Melhores prática operacionais do CRS-1 e IO XR

Índice

Introdução

Este documento ajuda a compreender o seguinte:

- Processo e linhas
- Tela do CRS-1
- Controle o plano
- Rommon e Monlib
- Módulo de interface da camada física (PLIM) e cartão modular do serviço (MSC)
- Gerenciamento de configuração
- Segurança
- Fora da faixa
- Protocolo simples de gerenciamento de rede (SNMP)

Pré-requisitos

Requisitos

Cisco recomenda que você tem o conhecimento do ® XR do Cisco IOS.

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Cisco IOS XR Software
- CRS-1

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

Convenções

Consulte as Convenções de Dicas Técnicas da Cisco para obter mais informações sobre convenções de documentos.

Vista geral do Cisco IOS XR

O Cisco IOS XR é projetado escalar. O núcleo é uma arquitetura de Microkernel assim que proporciona somente serviços essenciais tais como o gerenciamento do processo, a programação, os sinais, e os temporizadores. Todos os outros serviços tais como sistemas de arquivos, direcionadores, pilhas de protocolos e aplicativos são considerados como gerenciadores de recurso e corrida no espaço do usuário protegido da memória. Estes outros serviços podem ser adicionados ou removido no tempo de execução, que depende do projeto de programa. A pegada de Microkernel é o kb somente 12. O Microkernel e o sistema operacional subjacente são dos sistemas de software QNX, e são chamados Neutrino. QNX especializa-se no projeto de sistema operacional em tempo real. O Microkernel é preventivo, e o planificador é prioridade baseada. Isto assegura-se de que o interruptor de contexto entre processos seja muito rápido, e as linhas as mais prioritárias têm sempre o acesso ao CPU se necessário. Estes são alguns dos benefícios de que o Cisco IOS XR toma a vantagem. Mas, o benefício o mais grande é o projeto herdar de comunicações inter do processo dentro do núcleo dos sistemas operacionais.

O Neutrino é uma mensagem que passa o sistema operacional, e as mensagens são os meios básicos das comunicações entre processos entre todas as linhas. Quando um servidor particular quer proporcionar um serviço, cria um canal para mensagens de troca. Os clientes anexam aos server o canal diretamente traçando ao descritor de arquivo relevante a fim utilizar o serviço. Todas as comunicações entre o cliente e servidor são pelo mesmo mecanismo. Este é um benefício enorme para um computador super, que o CRS-1 seja. Considere estes quando uma operação de leitura local é executada em um núcleo padrão de UNIX:

- Interrupção de software no núcleo.
- Expedições do núcleo no sistema de arquivos.
- Os dados são recebidos.

Considere estes no caso remoto:

- Interrupção de software no núcleo.
- O núcleo despacha o NFS.
- O NFS chama o componente de rede de comunicação.
- O telecontrole despacha o componente de rede de comunicação.
- O NFS é chamado.
- O núcleo despacha o sistema de arquivos.

A semântica para o local lido e o telecontrole lido não são a mesma. Os argumentos e os parâmetros para o travamento de arquivo e o ajuste de permissões são diferentes.

Considere o QNX caso local:

- Interrupção de software no núcleo.
- O núcleo executa a mensagem que passa no sistema de arquivos.

Considere o caso não local:

- Interrupção de software no núcleo.
- O núcleo entra em QNET, que é o mecanismo de transporte IPC.
- QNET entra no núcleo.
- O núcleo despacha o sistema de arquivos.

Toda a semântica que se refere ao argumento que passa e os parâmetros de sistema de arquivos são idênticos. Tudo foi decuplado na relação IPC que permite que o cliente e servidor seja segregado completamente. Isto significa que todo o processo pode ser executado em qualquer lugar em qualquer momento a tempo. Se um processador da rota particular é pedidos de conservação demasiado ocupados, você pode facilmente migrar aqueles serviços a um CPU diferente que seja executado em um DRP. Um computador super que dirigisse serviços diferentes em CPU diferentes espalhou através dos nós múltiplos que podem facilmente se comunicar com todo o outro nó. A infraestrutura é no lugar a fim fornecer a oportunidade de escalar. Cisco utilizou esta vantagem e escreveu o software adicional que engancha nas operações principal da mensagem que passa o núcleo que permite que o roteador CR escale aos milhares de Nós, onde um nó, neste caso um CPU, executa um exemplo do OS, se é um processo da rota (RP), um processador de rotas distribuído (DRP), um cartão modular dos serviços (MSC), ou um switch processor (SP).

Processo e linhas

Dentro dos limites do Cisco IOS XR, um processo é uma área de memória protegida que contenha umas ou várias linhas. Da perspectiva dos programadores, as linhas fazem o trabalho, e

cada um termina um trajeto lógico da execução a fim executar uma tarefa específica. A memória que as linhas exigem durante o fluxo da execução pertence ao processo se operam dentro, protegido de todas as outras linhas dos processos. Uma linha é uma unidade de execução, com um contexto da execução que inclua uma pilha e se registre. Um processo é um grupo de linhas que compartilham de um espaço de endereço virtual, embora um processo possa conter uma única linha mas contém mais frequentemente mais. Se uma outra linha em um processo diferente tenta escrever à memória em seu processo, o processo de ofensa está matado. Se há mais de uma linha que se opera dentro de seu processo, a seguir essa linha tem o acesso à mesma memória dentro de seu processo, e em consequência é capaz overwrite os dados de uma outra linha. Termine as etapas em um procedimento a fim manter a sincronização aos recursos a fim impedir esta linha dentro do mesmo processo.

Uma linha usa um objeto chamado uma exclusão mútua (MUTEX) a fim assegurar a exclusão mútua aos serviços. A linha que tem os MUTEX é a linha que pode escrever a uma área particular da memória como um exemplo. Outras linhas que não têm os MUTEX não podem. Há igualmente outros mecanismos a fim assegurar a sincronização aos recursos, e estes são semáforos, variáveis condicionais ou Condvars, barreiras, e Sleepons. Estes não são discutidos aqui, mas proporcionam serviços de sincronização como parte de seus deveres. Se você iguala os princípios discutidos aqui ao Cisco IOS, a seguir o Cisco IOS é um único processo que opera muitas linhas, com todas as linhas que têm o acesso ao mesmo espaço de memória. Mas, o Cisco IOS chama estes processos das linhas.

Estados do processo e da linha

Dentro do Cisco IOS XR há os server que fornecem os serviços e os clientes que usam os serviços. Um processo particular pode ter um número de linhas que proporcionam o mesmo serviço. Um outro processo pode ter um número de clientes que puderam exigir um serviço particular em qualquer momento a tempo. O acesso aos server não está sempre disponível, e se um acesso do pedido do cliente a um serviço que se sente lá e se espere o server para estar livre. Neste caso o cliente seriam obstruído. Isto é chamado um modelo cliente/servidor de obstrução. O cliente pôde ser obstruído porque espera um recurso tal como um MUTEX, ou devido ao fato de que o server não respondeu ainda.

Emita um **comando ospf do processo da mostra** a fim verificar o estado das linhas no processo OSPF:

```
RP/0/RP1/CPU0:CWDCRS#show process ospf Job Id: 250 PID: 110795 Executable path: /disk0/hfr-rout-
3.2.3/bin/ospf Instance #: 1 Version ID: 00.00.0000 Respawn: ON Respawn count: 1 Max. spawns per
minute: 12 Last started: Tue Jul 18 13:10:06 2006 Process state: Run Package state: Normal
Started on config: cfg/gl/ipv4-ospf/proc/101/ord_a/routerid core: TEXT SHAREDMEM MAINMEM Max.
core: 0 Placement: ON startup_path: /pkg/startup/ospf.startup Ready: 1.591s Available: 5.595s
Process cpu time: 89.051 user, 0.254 kernel, 89.305 total JID TID Stack pri state HR:MM:SS:MSEC
NAME 250 1 40K 10 Receive 0:00:11:0509 ospf 250 2 40K 10 Receive 0:01:08:0937 ospf 250 3 40K 10
Receive 0:00:03:0380 ospf 250 4 40K 10 Condvar 0:00:00:0003 ospf 250 5 40K 10 Receive
0:00:05:0222 ospf
```

Note que o processo OSPF está dado um trabalho ID (JID), que seja 250. Isto nunca muda em um roteador running e geralmente em uma versão específica do Cisco IOS XR. Dentro do processo OSPF há cinco rosqueia cada um com sua própria linha ID (TID). Listado está o espaço de pilha para cada linha, a prioridade de cada linha e seu estado.

Passagem síncrono da mensagem

Menciona-se mais cedo que QNX é uma mensagem que passa o sistema operacional. É

realmente uma mensagem síncrono que passa o sistema operacional. Muitas edições do sistema operacional são refletidas na Mensagem síncrono. Não se diz que a passagem síncrono da mensagem causa todos os problemas, mas o sintoma do problema é refletido um pouco na passagem síncrono da mensagem. Porque é síncrono, o ciclo de vida ou a informação de estado são facilmente acessível ao operador do CRS-1, que ajuda no processo de Troubleshooting. A mensagem que passa o ciclo de vida é similar a esta:

- Um server cria um canal da mensagem.
- Um cliente conecta ao canal de um server (análogo ao posix aberto).
- Um cliente envia uma mensagem a um server (MsgSend) e esperas para uma resposta e blocos.
- O server recebe (MsgReceive) uma mensagem de um cliente, processa a mensagem, e responde ao cliente.
- O cliente desbloqueia e processa a resposta do server.

Este client-server model de obstrução é a passagem síncrono da mensagem. Isto significa que o cliente envia uma mensagem e blocos. O server recebe a mensagem, processa-a, responde de volta ao cliente e então o cliente desbloqueia. Estes são os detalhes específicos:

- O server espera RECEBE dentro o estado.
- O cliente envia uma mensagem ao server e torna-se OBSTRUÍDO.
- O server recebe a mensagem e desbloqueia-a, se esperando dentro receba o estado.
- O cliente transporta-se ao estado da RESPOSTA.
- O server move-se para o estado de execução.
- Processos de servidor a mensagem.
- O server responde ao cliente.
- O cliente desbloqueia.

Emita o **comando show process** a fim ver em que estados o cliente e servidor é.

```
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
```

Processo bloqueado e estados de processo

Emita o comando **obstruído processo da mostra** a fim ver que processo está no estado obstruído.

```
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#
```

A passagem sincronizada da mensagem permite-o de seguir facilmente o ciclo de vida do Inter-Process Communication entre as linhas diferentes. A qualquer hora, uma linha pode estar em um estado específico. Um estado obstruído pode ser um sintoma de um problema. Isto não significa que se uma linha está no estado obstruído então há um problema, assim que não emite o comando **obstruído processo da mostra** e abre um caso com Suporte técnico de Cisco. As linhas obstruídas são igualmente muito normais.

Note a saída precedente. Se você olha a primeira linha na lista, note-a é o KSH, e sua resposta é obstruída no devc-conaux. O cliente, o KSH neste caso, enviou uma mensagem ao processo do devc-conaux, o server, que é devc-conaux, resposta das posses KSH obstruída até que responda. O KSH é o shell unix que alguém usa no console ou no porto auxiliar. O KSH espera a entrada do console, e se não há nenhuns porque o operador não está datilografando, a seguir permanece obstruído até tal hora que processa alguma entrada. Após o processamento, o KSH retorna à resposta obstruída no devc-conaux.

Isto é normal e não ilustra um problema. O ponto é que as linhas obstruídas são normais, e depende do que versão XR, o tipo de sistema você tem, do que você configurou e quem faz o que aquele altera a saída do comando **obstruído processo da mostra**. O uso do comando **obstruído processo da mostra** é uma boa maneira de começar pesquisar defeitos o tipo problemas do OS. Se há um problema, por exemplo o CPU é alto, a seguir usa o comando precedente a fim ver se qualquer coisa olha fora do normal.

Compreenda o que é normal para seu roteador de funcionamento. Isto fornece uma linha de base para que você use-se como uma comparação quando você pesquisa defeitos ciclos de vida do processo.

A qualquer hora, uma linha pode estar em um estado específico. Esta tabela fornece uma lista dos estados:

Se o estado é:	A linha é:
INOPERANTE	Inoperante. O núcleo está esperando para liberar os recursos das linhas.
EXECUTANDO	Ativamente sendo executado em um CPU
PRONTO	Não ser executado em um CPU mas é pronto para ser executado
PARADO	Suspendido (sinal SIGSTOP)

ENVIE	Esperando um server para receber uma mensagem
RECEPÇÃO	Esperando um cliente para enviar uma mensagem
RESPOSTA	Esperando um server para responder a uma mensagem
PILHA	Esperando mais pilha para ser atribua
WAITPAGE	Esperando o gerente do processo para resolver uma falha de página
SIGSUSPEND	Esperando um sinal
SIGWAITINFO	Esperando um sinal
NANOSLEEP	Sono por um período de tempo
MUTEX	Espera para adquirir um MUTEX
CONDVAR	Esperando uma variável condicional a ser sinalizada
UNIÃO	Esperando a conclusão de uma outra linha
INTR	Esperando uma interrupção
SEM	Espera para adquirir um semáforo

Processos importantes e suas funções

O Cisco IOS XR tem muitos processos. Estes são alguns importantes com suas funções explicadas aqui.

Monitor de sistema Watchdog (WDSysmon)

Este é um serviço proporcionado para a detecção de processo pendura e condições de memória baixa. A memória baixa pode ocorrer em consequência de um escape de memória ou de alguma outra circunstância estranha. Um cair pode ser o resultado de um número de condições tais como paralizações completas do processo, loop infinitos, aprisionamentos do núcleo ou erros da programação. Em todo o ambiente multi-rosqueado o sistema pode obter em um estado conhecido como uma condição da paralização completa, ou apenas simplesmente a paralização completa. Uma paralização completa pode ocorrer quando umas ou várias linhas são incapazes de continuar devido à contenção de recursos. Por exemplo, rosqueie A pode enviar uma mensagem para rosquear B quando simultaneamente a linha B enviar uma mensagem para rosquear o A. Ambas as linhas esperam em se e podem estar dentro enviam o estado obstruído, e ambas as linhas esperam para sempre. Este é um caso simples que envolva duas linhas, mas se um server é responsável para um recurso que esteja usado por muitas linhas é obstruído em uma outra linha, a seguir em muitas linhas que pedem acesso a esse recurso pode ser enviam a espera obstruída no server.

As paralizações completas podem ocorrer entre algumas linhas, mas podem impactar outras linhas em consequência. As paralizações completas são evitadas pelo bom projeto de programa, mas independentemente de como magnificamente um programa é projetado e escrito. Às vezes uma sequência de evento particular que são dependente dos dados com sincronismos específicos pode causar uma paralização completa. As paralizações completas não são sempre determinísticas e são geralmente muito difíceis de reproduzir. WDSysmon tem muitas linhas com

uma que é executado na prioridade mais alta que o Neutrino apoia, 63. Ser executado na prioridade 63 assegura-a que a linha obtém o processador central - tempo em um ambiente preventivo baseado prioridade da programação. WDSysmon trabalha com a capacidade e os relógios do cão de guarda do hardware sobre os processos de software que procuram condições do cair. Quando tais circunstâncias são detectadas, WDSysmon recolhe a informação adicional em torno da circunstância, pode coredump o processo ou o núcleo, para escrever para fora aos Syslog, executa scripts, e mata os processos bloqueados. Dependente em cima de como drástico o problema é, pode iniciar um interruptor do processador de rotas sobre a fim manter a operação de 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
```

O processo WDSysmon tem nove linhas. Quatro executados na prioridade 10, os outros quatro estão em 11, em 19, em 20 e em 63. Quando um processo é projetado, o programador considera com cuidado a prioridade que cada linha dentro do processo deve ser dada. Como discutido previamente, o planificador é a prioridade baseada, que significa que uma linha mais prioritária cancela sempre uma da baixa prioridade. A prioridade 63 é a prioridade mais alta que uma linha pode executar em, que seja a linha 7 neste caso. A linha 7 é a linha do observador, a linha essa CPU hog das trilhas. Deve ser executado em uma prioridade mais alta do que as outras linhas que as olha de outra maneira não puderam obter a possibilidade ser executado de todo, que a impede das etapas que foi projetada executar.

Netio

No Cisco IOS, há o conceito da comutação rápida do interruptor e do processo. O interruptor rápido usa o código de CEF e ocorre no tempo de interrupção. Processe o ip_input dos usos do interruptor, que é o código da Comutação IP, e seja um processo agendado. Em Plataformas de uma extremidade mais alta o CEF switching é feito no hardware, e o ip_input é programado no CPU. O equivalente do ip_input no Cisco IOS XR é 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
```

Processo dos serviços do grupo (GSP)

Há uma necessidade para uma comunicação em todo o super-computador com os diverso mil Nós esse que cada um executa seu próprio exemplo do núcleo. No Internet, um aos muitos uma comunicação é feito eficientemente através dos protocolos do multicasting. O GSP é o protocolo

interno do multicasting que é usado para o IPC dentro do CRS-1. O GSP fornece um aos muitos uma comunicação segura do grupo que seja sem conexão com semântica assíncrona. Isto permite que o GSP escale ao mil dos Nós.

```
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 do índice do volume BCDL

BCDL é confiantemente dados de transmissão múltipla usados aos vários Nós tais como RP e MSC. Usa o GSP como o transporte subjacente. Garantias BCDL **na** entrega de **ordem das** mensagens. Dentro de BCDL há um agente, um produtor e um consumidor. O agente é o processo que se comunica com o produtor a fim recuperar e proteger os dados antes de seus Multicast aos consumidores. O produtor é o processo que produz os dados que todos quer, e o consumidor é o processo interessado receber os dados fornecidos pelo produtor. BCDL é usado durante elevações do Software Cisco IOS XR.

Mensagem de pouco peso (LWM)

O LWM é um formulário Cisco-criado da Mensagem que foi projetada criar uma camada de abstração entre os aplicativos que o processo inter comunica um com o otro e Neutrino, com o objetivo como a independência do sistema operacional e da camada de transporte. Se Cisco deseja mudar o vendedor do OS de QNX a alguma outra pessoa, uma camada de abstração entre as funções rudimentarmente das ajudas do sistema operacional subjacente remove a dependência no sistema operacional e nos auxílios em mover a um outro sistema operacional. O LWM fornece a entrega de mensagem garantida síncrono, que gostam da mensagem nativa do Neutrino que passa, faz com que o remetente obstrua até que o receptor responda.

O LWM igualmente fornece a entrega do mensagem assíncrona através de 40 pulsos do bit. Os mensagens assíncrona são enviados assincronamente, que significa que a mensagem está enfileirada e o remetente não obstrui, mas não está recebido pelo server assincronamente, mas quando o server vota para a mensagem disponível seguinte. O LWM é estruturado como o cliente/server. O server cria um canal que lhe dê uma **orelha** para escutar dentro mensagens e o sente em um tempo o laço faça uma mensagem receba a escuta no canal, que apenas criou. Quando uma mensagem chega desbloqueia e obtém um identificador de cliente, que seja eficazmente a mesma coisa que a recepção ID da mensagem recebeu. O server executa então algum que processa e faz mais tarde uma resposta da mensagem ao identificador de cliente.

No lado do cliente faz uma mensagem conecta. Obtém passado um identificador a quem conecta e faz então uma mensagem envia e é obstruído. Quando o server termina processar, responde e o cliente torna-se desbloqueado. Este é virtualmente o mesmo que a mensagem nativa dos Neutrinos que passa, assim que a camada de abstração é muito finamente.

O LWM é projetado com um número mínimo de chamadas de sistema e de Switches do contexto para o alto desempenho, e é o método preferido do IPC no ambiente do Cisco IOS XR.

Envmon

No máximo o fundamento em nível, o sistema de monitoramento ambiental for responsável para advertir quando parâmetros físicos, por exemplo temperatura, tensão, velocidade do fã e assim por diante, queda fora das escalas operacionais, e de fechar o hardware que aproxima os níveis críticos onde o hardware pôde ser danificado. Periodicamente monitora cada sensor disponível do hardware, compara o valor medido contra pontos iniciais cartão-específicos, e levanta alarmes como necessário a fim realizar esta tarefa. Um processo persistente, começado na inicialização do sistema, que vota periodicamente todos os sensores do hardware, por exemplo tensão, temperatura, e velocidade do fã, no chassi e fornece estes dados aos clientes externos do Gerenciamento. Além, o processo periódico compara leituras do sensor com os limiares de alarme e publica alertas ambientais ao base de dados de sistema para a ação subsequente pelo gerente da falha. Se as leituras do sensor são perigosamente fora da escala, o processo do monitoramento ambiental pôde causar o cartão ser parada programada.

Introdução da tela do CRS-1

- Tela de vários estágios — topologia de Benes de 3 fases
- Roteamento dinâmico dentro da tela para minimizar a congestão
- Pilha baseada: 136 células de byte, payload de dados de 120 bytes
- Controle de fluxo para melhorar o isolamento de tráfego e para minimizar exigências da proteção na tela
- Fase para encenar o fornecimento da aceleração
- Dois moldes do tráfego apoiados (unicast & Multicast)
- Duas prioridades do tráfego apoiadas pelo molde (alto e baixo)
- Apoio para grupos de transmissão múltipla da tela de 1M (FGIDs)
- Tolerância de defeito eficaz na redução de custos: Redundância N+1 ou N+k usando planos da tela ao contrário de 1+1 a custo extremamente aumentado

Quando você é executado no modo do chassi único, o asics S1, S2 e S3 está ficado situado nas mesmas placas de fábrica. Este cartão é referido igualmente geralmente como o **cartão S123**. Em uma configuração dos Multi-chassis, o S2 é separado e está no chassi de placa de fábrica (FCC). Esta configuração exige duas placas de fábrica formar um plano, um cartão S2, e um cartão S13. Cada MSC conecta a oito planos da tela a fim fornecer a Redundância de modo que se você afrouxa uns ou vários planos, sua tela ainda passe o tráfego embora o tráfego agregado, que pode atravessar a tela, seja mais baixo. Os CR podem ainda operar-se na taxa de linha para a maioria de tamanhos do pacote com somente sete planos. A pressão contrária é enviada sobre a tela sobre um impar e mesmo plano. Não se recomenda executar com menos do que um sistema dois planos, em um impar e mesmo plano. Qualquer coisa menos de dois planos não é uma configuração suportada.

O plano da tela

O diagrama precedente representa um plano. Você tem que multiplicar esse diagrama por oito. Isso significa que o pulverizador (ingressq) asic de um LC conecta a 8 S1 (1 S1 pelo plano). O S1 em cada plano da tela conecta a 8 pulverizadores:

- os 8 LC superiores do chassi
- os 8 LC inferiores

Há 16 S1 por 16 chassis do entalhe LC: 8 para a parte superior LC (1 pelo plano) + 8 para a parte

inferior LC.

Em uns único 16 chassis do entalhe, uma placa de fábrica S123 tem 2 S1, 2 S2 e 4 S3. Aquela é parte da computação da aceleração da tela. Há duas vezes mais tráfego, que pode retirar a tela como o tráfego pode entrar. Há igualmente atualmente duas esponjas (fabricq) pelo LC comparado a 1 pulverizador. Isto permite protegendo na saída LC quando mais de um ingresso LC sobrecarrega uma saída LC. A saída LC pode absorver essa largura de banda extra da tela.

Monitoração da tela

Disponibilidade e Conectividade planas:

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

Verifique se planos estão recebendo/pilhas transmissores e alguns erros estão incrementando:

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

Os acrônimos no comando precedente:

- CE — Erro corrigível
- ECU — Erro incorrigível
- PE — Erro de paridade

Não se preocupe se observam alguns erros, porque este pode acontecer na inicialização. Os campos não devem incrementar no tempo de execução. Se são, pode ser uma indicação de um problema na tela. Emita este comando a fim obter uma divisão dos erros pelo plano da tela:

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

Controle a vista geral plana

A Conectividade do plano do controle entre o chassi da placa de linha e o chassi da tela é atualmente através das portas de Ethernet Gigabit nos RP (LCC) e em SCGE (FCC). A interconexão entre as portas é fornecida através de um par de Catalyst 6500 Switch, que podem ser conectados através de dois ou mais portas de Ethernet Gigabit.

Configuração do Catalyst 6500

Esta é configuração recomendada para os Catalyst Switches usados para o plano do controle dos multi-chassis:

- Um único VLAN é usado em todas as portas.
- Todas as portas executadas no modo de acesso (nenhum entroncamento).
- a Medir-árvore 802.1w/s é usada para a prevenção do laço.
- Dois ou mais links são usados a fim cruz-conectar os dois Switches e o STP é usado para laço-impede. Canalizar não é recomendada.
- Portas que conectam modo do PRE-padrão ao uso do CRS-1 RP e SCGE desde que IOS-XR não apoia o 802.1s baseado padrões.
- O UDLD deve ser permitido nas portas que conectam entre o Switches e entre o Switches e o RP/SCGE.
- O UDLD é permitido à revelia no CRS-1.

Refira trazer acima o Software Cisco IOS XR em um sistema de Multishelf para obter mais informações sobre de como configurar um Catalyst 6500 em um sistema de Multishelf.

Gerenciamento do plano do controle dos Multi-chassis

O chassi do catalizador 6504-E, que fornece a Conectividade do plano do controle para o sistema dos multi-chassis, é configurado para estes serviços da gerência:

- Gerenciamento In-Band através do gigabit de porta 1/2, que conecta a um switch LAN em cada PNF. O acesso é permitido somente para uma escala pequena das sub-redes e dos protocolos.
- O NTP é usado a fim ajustar o tempo de sistema.
- A informações de syslog é executada aos anfitriões padrão.
- O polling snmp e as armadilhas podem ser permitidos para funções crítica.

Nota: Nenhuma mudança deve ser feita ao catalizador na operação. Os testes prévios devem ser feitos em toda a alteração planejada e é altamente recomendado que este está feito durante uma janela de manutenção.

Esta é uma amostra de configuração de gerenciamento:

```
#In-band management connectivity interface GigabitEthernet2/1 description *CRS Multi-chassis
Management Ethernet - DO NOT TOUCH* ip address [ip address] [netmask] ip access-group
control_only in ! ! ip access-list extended control_only permit udp [ip address] [netmask] any
eq snmp permit udp [ip address] [netmask] eq ntp any permit tcp [ip address] [netmask] any eq
telnet #NTP ntp update-calendar ntp server [ip address] #Syslog logging source-interface
Loopback0 logging [ip address] logging buffered 4096000 debugging no logging console #RADIUS aaa
new-model aaa authentication login default radius enable enable password {password} radius-
server host [ip address] auth-port 1645 acct-port 1646 radius-server key {key} #Telnet and
console access ! access-list 3 permit [ip address] ! line con 0 exec-timeout 30 0 password
{password} line vty 0 4 access-class 3 in exec-timeout 0 0 password {password}
```

ROMMON e Monlib

O monlib de Cisco é um programa executável que seja armazenado no dispositivo e carregado em RAM para a execução pelo ROMMON. O ROMMON usa o monlib a fim alcançar arquivos no dispositivo. As versões rommon podem ser promovidas e devem ser feitas assim sob a recomendação do Suporte técnico de Cisco. A versão rommon a mais atrasada é 1.40.

Instruções de upgrade

Conclua estes passos:

1. Transfira os binários ROMMON do CRS-1 ROMMON de Cisco (clientes registrados somente).
2. Desembale o arquivo TAR e copie os arquivos bin 6 no diretório raiz CR do disco 0.`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. Use o **diag da mostra | ROM inc|NÓ|**Comando **PLIM** a fim ver a versão rommon atual.`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. Entra no modo ADMIN e usa o **rommon da elevação todo o** comando do **disco 0** a fim promover o 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. Retire o modo ADMIN e incorpore o **log da mostra | o inc "APROVADO, o ROMMON A"** e certificam-se de todos os Nós promovidos com sucesso. Se alguns dos Nós falham, passe para trás a etapa 4 e reprogram.`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. Entra no modo ADMIN e usa o **rommon b da elevação todo o** comando do **disco 0** a fim promover o 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. Retire o modo ADMIN e incorpore o **log da mostra | o inc "APROVADO, o ROMMON B"** e certificam-se de todos os Nós promovidos com sucesso. Se alguns dos Nós falham, passe para trás a etapa 4 e reprogram.`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. **O comando upgrade** apenas queima uma seção reservado especial do bootflash com o ROMMON novo. Mas o ROMMON novo permanece inativo até que o cartão esteja recarregado. Assim quando você o reload o cartão, o ROMMON novo for ativo. Restaure cada nó um de cada vez ou apenas restaure o roteador inteiro a fim fazer isto.
```
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. Use o **diag da mostra | ROM inc|NÓ|**Comando **PLIM** a fim verificar a versão do ROMmon atual.
```
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: Em CRS-8 e em chassi da tela, o ROMMON igualmente ajusta as velocidades do fã à velocidade padrão de 4000 RPM.

Vista geral PLIM e MSC

Isto representa o fluxo de pacote de informação no roteador do CRS-1, e estes termos são usados permutavelmente:

IngressQ ASIC é chamado igualmente o pulverizador ASIC.

FabricQ ASIC é chamado igualmente a esponja ASIC.

EgressQ ASIC é chamado igualmente o Sharq ASIC.

Os SPP são chamados igualmente o PSE (motor de switch de pacotes) ASIC.

RX PLIM > RX SPP > ingresso Q > tela > tela Q > Tx SPP > saída Q > Tx PLIM (pulverizador) (esponja) (Sharq)

Os pacotes são recebidos no módulo de interface da camada física (PLIM).

O PLIM contém as interfaces física para o MSC com que se acopla. O PLIM e o MSC são cartões separados conectados através do chassi de placa mãe. Em consequência os tipos de interface

para um MSC particular são definidos pelo tipo do PLIM que se acoplou com. O dependente em cima do tipo de PLIM, o cartão contém um vário número de ASIC que fornecem os meios físicos e a moldação para as relações. A finalidade do PLIM ASIC é fornecer a relação entre o MSC e as conexões física. Termina a fibra, faz a luz à conversão elétrica, termina os media que moldam sendo SDH/Sonet/Ethernet/HDLC/PPP, verifica o CRC, adiciona alguma informação de controle chamada o cabeçalho de buffer e para a frente os bit que permanece no MSC. O PLIM não faz fonte/dissipador o HDLC ou as manutenções de atividade de PPP. Estes são segurados pelo CPU no MSC.

O PLIM igualmente fornece estas funções:

- MAC que filtra para 1/10 de Gigabit Ethernet
- Ingresso/saída MAC que esclarece 1/10 de Gigabit Ethernet
- VLAN que filtra para 1/10 de Gigabit Ethernet
- VLAN que esclarece 1/10 de Gigabit Ethernet
- Proteção e notificação de congestionamento do ingresso

Sobreassinatura PLIM

10GE PLIM

Os 8 X10G PLIM oferecem a capacidade de terminar aproximadamente o 80 Gbps do tráfego enquanto a capacidade de encaminhamento do MSC é um máximo do 40 Gbps. Se todas as portas disponíveis no PLIM são povoadas, a seguir a sobreassinatura ocorre e a modelagem de QoS torna-se extremamente importante assegurar-se de que o tráfego superior não esteja deixado cair inadvertidamente. Para algum, a sobreassinatura não é uma opção e deve ser evitada. Somente quatro das oito portas devem ser usadas a fim fazer isto. Além, deve ser tomado para assegurar-se de que a largura de banda a melhor dentro do MSC e do PLIM esteja disponível a cada um das quatro portas.

Nota: As mudanças do mapeamento de porta da liberação 3.2.2 avante. Veja estes diagramas.

Mapeamento de porta até a liberação 3.2.1 Mapeamento de porta da liberação 3.2.2 avante
Como mencionado previamente, as portas física são prestadas serviços de manutenção por um dos dois FabricQ ASIC. A atribuição das portas ao ASIC estaticamente é definida e não pode ser alterada. Além, os 8 X10G PLIM têm dois PLA ASIC. O primeiro PLA presta serviços de manutenção às portas 0 3, os segundos serviços 4 ao 7. A capacidade de largura de banda de um único PLA nos 8 X10G PLIM é aproximadamente 24 Gbps. A capacidade de switching de um único FabricQ ASIC é aproximadamente 62 Mpps.

Se você povoa a porta 0 3 ou portas 4 a 7, a capacidade de largura de banda do PLA (24 Gbps) está compartilhada entre todos os quatro das portas que restringem o throughput geral. Se você povoa as portas 0,2,4 & o 6 (até 3.2.1) ou 0,1,4 & 5 (3.2.2 avante) como todas estas portas é prestado serviços de manutenção pelo um FabricQ ASIC, cuja a capacidade de switching é 62 Mpps, outra vez, que restringe a capacidade de taxa de transferência.

É o melhor utilizar de um modo as portas que obtém a eficiência mais elevada dos PLAs e do FabricQ ASIC a fim conseguir o desempenho ideal.

SIP-800/SPA

O SIP-800 PLIM oferece a capacidade para operar-se com os cartões de interface modular conhecidos como adaptadores de porta do serviço (termas). O SIP-800 fornece baías dos TERMAS 6 uma capacidade teórica da relação de 60 Gbps. A capacidade de encaminhamento do MSC é um máximo do 40 Gbps. Se todas as baías no SIP-800 deviam ser povoado, a seguir, dependente do tipo dos TERMAS, é possível que a sobreassinatura ocorre e modelagem de QoS se torna extremamente importante a fim se assegurar de que o tráfego superior não esteja deixado cair inadvertidamente.

Nota: A sobreassinatura não é apoiada com interfaces pos. Mas, a colocação dos TERMAS 10 Gb POS deve ser apropriada a fim assegurar-se de que a capacidade de taxa de transferência correta esteja fornecida. Os TERMAS dos Ethernet 10 Gb são apoiados somente na liberação 3.4 IOS-XR. Estes TERMAS oferecem capacidades da sobreassinatura.

Para algum, a sobreassinatura não é uma opção e deve ser evitada. Somente quatro dos sixbays devem ser usados a fim fazer isto. Além, o cuidado deve ser ordem recolhida para assegurar-se de que a largura de banda a melhor dentro do MSC e do PLIM esteja disponível a cada um das quatro portas.

Mapeamento da baía dos TERMAS

Como mencionado em previamente, as portas física são prestadas serviços de manutenção por um dos dois FabricQ ASIC. A atribuição das portas ao ASIC estaticamente é definida e não pode ser alterada. Além, o SIP-800 PLIM tem dois PLA ASIC. O primeiro PLA presta serviços de manutenção às portas 0,1 & 3, os segundos serviços 2, 4 & 5.

A capacidade de largura de banda de um único PLA no SIP-800 PLIM é aproximadamente 24 Gbps. A capacidade de switching de um único FabricQ ASIC é aproximadamente 62 Mpps.

Se você povoa a porta 0,1 & 3 ou portas 2, 4 & 5, a capacidade de largura de banda do PLA (24 Gbps) está compartilhada entre todos os três das portas que restringem o throughput geral. Desde que um único FabricQ cada serviços aqueles grupos de porta, a taxa do pacote máximo do grupo de porta é 62 Mpps. É o melhor utilizar de um modo as portas que obtém a eficiência mais elevada dos PLAs a fim conseguir a largura de banda a melhor.

Colocação sugerida:

	Bay# dos TERMAS	**Bay# dos TERMAS**	**Bay# dos TERMAS**	**Bay# dos TERMAS**
Opção 1	0	1	4	5
Opção 2	1	2	3	4

Se você quer povoar o cartão com mais os TERMAS de quatro, a recomendação é terminar uma das opções alistadas previamente, que espalham as relações entre os dois grupos de porta (0,1 & 3 & 2,4 & 5). Você deve então colocar os módulos seguintes dos TERMAS em uma das portas aberta em qualquer um os 0,1 & os 3 & os 2,4 & os grupos de porta 5.

DWDM XENPACKs

Da liberação 3.2.2 avante, o DWDM XENPACKs pode ser instalado e fornecido os módulos **ajustáveis do** sistema ótico. As exigências refrigerando de tais módulos XENPACK exigem que haja um entalhe vazio entre os módulos instalados. Além, se um único módulo DWDM XENPACK é instalado, um máximo de quatro portas pode ser usado, mesmo se o os módulos XENPACK não são dispositivos DWDM. Isto tem consequentemente um impacto direto no FabricQ ao PLA ao mapeamento de porta. A atenção precisa de ser pagada a esta exigência e é considerada nesta tabela.

Colocação sugerida:

	Port- do sistema ótico	**Port- do sistema ótico**	**Port- do sistema ótico**	**Port- do sistema ótico**
Opção 1 ou DWDM XENPACK	0	2	5	7
Opção 2	1	3	4	6

Para uns 3.2.2 ou mais atrasado ou 3.3 a instalação, evite a mudança do mapeamento de FabricQ. Um teste padrão mais simples da colocação pode consequentemente ser usado para o regular e os módulos DWDM XENPACK.

	Port- do sistema ótico	**Port- do sistema ótico**	**Port- do sistema ótico**	**Port- do sistema ótico**
Opção 1	0	2	4	6
Opção 2	1	3	5	7

Se você quer povoar o cartão com mais de quatro portas NON-DWDM XENPACK, a recomendação é terminar uma das opções alistadas, que espalha os módulos de interface ótica entre os dois grupos de porta (0-3 & 4-7). Você precisa de colocar então os módulos de interface ótica seguintes em uma das portas aberta nos 0-3 ou 4-7 grupos de porta. Se você usa o grupo de porta 0-3 para o módulo de interface ótica #5, os módulos de interface ótica #6 devem ser colocados no grupo de porta 4-7.

Refira os módulos DWDM XENPAK para mais detalhes.

Gerenciamento de configuração

A configuração em IOS-XR é feita com uma configuração de duas fases, a configuração é entrada pelo usuário na primeira fase. Esta é a fase onde somente a sintaxe de configuração é verificada pelo CLI. A configuração incorporada a esta fase é sabida somente ao processo do agente da configuração, por exemplo, CLI/XML. A configuração não é verificada desde que não se escreve ao server do sysdb. O aplicativo backend não é notificado e não pode alcançar ou ter nenhum conhecimento da configuração nesta fase.

No segundo estágio, a configuração é comprometida explicitamente pelo usuário. Nesta fase a configuração é escrita ao server do sysdb, os aplicativos backend verificam que as configurações

e as notificações estão geradas pelo sysdb. Você pode abortar uma sessão da configuração antes que você comprometa a configuração incorporada à primeira fase. Assim, não é seguro supor que toda a configuração incorporada à fase uma está comprometida sempre na fase dois.

Além, o funcionamento e/ou a configuração do corredor do roteador podem ser alterados por usuários múltiplos durante a fase uma e a fase dois. Assim, nenhum teste do roteador que executa a configuração e/ou o estado operacional na fase uma não pôde ser válido na fase dois onde a configuração é comprometida realmente.

Sistemas de arquivo de configuração

O sistema de arquivo de configuração (CF) é um grupo de arquivos e diretórios usados a fim armazenar a configuração do roteador. Os CF são armazenados sob o diretório disk0:/config/, que é o media do padrão usado no RP. Os arquivos e os diretórios nos CF são internos ao roteador e devem nunca ser alterados ou removido pelo usuário. Isto pode conduzir à perda ou à corrupção da configuração e afeta o serviço.

Os CF são checkpointed ao à espera-RP após o cada comprometem. Isto ajuda a conserva o arquivo de configuração do roteador após uma falha sobre.

Durante a inicialização do roteador, a última configuração ativa é aplicada da configuração compromete o base de dados armazenado nos CF. Não é necessário que o usuário salvar manualmente a configuração ativa depois que cada configuração compromete, desde que esta está feita automaticamente pelo roteador.

Não é aconselhável fazer alterações de configuração quando a configuração for aplicada durante a inicialização. Se o aplicativo da configuração não está completo, você vê esta mensagem quando você entra ao roteador:

Processo da configuração de sistema

A configuração de inicialização para este dispositivo está carregando presentemente. Isto pode tomar alguns minutos. Você é notificado em cima da conclusão. Por favor não tente reconfigurar o dispositivo até que este processo esteja completo. Em alguns casos raros, pôde ser desejável restaurar a configuração de roteador de um arquivo de configuração fornecido usuário ASCII em vez de restaurar a última configuração ativa dos CF.

Você pode forçar o aplicativo de um arquivo de configuração por:

```
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=<config-file-path> rommon>boot <image>
```

Quando você restaurar a configuração de roteador, uma ou vária o item de configuração pôde não toma o efeito. Toda a configuração falhada salvar nos CF e é mantida até o reload seguinte.

Você pode consultar a configuração falhada, endereçar os erros e reaplicar a configuração.

Estas são algumas pontas a fim endereçar a configuração falhada durante a partida do roteador.

Em IOX, a configuração pode ser classificada como a configuração falhada para três razões:

1. Erros de sintaxe — O parser gerencie os erros de sintaxe, que indicam geralmente que há uma incompatibilidade com comandos CLI. Você deve corrigir os erros de sintaxe e reaplicar a configuração.
2. Erros semânticos — Os erros semânticos estão gerados pelos componentes backend quando o gerenciador de configuração restaura a configuração durante a partida do roteador. É importante notar que o cfgmgr não é responsável para assegurar a configuração está aceitado como parte de configuração running. Cfgmgr é meramente um médio-**homem** e relata somente todas as falhas semânticas que os componentes backend gerarem. Incumbe cada proprietário componente backend para analisar a razão da falha e para determinar a razão para a falha. Os usuários podem executar o **commands> da descrição <CLI do** modo de configuração a fim encontrar facilmente o proprietário do verificador componente backend. Por exemplo, se o **BGP 217 do roteador** aparece como a configuração falhada, o comando da **descrição** mostra que o verificador componente é o 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. Aplique erros — A configuração com sucesso foi verificada e aceitada como parte de configuração running mas o componente backend não pode atualizar seu estado operacional por qualquer motivo. A configuração mostra em ambos a configuração sendo executado, desde que se verificou corretamente, e como a configuração falhada devido ao erro operacional backend. O comando da **descrição** pode outra vez ser executado no CLI que não se aplicou a fim encontrar o componente para aplicar o proprietário.Termine estas etapas a fim consultar e reaplicar a configuração falhada durante operadores startup:Para os operadores R3.2 pode usar este procedimento a fim reaplicar a configuração falhada:Os operadores podem usar o comando da **inicialização falha da configuração da mostra** a fim consultar a configuração falhada salvar durante a inicialização do roteador.Os operadores devem executar o **noerror da inicialização falha da configuração da mostra | arquive o comando myfailed.cfg** a fim salvar a configuração falhada startup a um arquivo.Os operadores devem ir ao **modo de configuração** e à **carga do** uso**/comandos commit** a fim reaplicar isto configuração falhada: `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 operadores das imagens R3.3 pode usar este procedimento actualizado:Os operadores devem usar o comando da **inicialização falha da configuração da mostra** e o comando da **inicialização falha da configuração de carga** a fim consultar e reaplicar toda a configuração falhada.`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`**

Descarregador do núcleo

Àrevelia IOS-XR escreve um dump principal ao disco duro se um impacto do processo, mas não se o núcleo próprio causa um crash. Note que para um sistema dos multi-chassis esta funcionalidade está apoiada atualmente somente para o chassi 0 da placa de linha. O outro chassi é apoiado em uma liberação futura do software.

Sugere-se que as descargas do núcleo para os RP e MSC estejam permitidas com o uso dos estes configuração as configurações em padrão e em admin-MODE:

```
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
```

Configuração da descarga do núcleo

Isto conduz a esta ocorrência para um impacto do núcleo:

1. Um RP causa um crash e uma descarga é escrita ao disco duro nesse RP no diretório raiz do disco.
2. Se um MSC causa um crash, uma descarga está escrita ao disco duro de RP0 no diretório raiz do disco.

Isto não tem nenhum impacto em tempos do Failover RP desde que a transmissão sem parar (NSF) é configurada para os protocolos de roteamento. Pode tomar alguns minutos extra para que o RP ou a placa de linha causada um crash torne-se disponível outra vez depois que segue um impacto quando escrever o núcleo.

Um exemplo da adição desta configuração ao padrão e à configuração de modo admin é mostrado aqui. Note que a configuração de modo admin exige DRP ser usada.

Esta saída mostra um exemplo de configuração da descarga do 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)#
```

Segurança

LPTS

Do pacote local de transporte dos serviços (LPTS) dos punhos pacotes destinados localmente. LPTS é feito de vários componentes diferentes.

1. Principal é chamado o processo do árbitro da porta. Escuta pedidos do soquete dos processos de protocolo diferentes, por exemplo, BGP, IS-IS e mantém-se a par de toda a informação obrigatória para aqueles processos. Por exemplo, se um processo BGP escuta no número do soquete 179, o PA obtém essa informação dos processos BGP, e atribui então um emperramento a esse processo em um IFIB.
2. O IFIB, é um outro componente do processo LPTS. Ajuda a manter um diretório de onde um

processo esteja que esteja escutando um emperramento específico da porta. O IFIB é gerado pelo processo do árbitro da porta e mantido com o árbitro da porta. Gerencie então subconjuntos múltiplos desta informação.O primeiro subconjunto é uma fatia do IFIB. Esta fatia pode ser associada ao protocolo do IPv4 e assim por diante. As fatias são enviadas então para apropriar os gerentes do fluxo, que usam então a fatia IFIB a fim enviar o pacote ao processo apropriado.O segundo subconjunto é um PRE-IFIB, permite que o LC envie o pacote ao processo apropriado se somente um processo existe ou a um gerente apropriado do fluxo.

3. Os gerentes do fluxo ajudam mais a distribuir os pacotes se o olhar é acima NON-trivial, por exemplo, processos múltiplos para o BGP. Cada gerente do fluxo tem uma fatia ou umas fatias múltiplas do IFIB e corretamente para a frente dos pacotes aos processos apropriados associados com a fatia do IFIB.
4. Se uma entrada não é definida para a porta do destino então pode ser deixada cair ou enviado ao gerente do fluxo. Um pacote está enviado sem porta associada se há uma política associada para a porta. As ajudas do gerente do fluxo então gerenciem uma entrada nova da sessão.

Como um pacote interno é enviado?

Há dois tipos de fluxos, fluxos da camada 2 (HDLC, PPP) e mergulha 4 fluxos ICMP/PING e o roteamento flui.

1. Camada 2 HDLC/PPP — Estes pacotes são identificados pelo Protocol Identifier e enviados diretamente às filas CPU no pulverizador. Os pacotes de protocolo da camada 2 obtêm a alta prioridade e são pegarados então pelo CPU (através do calamar) e processados. Daqui o Keepalives para a camada 2 é respondido diretamente através do LC através do CPU. Isto evita a necessidade de ir dentro ao RP para respostas e jogos com o tema do gerenciamento de interface distribuído.
2. Os pacotes ICMP (camada 4) são recebidos no LC e são enviados através da consulta com o IFBI nas filas CPU no pulverizador. Estes pacotes então são enviados ao CPU (através do calamar) e processados. A resposta é enviada então através das filas da saída do pulverizador a fim ser enviada através da tela. Isto é caso que um outro aplicativo igualmente precisa a informação (replicated através da tela). Uma vez através da tela o pacote é destinado à saída apropriada LC e através da fila apropriada da esponja e do controle.
3. Distribuindo fluxos são olhados acima no IFIB e enviados então às filas moldadas de saída (8000 filas) um de que é reservado para pacotes de controle. Esta é não uma fila moldada e é prestada serviços de manutenção simplesmente cada vez que está completo. – alta prioridade. O pacote é enviado então com a tela em filas de alta prioridade em um grupo de filas CPU na esponja (similar ao calamar se enfileira no pulverizador), e então os processos pelo processo apropriado, pelo gerente do fluxo ou pelo processo real. Uma resposta é enviada para trás através da esponja da placa de linha da saída e então para fora da placa de linha. A esponja da saída LC tem uma fila especial reservada para segurar pacotes de controle. As filas na esponja são rachadas na alta prioridade, no controle e nos pacotes de baixa prioridade, pela base da porta de saída.
4. O PSE tem um grupo de vigilantes que são configurados para a taxa que limita a camada 4, a camada 2 e os pacotes de roteamento. Estes são pré-ajustados e mudados para ser

usuário configurável em um outro dia.

Um do problema mais comum com LPTS é os pacotes que estão deixados cair, quando você tenta sibilar o roteador. Os vigilantes LPTS são geralmente taxa que limita estes pacotes. Este é o caso a fim 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

Os pacotes IP são inerentemente incertos. O IPsec é um método usado para proteger os pacotes IP. O IPsec do CRS-1 é executado no trajeto de encaminhamento do software, consequentemente a sessão IPSec é terminada no RP/DRP. Um número total de 500 sessões IPSec pelo CRS-1 é apoiado. O número depende da velocidade CPU e dos recursos atribuídos. Não há nenhuma limitação do software a este, simplesmente o tráfego local-originado e local-terminado no RP é elegível para a manipulação do IPsec. O modo transporte de IPpsec ou o modo de túnel podem ser usados para o tipo de tráfego, embora o anterior são preferido devido a menos despesas gerais no processamento de IPSec.

R3.3.0 apoia a criptografia do BGP e do OSPFv3 sobre o IPsec.

Refira o manual de configuração da segurança de sistema do Cisco IOS XR para obter mais informações sobre de como executar o IPsec.

Nota: O IPsec exige a torta cripto, por exemplo, hfr-k9sec-p.pie-3.3.1.

Fora da faixa

Console e acesso AUX

O CRS-1 RP/SCs tem uma Console e Porta AUX disponível para fora de propósitos do gerenciamento da faixa, assim como uma porta Ethernet de gerenciamento para fora da banda através do IP.

A Console e Porta AUX de cada RP/SCGE, dois por chassis, pode ser conectada a um servidor de console. Isto significa que o sistema do chassi único exige quatro portas de Console, e os sistemas dos multi-chassis exigem 12 portas mais duas portas mais adicionais para os motores do supervisor no catalizador 6504-E.

A conexão do porto auxiliar é importante desde que fornece o acesso ao núcleo IOS-XR e pode permitir a recuperação de sistema quando esta não é possível através da porta de Console. O acesso através do porto auxiliar está somente disponível aos usuários definido localmente no sistema, e somente quando o usuário tem o acesso nivelado do raiz-sistema ou do Cisco-apoio. Além o usuário deve ter uma **senha secundária** definida.

Acesso de terminal virtual

O telnet & o Shell Seguro (ssh) podem ser usados a fim alcançar o CRS-1 através das portas VTY. Ambos são desabilitados à revelia, e o usuário precisa de permiti-los explicitamente.

Nota: O IPsec exige a torta cripto, por exemplo, hfr-k9sec-p.pie-3.3.1.

Gerencia primeiramente chaves RSA e DSA segundo as indicações deste exemplo a fim permitir o 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
```

Informações Relacionadas

- Apoio do Roteadores
- Suporte Técnico e Documentação - Cisco Systems