componentes es el 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. Aplicar errores: la configuración se ha verificado y aceptado correctamente como parte de la configuración en ejecución, pero el componente backend no puede actualizar su estado operativo por alguna razón. La configuración se muestra en ambas configuraciones en ejecución, ya que se verificó correctamente y como configuraciones fallidas debido al error de funcionamiento del motor. El comando **description** se puede ejecutar nuevamente en la CLI que no se pudo aplicar para encontrar el propietario de aplicación del componente.Complete estos pasos para examinar y volver a aplicar la configuración fallida durante los operadores de inicio:Para los operadores R3.2 pueden utilizar este procedimiento para volver a aplicar la configuración fallida:Los operadores pueden utilizar el comando **show configuration failed startup** para examinar la configuración fallida guardada durante el inicio del router.Los operadores deben ejecutar el **error show configuration failed startup | file myfailed.cfg** para guardar la configuración de inicio fallida en un archivo.Los operadores deben ir al modo de **configuración** y utilizar los comandos **load/commit** para volver a aplicar esta configuración fallida:

```
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
```

Para imágenes R3.3, los operadores pueden utilizar este procedimiento actualizado:Los operadores deben utilizar el comando **show configuration failed startup** y el comando **load configuration failed startup** para examinar y volver a aplicar cualquier configuración fallida.

```
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
```

Vaciado del núcleo

De forma predeterminada, IOS-XR escribe un vaciado de memoria en el disco duro en caso de que se produzca un desperfecto en el proceso, pero no si el propio kernel se desmorona. Tenga en cuenta que para un sistema de varios chasis, esta funcionalidad sólo se admite actualmente para el chasis de tarjeta de línea 0. El otro chasis se admite en una futura versión de software.

Se sugiere que los vaciados de kernel para los RPs y los MSCs se habiliten con el uso de esta configuración tanto en las configuraciones estándar como en las de modo administrador:

```
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
```

Configuración de volcado del núcleo

Esto da como resultado esta ocurrencia para un desperfecto del kernel:

1. Un RP se bloquea y se escribe un volcado en el disco duro de ese RP en el directorio raíz del disco.
2. Si un MSC falla, se escribe un volcado en el disco duro del RP0 en el directorio raíz del disco.

Esto no afecta a los tiempos de recuperación tras fallos del RP, ya que el reenvío ininterrumpido (NSF) está configurado para los protocolos de routing. Puede tomar unos minutos adicionales para que el RP dañado o la tarjeta de línea vuelvan a estar disponibles después de que se produzca una caída mientras escribe el núcleo.

Aquí se muestra un ejemplo de la adición de esta configuración tanto a la configuración de modo estándar como de modo admin. Tenga en cuenta que la configuración del modo admin requiere que se utilicen DRP.

Este resultado muestra un ejemplo de configuración de volcado del núcleo:

```
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

Los servicios de transporte de paquetes locales (LPTS) gestionan los paquetes con destino local. El LPTS se compone de varios componentes diferentes.

1. El principal se denomina proceso del árbitro de puerto. Escucha las solicitudes de socket de diferentes procesos de protocolo, por ejemplo, BGP, IS-IS y realiza un seguimiento de toda la información de enlace para esos procesos. Por ejemplo, si un proceso BGP escucha en el socket número 179, el PA obtiene esa información de los procesos BGP y luego asigna un enlace a ese proceso en un IFIB.
2. La IFIB es otro componente del proceso LPTS. Ayuda a mantener un directorio donde un proceso escucha un enlace de puerto específico. La IFIB es generada por el proceso del Árbitro de Puerto y se mantiene con el árbitro de puerto. A continuación, genera varios subconjuntos de esta información.El primer subconjunto es la parte a de la IFIB. Esta porción se puede asociar al protocolo IPv4, etc. Luego se envían las porciones a los administradores de flujo apropiados, que luego utilizan la porción IFIB para reenviar el paquete al proceso adecuado.El segundo subconjunto es un pre-IFIB, permite que la LC reenvíe el paquete al proceso adecuado si sólo existe un proceso o a un administrador de flujo adecuado.
3. Los administradores de flujo ayudan a distribuir aún más los paquetes si la búsqueda no es trivial, por ejemplo, múltiples procesos para BGP. Cada administrador de flujo tiene una porción o varias porciones de la IFIB y reenvía los paquetes correctamente a los procesos adecuados asociados con la porción de la IFIB.
4. Si no se define una entrada para el puerto de destino, se puede descartar o reenviar al administrador de flujo. Un paquete se reenvía sin puerto asociado si hay una política asociada para el puerto. A continuación, el administrador de flujo ayuda a generar una nueva entrada de sesión.

¿Cómo se reenvía un paquete interno?

Hay dos tipos de flujos: flujos de capa 2 (HDLC, PPP) y flujos de ICMP/PING de capa 4 y flujos de routing.

1. HDLC/PPP de Capa 2: estos paquetes son identificados por el identificador de protocolo y se envían directamente a las colas de CPU en el pulverizador. Los paquetes de protocolo de

capa 2 obtienen una alta prioridad y luego son recogidos por la CPU (a través del Squid) y procesados. Por lo tanto, las señales de mantenimiento para la Capa 2 se responden directamente a través de la LC a través de la CPU. Esto evita la necesidad de ir al RP para obtener respuestas y funciona con el tema de la administración de interfaz distribuida.

2. Los paquetes ICMP (Capa 4) se reciben en la LC y se envían a través de la búsqueda a través de IFBI a las colas de CPU en el pulverizador. Estos paquetes se envían luego a la CPU (a través del Squid) y se procesan. La respuesta se envía luego a través de las colas de salida de Sprayer para reenviarse a través del entramado. En este caso, otra aplicación también necesita la información (replicada a través del fabric). Una vez a través del entramado, el paquete se dirige a la LC de salida adecuada y a través de la cola de esponja y control adecuada.
3. Los flujos de routing se buscan en la IFIB y luego se envían a las colas de modelado de salida (8000 colas) una de las cuales está reservada para los paquetes de control. Se trata de una cola no modelada a la que se presta servicio cada vez que está llena. - alta prioridad. Luego, el paquete se envía a través del entramado en las colas de alta prioridad a un conjunto de colas de CPU en el Esponge (similar a las colas Squid en el Sprayer) y luego se procesa por el proceso adecuado, el administrador de flujo o el proceso real. Se envía una respuesta a través de la esponja de la tarjeta de línea de salida y luego a través de la tarjeta de línea. La esponja LC de salida tiene una cola especial reservada para manejar los paquetes de control. Las colas en el Esponge se dividen en paquetes de alta prioridad, control y baja prioridad, por puerto de egreso.
4. El PSE tiene un conjunto de reguladores que se configuran para limitar la velocidad de los paquetes de capa 4, capa 2 y routing. Estos parámetros están predefinidos y se cambian para que el usuario pueda configurarlos posteriormente.

Uno de los problemas más comunes con LPTS son los paquetes que se descartan cuando intenta hacer ping al router. Los reguladores LPTS suelen limitar la velocidad de estos paquetes. Este es el caso para confirmar:

```
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

Los paquetes IP son inherentemente inseguros. IPSec es un método utilizado para proteger los paquetes IP. CRS-1 IPsec se implementa en la trayectoria de reenvío de software, por lo tanto la sesión IPsec se termina en el RP/DRP. Se soporta un número total de 500 sesiones IPsec por

CRS-1. El número depende de la velocidad de la CPU y de los recursos asignados. No hay limitación de software para esto, solamente el tráfico de origen local y el tráfico terminado localmente en RP son elegibles para el manejo de IPSec. El modo de transporte IPSec o el modo de túnel se pueden utilizar para el tipo de tráfico, aunque el primero se prefiere debido a una menor sobrecarga en el procesamiento de IPSec.

R3.3.0 soporta el encripción de BGP y OSPFv3 sobre IPsec.

Consulte Guía de Configuración de Seguridad del Sistema XR de Cisco IOS para obtener más información sobre cómo implementar IPSec.

Nota: IPSec requiere la conversión de criptografía, por ejemplo, hfr-k9sec-p.pie-3.3.1.

Fuera de banda

Acceso de consola y AUX

Los RP/SCs CRS-1 tienen una consola y un puerto AUX disponibles para fines de administración fuera de banda, así como un puerto Ethernet de administración para fuera de banda a través de IP.

La consola y el puerto AUX de cada RP/SCGE, dos por chasis, se pueden conectar a un servidor de consola. Esto significa que el sistema de chasis único requiere cuatro puertos de consola, y los sistemas de varios chasis requieren 12 puertos más dos para los motores supervisores en el Catalyst 6504-E.

La conexión del puerto AUX es importante ya que proporciona acceso al núcleo IOS-XR y puede permitir la recuperación del sistema cuando esto no es posible a través del puerto de la consola. El acceso a través del puerto AUX sólo está disponible para los usuarios definidos localmente en el sistema, y sólo cuando el usuario tiene acceso de nivel de soporte de cisco o sistema raíz. Además, el usuario debe tener una contraseña **secreta** definida.

Acceso a terminal virtual

Se puede utilizar Telnet & Secure Shell (SSH) para alcanzar CRS-1 a través de los puertos vty. De forma predeterminada, ambos están desactivados y el usuario debe habilitarlos explícitamente.

Nota: IPSec requiere la conversión de criptografía, por ejemplo, hfr-k9sec-p.pie-3.3.1.

Primero genere las claves RSA y DSA como se muestra en este ejemplo para habilitar SSH:

```
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
```

Información Relacionada

- Soporte de Routers
- Soporte Técnico y Documentación - Cisco Systems