Identificar e Solucionar Problemas de Operações do Plano de Controle nos Switches Catalyst 9000

Contents

Introdução

Este documento descreve como solucionar problemas e validar a integridade do plano de controle nos switches da família Catalyst 9000 que executam o Cisco IOS® XE.

Informações de Apoio

O principal trabalho de um switch é encaminhar pacotes o mais rápido possível. A maioria dos pacotes é encaminhada no hardware, mas determinados tipos de tráfego devem ser manipulados pela CPU do sistema. O tráfego que chega à CPU é tratado o mais rápido possível. Espera-se que uma certa quantidade de tráfego seja vista na CPU, mas uma superabundância leva a problemas operacionais. A família de switches Catalyst 9000 incorpora um mecanismo robusto de Políticas de Plano de Controle (CoPP - Control Plane Policing) por padrão para evitar problemas causados pela saturação do tráfego da CPU.

Problemas inesperados surgem em certos casos de uso em função da operação normal. A correlação entre causa e efeito não é óbvia às vezes, o que torna o problema difícil de abordar. Este documento fornece ferramentas para validar a integridade do plano de controle e fornece um fluxo de trabalho sobre como abordar problemas que envolvem o caminho de inserção ou punt do plano de controle. Ele também fornece vários cenários comuns com base nos problemas observados no campo.

Tenha em mente que o caminho de punt da CPU é um recurso limitado. Os switches modernos de encaminhamento de hardware podem lidar com um volume de tráfego exponencialmente maior. A família de switches Catalyst 9000 suporta aproximadamente 19.000 pacotes por segundo (pps) agregados na CPU a qualquer momento. Exceda esse limite, e o tráfego pontuado é policiado sem peso.

Terminologia

- Forwarding Engine Driver (FED): este é o coração do switch Cisco Catalyst e é responsável por toda a programação/encaminhamento de hardware
- IOSd: Este é o daemon Cisco IOS que é executado no kernel Linux. Ele é executado como um processo de software dentro do kernel
- Packet Delivery System (PDS): essa é a arquitetura e o processo de como os pacotes são entregues de e para os vários subsistemas. Como exemplo, ele controla como os pacotes são entregues do FED ao IOSd e vice-versa
- Plano de Controle (CP):O plano de controle é um termo genérico usado para agrupar as funções e o tráfego que envolvem a CPU do Switch Catalyst. Isso inclui o tráfego como o Spanning Tree Protocol (STP), o Hot Standby Router Protocol (HSRP) e os protocolos de roteamento que são destinados ao switch ou enviados do switch. Isso também inclui protocolos da camada de aplicação como Secure Shell (SSH) e Simple Network Management Protocol (SNMP) que devem ser tratados pela CPU
- Plano de dados (DP):Normalmente, o plano de dados abrange os ASICs de hardware e o tráfego que é encaminhado sem assistência do Plano de controle
- Punt:Pacote de controle de protocolo de entrada que interceptou no DP enviado ao CP para processá-lo
- Inserir:pacote de protocolo gerado pelo CP enviado ao DP para saída em interface(s) de E/S
- LSMPI:Interface de Pontuação de Memória Compartilhada do Linux

CoPP do Catalyst 9000

A base da proteção da CPU na família de switches Catalyst 9000 é o CoPP. Com o CoPP, uma política de Qualidade de Serviço (QoS) gerada pelo sistema é aplicada no caminho de punt/inserção da CPU. O tráfego vinculado à CPU é agrupado em várias classes diferentes e, subsequentemente, mapeado nos vigilantes de hardware individuais associados à CPU. Os vigilantes evitam a saturação da CPU por uma classe específica de tráfego.

Implementação de CoPP

O tráfego vinculado à CPU é classificado em filas. Essas filas/classes são definidas pelo sistema e não podem ser configuradas pelo usuário. Os vigilantes são configurados no hardware. A família Catalyst 9000 suporta 32 vigilantes de hardware para 32 filas.

Os valores específicos diferem de plataforma para plataforma. Em geral, há 32 filas definidas pelo sistema. Essas filas se relacionam a mapas de classe, que se relacionam a índices de vigilante. Os índices do vigilante têm uma taxa de vigilante padrão. Essa taxa é configurável pelo usuário, embora as alterações na política de CoPP padrão aumentem a susceptibilidade a um impacto inesperado no serviço.

Valores definidos pelo sistema para CoPP

Nomes dos mapas de classe	Índice do vigilante (Nº do vigilante)	Filas de CPU (Nº da fila)
system-cpp-police-data	WK_CPP_POLICE_DATA(0)	WK_CPU_Q_ICMP_GEN(3) WK_CPU_Q_BROADCAST(12) WK_CPU_Q_ICMP_REDIRECT(6)
system-cpp-police-l2-control	WK_CPP_POLICE_L2_ CONTROL(1)	WK_CPU_Q_L2_CONTROL(1)
system-cpp-police-routing-control	WK_CPP_POLICE_ROUTING_CONTROL(2)	WK_CPU_Q_ROUTING_CONTROL(4) WK_CPU_Q_LOW_LATENCY (27)
system-cpp-police-control-low-priority	WK_CPP_POLICE_CO NTROL_LOW_PRI(3)	WK_CPU_Q_GENERAL_PUNT(25)
system-cpp-police-punt-webauth	WK_CPP_POLICE_PU NT_WEBAUTH(7)	WK_CPU_Q_PUNT_WEBAUTH(22)
system-cpp-police-controle-	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTROL(15)

Nomes dos mapas de classe	Índice do vigilante (Nº do vigilante)	Filas de CPU (Nº da fila)
de-topologia		
system-cpp-police-multicast	WK_CPP_POLICE_MULTICAST(9)	WK_CPU_Q_TRANSIT_TRAFFIC(18) WK_CPU_Q_MCAST_DATA(30)
system-cpp-police-sys- data	WK_CPP_POLICE_SYS _DATA(10)	WK_CPU_Q_LEARNING_CACHE_OVFL(1 WK_CPU_Q_CRYPTO_CONTROL(23) WK_CPU_Q_EXCEPTION(24) WK_CPU_Q_EGR_EXCEPTION(28) WK_CPU_Q_NFL_SAMPLED_DATA(26) WK_CPU_Q_GOLD_PKT(31) WK_CPU_Q_RPF_FAILED(19)
system-cpp-police-dot1x-auth	WK_CPP_POLICE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH(0)
system-cpp-police-protocol-snooping	WK_CPP_POLICE_PR(12)	WK_CPU_Q_PROTO_SNOOPING(16)
system-cpp-police-sw-forward	WK_CPP_POLICE_SW_FWD (13)	WK_CPU_Q_SW_FORWARDING_Q(14) WK_CPU_Q_LOGGING(21) WK_CPU_Q_L2_LVX_DATA_PACK(11)
system-cpp-police-forus	WK_CPP_POLICE_FORUS(14)	WK_CPU_Q_FORUS_ADDR_RESOLUTIO WK_CPU_Q_FORUS_TRAFFIC(2)

Nomes dos mapas de classe	Índice do vigilante (Nº do vigilante)	Filas de CPU (Nº da fila)
system-cpp-police-multicast-end-station	WK_CPP_POLICE_MULTICAST_SNOOPING(15)	WK_CPU_Q_MCAST_END_STATION_SER
system-cpp-default	WK_CPP_POLICE_DEFAULT_POLICER(16)	WK_CPU_Q_DHCP_SNOOPING(17) WK_CPU_Q_UNUSED(7) WK_CPU_Q_EWLC_CONTROL(9) WK_CPU_Q_EWLC_DATA(10)
system-cpp-police-stackwise-virt-control	WK_CPP_STACKWISE_VIRTUAL_CONTROL(5)	WK_CPU_Q_STACKWISE_VIRTUAL_CON (29)
system-cpp-police-l2lvx-control	WK_CPP_ L2_LVX_CONT_PACK(4)	WK_CPU_Q_L2_LVX_CONT_PACK(8)

Cada fila está relacionada a um tipo de tráfego ou a um conjunto específico de recursos. Esta não é uma lista completa:

Filas de CPU e recursos associados

Filas de CPU (Nº da fila)	Recurso(s)
WK_CPU_Q_DOT1X_AUTH(0)	Autenticação baseada em porta IEEE 802.1x
WK_CPU_Q_L2_CONTROL(1)	Dynamic Trunking Protocol (DTP) Protocolo VLAN Trunking (VTP) Port Aggregation Protocol (PAgP)

Filas de CPU (Nº da fila)	Recurso(s)
	Protocolo de sinalização de informações do cliente (CISP) Protocolo de retransmissão de sessão de mensagem Protocolo de Registro de VLAN Múltiplo (MVRP) Rede Móvel Metropolitana (MMN) Protocolo LLDP Detecção de enlace unidirecional (UDLD) LACP (Link Aggregation Control Protocol, protocolo de controle de agregação de link) Cisco Discovery Protocol (CDP) STP (Spanning Tree Protocol)
WK_CPU_Q_FORUS_TRAFFIC(2)	Host como Telnet, Pingv4 e Pingv6 e SNMP Detecção de keepalive / loopback Inicie o protocolo IKE (Internet Key Exchange) (IPSec)
WK_CPU_Q_ICMP_GEN(3)	ICMP - destino inalcançável ICMP-TTL expirado
WK_CPU_Q_ROUTING_CONTROL(4)	Routing Information Protocol versão 1 (RIPv1) RIPv2 IGRP (Interior Gateway Routing Protocol) Protocolo de gateway de borda (BGP) PIM-UDP Virtual Router Redundancy Protocol (VRRP) Hot Standby Router Protocol versão 1

Filas de CPU (Nº da fila)	Recurso(s)
	(HSRPv1) HSRPv2 Protocolo de balanceamento de carga do gateway (GLBP) Protocolo de distribuição de rótulo (LDP - Label Distribution Protocol) Protocolo de Comunicação de Cache de Web (WCCP - Web Cache Communication Protocol) RIPng (Routing Information Protocol, protocolo de informações de roteamento de próxima geração) Abra o protocolo OSPF Open Shortest Path First versão 3(OSPFv3) Enhanced Interior Gateway Routing Protocol (EIGRP) Enhanced Interior Gateway Routing Protocol versão 6 (EIGRPv6) DHCPv6 Multicast independente de protocolo (PIM) Protocol Independent Multicast versão 6 (PIMv6) Protocolo de Roteador de Hot Standby de próxima geração (HSRPng) Controle de IPv6 Manutenção de atividade do Generic Routing Encapsulation (GRE) Pont de conversão de endereço de rede (NAT) Sistema intermediário para sistema intermediário (IS-IS)

Filas de CPU (Nº da fila)	Recurso(s)
WK_CPU_Q_FORUS_ADDR_RESOLUTION(5)	Address Resolution Protocol (ARP) Propaganda de vizinho IPv6 e solicitação de vizinho
WK_CPU_Q_ICMP_REDIRECT(6)	redirecionamento de Internet Control Message Protocol (ICMP)
WK_CPU_Q_INTER_FED_TRAFFIC(7)	Inserção de domínio de bridge da camada 2 para comunicação interna.
WK_CPU_Q_L2_LVX_CONT_PACK(8)	Pacote XID (Exchange ID)
WK_CPU_Q_EWLC_CONTROL(9)	Controlador sem fio integrado (eWLC) [Controle e provisionamento de pontos de acesso sem fio (CAPWAP) (UDP 5246)]
WK_CPU_Q_EWLC_DATA(10)	Pacote de dados eWLC (dados CAPWAP, UDP 5247)
WK_CPU_Q_L2_LVX_DATA_PACK(11)	Pacote unicast desconhecido apontado para solicitação de mapa.
WK_CPU_Q_BROADCAST(12)	Todos os tipos de broadcast
WK_CPU_Q_OPENFLOW(13)	Estouro de cache de aprendizagem (Camada 2 + Camada 3)
WK_CPU_Q_CONTROLLER_PUNT(14)	Dados - lista de controle de acesso (ACL) completa Dados - opções de IPv4 Dados - IPv6 salto a salto Dados - sem recursos / capturar tudo Dados - Encaminhamento de Caminho Reverso (RPF) incompleto

Filas de CPU (Nº da fila)	Recurso(s)
	Pacote Glean
WK_CPU_Q_TOPOLOGY_CONTROL(15)	STP (Spanning Tree Protocol) Protocolo Ethernet resiliente (REP) Protocolo Spanning Tree Compartilhado (SSTP)
WK_CPU_Q_PROTO_SNOOPING(16)	Rastreamento Address Resolution Protocol (ARP) para Dynamic ARP Inspection (DAI)
WK_CPU_Q_DHCP_SNOOPING(17)	rastreamento de DHCP
WK_CPU_Q_TRANSIT_TRAFFIC(18)	Isso é usado para pacotes apontados pelo NAT, que precisam ser manipulados no caminho do software.
WK_CPU_Q_RPF_FAILED(19)	Dados - falha de mRPF (RPF multicast)
WK_CPU_Q_MCAST_END_STATION _SERVICE(20)	Internet Group Management Protocol (IGMP) / Controle de descoberta de ouvinte multicast (MLD)
WK_CPU_Q_LOGGING(21)	Registro da lista de controle de acesso (ACL)
WK_CPU_Q_PUNT_WEBAUTH(22)	Autenticação da Web
WK_CPU_Q_HIGH_RATE_APP(23)	Broadcast
WK_CPU_Q_EXCEPTION(24)	indicação IKE Violação de aprendizagem de IP Violação de segurança de porta IP Violação de endereço IP estático Verificação de escopo IPv6

Filas de CPU (Nº da fila)	Recurso(s)
	Exceção de Remote Copy Protocol (RCP) Falha de RPF unicast
WK_CPU_Q_SYSTEM_CRITICAL(25)	Sinalização de mídia/Proxy sem fio ARP
WK_CPU_Q_NFL_SAMPLED_DATA(26)	Dados de amostra do Netflow e Media Services Proxy (MSP)
WK_CPU_Q_LOW_LATENCY(27)	Detecção de encaminhamento bidirecional (BFD - Bidirectional Forwarding Detection), protocolo de tempo de precisão (PTP - Precision Time Protocol)
WK_CPU_Q_EGR_EXCEPTION(28)	Exceção de resolução de saída
WK_CPU_Q_STACKWISE_VIRTUAL _CONTROL(29)	Protocolos de empilhamento frontal, ou seja, SVL
WK_CPU_Q_MCAST_DATA(30)	Criação de dados - (S,G) Dados - joins locais Dados - Registro PIM Dados - switchover de SPT Dados - Multicast
WK_CPU_Q_GOLD_PKT(31)	Ouro

Política padrão

Por padrão, a política de CoPP gerada pelo sistema é aplicada ao caminho de punt/inserção. A política padrão pode ser visualizada usando comandos comuns baseados em MQC. Ele também pode ser visto na configuração do switch. A única política que pode ser aplicada na entrada ou saída da CPU/plano de controle é a política definida pelo sistema.

Use "show policy-map control-plane" para exibir a política aplicada ao plano de controle:

```
<#root>

Catalyst-9600#

show policy-map control-plane


Control Plane

 Service-policy input: system-cpp-policy

   Class-map: system-cpp-police-ios-routing (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: none
      police:
          rate 17000 pps, burst 4150 packets
        conformed 95904305 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          drop

<snip>

   Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Ajustar CoPP

As taxas de vigilante CoPP são configuráveis pelo usuário. Os usuários também podem desativar filas.

Este exemplo demonstra como ajustar um valor de vigilante individual. Neste exemplo, a classe ajustada é "system-cpp-police-protocol-snooping".

```
<#root>

Device>

enable


Device#

configure terminal



Device(config)#

policy-map system-cpp-policy


Device(config-pmap)#
```

```
Device(config-pmap)#
class system-cpp-police-protocol-snooping

Device(config-pmap-c)#
Device(config-pmap-c)#
police rate 100 pps

Device(config-pmap-c-police)#
Device(config-pmap-c-police)#
exit

Device(config-pmap-c)#
exit

Device(config-pmap)#
exit

Device(config)#
Device(config)#
control-plane

Device(config-cp)#
Device(config)#
control-plane

Device(config-cp)#
service-policy input system-cpp-policy

Device(config-cp)#
Device(config-cp)#
end


Device#
show policy-map control-plane
```

Este exemplo demonstra como desativar totalmente uma fila. Tenha cuidado ao desativar filas, pois isso pode levar a uma possível saturação da CPU.

```
<#root>
Device>
enable

Device#
configure terminal


Device(config)#
policy-map system-cpp-policy

Device(config-pmap)#
Device(config-pmap)#
class system-cpp-police-protocol-snooping

Device(config-pmap-c)#

Device(config-pmap-c)#
no police rate 100 pps
Device(config-pmap-c)#
end
```

Troubleshooting

Metologia

A utilização da CPU é afetada por duas atividades básicas: processos e interrupção. Os processos são atividades estruturadas que a CPU executa enquanto a interrupção se refere a pacotes interceptados no plano de dados e enviados à CPU para ação. Juntas, essas atividades compreendem a utilização total da CPU. Como o CoPP é ativado por padrão, um impacto no serviço não se correlaciona necessariamente com a alta utilização da CPU. Se o CoPP fizer seu trabalho, a utilização da CPU não sofrerá grande impacto. É importante considerar a utilização geral da CPU, mas a utilização geral não conta toda a história. Os comandos e utilitários show nesta seção são usados para avaliar rapidamente a integridade da CPU e identificar detalhes relevantes sobre o tráfego vinculado à CPU.

Diretrizes:

- Determine se o problema está relacionado ao plano de controle. A maior parte do tráfego de trânsito é encaminhado no hardware. Apenas certos tipos de tráfego e certos cenários envolvem a CPU e o plano de controle, portanto, tenha isso em mente durante toda a investigação.

- Entenda sua linha de base de utilização. É importante entender como é a utilização normal para que possam ser identificados desvios da norma.
- Valide a utilização geral para processos e interrupções. Identificar todos os processos que ocupam volumes inesperados de ciclos da CPU. Se a utilização ficar fora do intervalo esperado, isso pode ser motivo de preocupação. É importante entender a utilização média de um sistema, para que desvios fora da norma sejam reconhecidos. Tenha em mente que a utilização isolada não é uma imagem completa da integridade do plano de controle.
- Determine se há aumentos ativos de quedas em CoPP. As quedas de CoPP nem sempre são indicativas de um problema, mas se você solucionar um problema relacionado a uma classe de tráfego que está ativamente vigiada, esse é um forte indicador de relevância.

Comandos show úteis

O switch permite uma supervisão rápida da integridade da CPU e das estatísticas de CoPP. Há também uma CLI útil para determinar rapidamente o ponto de ingresso do tráfego vinculado à CPU.

Determine a utilização geral e histórica

- "Show processes cpu sorted" é usado para visualizar a utilização geral da CPU. O argumento "sorted" classifica a saída do processo com base no percentual de uso. Os processos que usam mais recursos da CPU estão na parte superior da saída. A utilização devido a interrupções também é fornecida como uma porcentagem.

<#root>

```
Catalyst-9600#

show processes cpu sorted


CPU utilization for five seconds: 92%/13%; one minute: 76%; five minutes: 73%

<<<--- Utilization is displayed for 5 second (both process and interrupt), 1 minute and 5 minute interva


                                                                                92% refers to the c


                                                                                The 13% value refer

PID Runtime(ms)    Invoked      uSecs   5Sec   1Min   5Min  TTY  Process

<<<--- Runtime statistics, as well as utilization averages are displayed here. The process is also ident


344   547030523  607054509        901  38.13% 30.61% 29.32%   0  SISF Switcher Th
345   394700227  615024099        641  31.18% 22.68% 21.66%   0  SISF Main Thread
 98   112308516  119818535        937   4.12%  4.76%  5.09%   0  Crimson flush tr
247    47096761   92250875        510   2.42%  2.21%  2.18%   0  Spanning Tree
123    35303496  679878082         51   1.85%  1.88%  1.84%   0  IOSXE-RP Punt Se
234         955       1758        543   1.61%  0.71%  0.23%   3  SSH Process
547     5360168    5484910        977   1.04%  0.46%  0.44%   0  DHCPD Receive
```

```
229    27381066  963726156         28   1.04%  1.34%  1.23%   0 IP Input
 79    13183805  108951712        121   0.48%  0.55%  0.55%   0 IOSD ipc task
  9     1073134     315186       3404   0.40%  0.06%  0.03%   0 Check heaps
 37    11099063  147506419         75   0.40%  0.54%  0.52%   0 ARP Input
312     2986160  240782059         12   0.24%  0.12%  0.14%   0 DAI Packet Proce
<snip>
565           0          1          0   0.00%  0.00%  0.00%   0 LICENSE AGENT
566          14       1210         11   0.00%  0.00%  0.00%   0 DHCPD Timer
567          40         45        888   0.00%  0.00%  0.00%   0 OVLD SPA Backgro
568          12       2342          5   0.00%  0.00%  0.00%   0 DHCPD Database
569           0         12          0   0.00%  0.00%  0.00%   0 SpanTree Flush
571           0          1          0   0.00%  0.00%  0.00%   0 EM Action CNS
572         681     140276          4   0.00%  0.00%  0.00%   0 Inline power inc
```

- "Show processes cpu history" fornece um gráfico histórico da utilização da CPU nos últimos 60 segundos, 5 minutos e 72 horas.

```
<#root>

Catalyst-9600#

show processes cpu history


    999777776666688888666667777777778888877777666669999988888866

<<<--- The numbers at the top of each column represent the highest value seen throughout the time period

    222555559999944444444440000088888888811111777773333335555500

It is read top-down. "9" over "2" in this example means "92%" for example.


100
90  *** ***** **********
80  ******** ***** ********** **********
70  ****************** ***********************************
60  ************************************************************
50  ************************************************************
40  ************************************************************
30  ************************************************************
20  ************************************************************
10  ************************************************************

<<<--- The "*" represents the highest value during the given time period. This relates to a momentary sp

0....5....1....1....2....2....3....3....4....4....5....5....6

In this example, utilization spiked to 92% in the last 5 seconds.


          0    5    0    5    0    5    0    5    0    5    0
          CPU% per second (last 60 seconds)
         * = maximum CPU% # = average CPU%
```

```
    9998989899998989989989898899999898898989999999899999999
    4318230911026353162351292837713365748928096040142309011335l1
100 ** *
90  ***** **************************************************
80  ************#***#*#**#***####*##*****#**#***#***#**********
70  ############################################################
```

<<<--- The "#" represents the average utilization. This indicates sustained utilization.

```
60  ############################################################
```

In this example, within the last 5 minutes the average utilization was sustained around 70% while

```
50  ############################################################
```

the maximum utilization spiked to 94%.

```
40  ############################################################
30  ############################################################
20  ############################################################
10  ############################################################
0....5....1....1....2....2....3....3....4....4....5....5....6
          0    5    0    5    0    5    0    5    0    5    0
          CPU% per minute (last 60 minutes)
         * = maximum CPU% # = average CPU%
```

```
    9999999999999999999999999999999999999999999999999999999999999999999999999999
    6656565666465556666556565756545565677375555675745455457759575546485767l57
100 ********** ****************** ******* ********* * ** ********* * *****
90  ********************************************************************************
80  ********************************************************************************
70  ################################################################################
60  ################################################################################
50  ################################################################################
40  ################################################################################
30  ################################################################################
20  ################################################################################
10  ################################################################################
0....5....1....1....2....2....3....3....4....4....5....5....6....6....7..
          0    5    0    5    0    5    0    5    0    5    0    5    0
               CPU% per hour (last 72 hours)
              * = maximum CPU% # = average CPU%
```

Verificar Políticas de Plano de Controle

- Use "show platform hardware fed <switch> ative qos queue stats internal cpu policer" para exibir estatísticas CoPP agregadas e informações adicionais sobre a estrutura de

fila/vigilante. Esta saída fornece uma visão histórica das estatísticas do vigilante desde a última reinicialização do plano de controle. Esses contadores também podem ser limpos manualmente. Geralmente, a evidência de quedas do plano de controle pelo vigilante aponta para um problema com a fila/classe associada, mas certifique-se de que as quedas sejam incrementadas ativamente enquanto o problema ocorre. Execute o comando várias vezes para observar o aumento dos valores de Queue Drop.

<#root>

```
Catalyst9500#

show platform hardware fed active qos queue stats internal cpu policer


                          CPU Queue Statistics
============================================================================================
                                                 (default) (set)     Queue        Queue
QId PlcIdx  Queue Name                Enabled    Rate      Rate      Drop(Bytes)  Drop(Frames)

<-- The top section of this output gives a historical view of CoPP drops. Run the command several times

--------------------------------------------------------------------------------------------

CPU queues correlate with a Policer Index (PlcIdx) and Queue (QId).

0    11     DOT1X Auth                  Yes     1000      1000     0            0

Note that multiple policer indices map to the same queue for some classes.

1    1      L2 Control                  Yes     2000      2000     0            0
2    14     Forus traffic               Yes     4000      4000     0            0
3    0      ICMP GEN                    Yes     750       750      0            0
4    2      Routing Control             Yes     5500      5500     0            0
5    14     Forus Address resolution    Yes     4000      4000     83027876     1297199
6    0      ICMP Redirect               Yes     750       750      0            0
7    16     Inter FED Traffic           Yes     2000      2000     0            0
8    4      L2 LVX Cont Pack            Yes     1000      1000     0            0
9    19     EWLC Control                Yes     13000     13000    0            0
10   16     EWLC Data                   Yes     2000      2000     0            0
11   13     L2 LVX Data Pack            Yes     1000      1000     0            0
12   0      BROADCAST                   Yes     750       750      0            0
13   10     Openflow                    Yes     250       250      0            0
14   13     Sw forwarding               Yes     1000      1000     0            0
15   8      Topology Control            Yes     13000     16000    0            0
16   12     Proto Snooping              Yes     2000      2000     0            0
17   6      DHCP Snooping               Yes     500       500      0            0
18   13     Transit Traffic             Yes     1000      1000     0            0
19   10     RPF Failed                  Yes     250       250      0            0
20   15     MCAST END STATION           Yes     2000      2000     0            0
21   13     LOGGING                     Yes     1000      1000     769024       12016
22   7      Punt Webauth                Yes     1000      1000     0            0
23   18     High Rate App               Yes     13000     13000    0            0
24   10     Exception                   Yes     250       250      0            0
25   3      System Critical             Yes     1000      1000     0            0
26   10     NFL SAMPLED DATA            Yes     250       250      0            0
27   2      Low Latency                 Yes     5500      5500     0            0
28   10     EGR Exception               Yes     250       250      0            0
29   5      Stackwise Virtual OOB       Yes     8000      8000     0            0
30   9      MCAST Data                  Yes     500       500      0            0
31   3      Gold Pkt                    Yes     1000      1000     0            0
```

* NOTE: CPU queue policer rates are configured to the closest hardware supported value

CPU Queue Policer Statistics

Policer Index	Policer Accept Bytes	Policer Accept Frames	Policer Drop Bytes	Policer Drop Frames
0	59894	613	0	0
1	15701689	57082	0	0
2	5562892	63482	0	0
3	3536	52	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	2347194476	32649666	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	577043	8232	769024	12016
14	719225176	11182355	83027876	1297199
15	132766	1891	0	0
16	0	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	0	0	0

Second Level Policer Statistics

<-- Second level policer information begins here. Catalyst CoPP is organized with two policers to allow

20	2368459057	32770230	0	0
21	719994879	11193091	0	0

Policer Index Mapping and Settings

level-2 PlcIndex	level-1 PlcIndex	(default) rate	(set) rate
20	1 2 8	13000	17000
21	0 4 7 9 10 11 12 13 14 15	6000	6000

Second Level Policer Config

QId	level-1 PlcIdx	level-2 PlcIdx	Queue Name	level-2 Enabled
0	11	21	DOT1X Auth	Yes
1	1	20	L2 Control	Yes
2	14	21	Forus traffic	Yes
3	0	21	ICMP GEN	Yes
4	2	20	Routing Control	Yes
5	14	21	Forus Address resolution	Yes
6	0	21	ICMP Redirect	Yes
7	16	-	Inter FED Traffic	No
8	4	21	L2 LVX Cont Pack	Yes
9	19	-	EWLC Control	No
10	16	-	EWLC Data	No
11	13	21	L2 LVX Data Pack	Yes
12	0	21	BROADCAST	Yes

```
13   10      21      Openflow                    Yes
14   13      21      Sw forwarding               Yes
15   8       20      Topology Control            Yes
16   12      21      Proto Snooping              Yes
17   6       -       DHCP Snooping               No
18   13      21      Transit Traffic             Yes
19   10      21      RPF Failed                  Yes
20   15      21      MCAST END STATION           Yes
21   13      21      LOGGING                     Yes
22   7       21      Punt Webauth                Yes
23   18      -       High Rate App               No
24   10      21      Exception                   Yes
25   3       -       System Critical             No
26   10      21      NFL SAMPLED DATA            Yes
27   2       20      Low Latency                 Yes
28   10      21      EGR Exception               Yes
29   5       -       Stackwise Virtual OOB       No
30   9       21      MCAST Data                  Yes
31   3       -       Gold Pkt                    No

                        CPP Classes to queue map

<-- Information on how different traffic types map to different queues are found here.
=========================================================================================

PlcIdx CPP Class                                 :  Queues
-----------------------------------------------------------------------------------------
0      system-cpp-police-data                    :  ICMP GEN/ BROADCAST/ ICMP Redirect/
10     system-cpp-police-sys-data                :  Openflow/ Exception/ EGR Exception/ NFL SAMPLED DATA/
13     system-cpp-police-sw-forward              :  Sw forwarding/ LOGGING/ L2 LVX Data Pack/ Transit Tra
9      system-cpp-police-multicast               :  MCAST Data/
15     system-cpp-police-multicast-end-station   :  MCAST END STATION /
7      system-cpp-police-punt-webauth            :  Punt Webauth/
1      system-cpp-police-l2-control              :  L2 Control/
2      system-cpp-police-routing-control         :  Routing Control/ Low Latency/
3      system-cpp-police-system-critical         :  System Critical/ Gold Pkt/
4      system-cpp-police-l2lvx-control           :  L2 LVX Cont Pack/
8      system-cpp-police-topology-control        :  Topology Control/
11     system-cpp-police-dot1x-auth              :  DOT1X Auth/
12     system-cpp-police-protocol-snooping       :  Proto Snooping/
6      system-cpp-police-dhcp-snooping           :  DHCP Snooping/
14     system-cpp-police-forus                   :  Forus Address resolution/ Forus traffic/
5      system-cpp-police-stackwise-virt-control  :  Stackwise Virtual OOB/
16     system-cpp-default                        :  Inter FED Traffic/ EWLC Data/
18     system-cpp-police-high-rate-app           :  High Rate App/
19     system-cpp-police-ewlc-control            :  EWLC Control/
20     system-cpp-police-ios-routing             :  L2 Control/ Topology Control/ Routing Control/ Low La
21     system-cpp-police-ios-feature             :  ICMP GEN/ BROADCAST/ ICMP Redirect/ L2 LVX Cont Pack/
```

Coletar informações sobre tráfego direcionado

Esses comandos são usados para coletar informações sobre o tráfego apontado para a CPU, incluindo o tipo de tráfego e os pontos físicos de entrada.

- "Show platform software fed <switch> ative punt cpuq all" ou "Show platform software fed <switch> ative punt cpuq <0-31 Queue ID>" podem ser usados para ver estatísticas relacionadas a todas ou a uma fila de CPU específica.

<#root>

C9300#

show platform software fed switch active punt cpuq all

```
Punt CPU Q Statistics
=============================================

CPU Q Id                        : 0
CPU Q Name                      : CPU_Q_DOT1X_AUTH
Packets received from ASIC      : 964
Send to IOSd total attempts     : 964
Send to IOSd failed count       : 0
RX suspend count                : 0
RX unsuspend count              : 0
RX unsuspend send count         : 0
RX unsuspend send failed count  : 0
RX consumed count               : 0
RX dropped count                : 0
RX non-active dropped count     : 0
RX conversion failure dropped   : 0
RX INTACK count                 : 964
RX packets dq'd after intack    : 0
Active RxQ event                : 964
RX spurious interrupt           : 0
RX phy_idb fetch failed: 0
RX table_id fetch failed: 0
RX invalid punt cause: 0

CPU Q Id                        : 1
CPU Q Name                      : CPU_Q_L2_CONTROL
Packets received from ASIC      : 80487
Send to IOSd total attempts     : 80487
Send to IOSd failed count       : 0
RX suspend count                : 0
RX unsuspend count              : 0
RX unsuspend send count         : 0
RX unsuspend send failed count  : 0
RX consumed count               : 0
RX dropped count                : 0
RX non-active dropped count     : 0
RX conversion failure dropped   : 0
RX INTACK count                 : 80474
RX packets dq'd after intack    : 16
Active RxQ event                : 80474
RX spurious interrupt           : 9
RX phy_idb fetch failed: 0
RX table_id fetch failed: 0
RX invalid punt cause: 0

CPU Q Id                        : 2
CPU Q Name                      : CPU_Q_FORUS_TRAFFIC
Packets received from ASIC      : 176669
Send to IOSd total attempts     : 176669
Send to IOSd failed count       : 0
RX suspend count                : 0
RX unsuspend count              : 0
RX unsuspend send count         : 0
RX unsuspend send failed count  : 0
RX consumed count               : 0
RX dropped count                : 0
```

```
RX non-active dropped count    : 0
RX conversion failure dropped  : 0
RX INTACK count                : 165584
RX packets dq'd after intack   : 12601
Active RxQ event               : 165596
RX spurious interrupt          : 11851
RX phy_idb fetch failed: 0
RX table_id fetch failed: 0
RX invalid punt cause: 0
<snip>

C9300#
```

show platform software fed switch active punt cpuq 16 <-- Queue ID 16 correlates with Protocol Snooping.

```
Punt CPU Q Statistics
=============================================

CPU Q Id                       : 16
CPU Q Name                     : CPU_Q_PROTO_SNOOPING
Packets received from ASIC     : 55661
Send to IOSd total attempts    : 55661
Send to IOSd failed count      : 0
RX suspend count               : 0
RX unsuspend count             : 0
RX unsuspend send count        : 0
RX unsuspend send failed count : 0
RX consumed count              : 0
RX dropped count               : 0
RX non-active dropped count    : 0
RX conversion failure dropped  : 0
RX INTACK count                : 55659
RX packets dq'd after intack   : 9
Active RxQ event               : 55659
RX spurious interrupt          : 23
RX phy_idb fetch failed: 0
RX table_id fetch failed: 0
RX invalid punt cause: 0

Replenish Stats for all rxq:
-----------------------------------------
Number of replenish            : 4926842
Number of replenish suspend    : 0
Number of replenish un-suspend : 0
-----------------------------------------
```

- Use "show platform software fed <switch> ative punt cause summary" para obter uma visão rápida de todos os diferentes tipos de tráfego vistos na CPU. Observe que somente causas diferentes de zero são mostradas.

<#root>

```
C9300#
```

show platform software fed switch active punt cause summary

```
Statistics for all causes

Cause  Cause Info                    Rcvd                 Dropped
```

```
------------------------------------------------------------------------------
7      ARP request or response         142962              0
11     For-us data                     490817              0
21     RP<->QFP keepalive              448742              0
24     Glean adjacency                 2                   0
55     For-us control                  415222              0
58     Layer2 bridge domain data packe 3654659             0
60     IP subnet or broadcast packet   37167               0
75     EPC                             17942               0
96     Layer2 control protocols        358614              0
97     Packets to LFTS                 964                 0
109    snoop packets                   48867               0
------------------------------------------------------------------------------
```

- Use o comando "show platform software fed <switch> ative punt rates interfaces" para visualizar rapidamente as interfaces que o tráfego vinculado à CPU entra no sistema. Esse comando mostra apenas interfaces com uma fila de entrada diferente de zero.

<#root>

```
C9300#

show platform software fed switch active punt rates interfaces

Punt Rate on Interfaces Statistics

Packets per second averaged over 10 seconds, 1 min and 5 mins
================================================================================================
                             |           |  Recv |  Recv |  Recv |  Drop |  Drop  |  Drop
 Interface Name              |  IF_ID    |  10s  |  1min |  5min |  10s  |  1min  |  5min
================================================================================================
 TenGigabitEthernet1/0/2       0x0000000a       5       5       5       0       0       0
 TenGigabitEthernet1/0/23      0x0000001f       1       1       1       0       0       0

------------------------------------------------------------------------------------------------
```

- Use "show platform software fed <switch> ative punt rates interfaces <IF-ID>" para detalhar e exibir as filas individuais da interface. Esse comando mostra estatísticas agregadas e pode ser usado para exibir a atividade da fila de entrada histórica e se o tráfego foi policiado.

<#root>

```
C9300#

show platform software fed switch active punt rates interfaces 0x1f <-- "0x1f" is the IF_ID of Te1/0/23,

Punt Rate on Single Interfaces Statistics

Interface : TenGigabitEthernet1/0/23 [if_id: 0x1F]

  Received                        Dropped
  --------                        -------
   Total          : 1010652          Total           : 0
```

```
   10 sec average : 1                10 sec average  : 0
    1 min average : 1                 1 min average  : 0
    5 min average : 1                 5 min average  : 0

Per CPUQ punt stats on the interface (rate averaged over 10s interval)
=============================================================================
 Q  |         Queue                  |  Recv   |  Recv  |  Drop  |  Drop  |
 no |         Name                   |  Total  |  Rate  |  Total |  Rate  |
=============================================================================
 0   CPU_Q_DOT1X_AUTH                      0        0        0        0
 1   CPU_Q_L2_CONTROL                   9109        0        0        0
 2   CPU_Q_FORUS_TRAFFIC              176659        0        0        0
 3   CPU_Q_ICMP_GEN                        0        0        0        0
 4   CPU_Q_ROUTING_CONTROL            447374        0        0        0
 5   CPU_Q_FORUS_ADDR_RESOLUTION       80693        0        0        0
 6   CPU_Q_ICMP_REDIRECT                   0        0        0        0
 7   CPU_Q_INTER_FED_TRAFFIC               0        0        0        0
 8   CPU_Q_L2LVX_CONTROL_PKT               0        0        0        0
 9   CPU_Q_EWLC_CONTROL                    0        0        0        0
 10  CPU_Q_EWLC_DATA                       0        0        0        0
 11  CPU_Q_L2LVX_DATA_PKT                  0        0        0        0
 12  CPU_Q_BROADCAST                   22680        0        0        0
 13  CPU_Q_CONTROLLER_PUNT                 0        0        0        0
 14  CPU_Q_SW_FORWARDING                   0        0        0        0
 15  CPU_Q_TOPOLOGY_CONTROL           271014        0        0        0
 16  CPU_Q_PROTO_SNOOPING                  0        0        0        0
 17  CPU_Q_DHCP_SNOOPING                   0        0        0        0
 18  CPU_Q_TRANSIT_TRAFFIC                 0        0        0        0
 19  CPU_Q_RPF_FAILED                      0        0        0        0
 20  CPU_Q_MCAST_END_STATION_SERVICE    2679        0        0        0
 21  CPU_Q_LOGGING                       444        0        0        0
 22  CPU_Q_PUNT_WEBAUTH                    0        0        0        0
 23  CPU_Q_HIGH_RATE_APP                   0        0        0        0
 24  CPU_Q_EXCEPTION                       0        0        0        0
 25  CPU_Q_SYSTEM_CRITICAL                 0        0        0        0
 26  CPU_Q_NFL_SAMPLED_DATA                0        0        0        0
 27  CPU_Q_LOW_LATENCY                     0        0        0        0
 28  CPU_Q_EGR_EXCEPTION                   0        0        0        0
 29  CPU_Q_FSS                             0        0        0        0
 30  CPU_Q_MCAST_DATA                      0        0        0        0
 31  CPU_Q_GOLD_PKT                        0        0        0        0

-----------------------------------------------------------------------------
```

Inspecionar o tráfego vinculado à CPU

A família de switches Catalyst 9000 oferece utilitários para monitorar e visualizar o tráfego vinculado à CPU. Use essas ferramentas para entender qual tráfego é apontado ativamente para a CPU.

Captura de pacotes incorporada (EPC)

O EPC no plano de controle pode ser feito em qualquer direção (ou em ambas). Para tráfego direcionado, capture a entrada. O EPC no plano de controle pode ser salvo em buffer ou em arquivo.

<#root>

```
C9300#

monitor capture CONTROL control-plane in match any buffer circular size 10


C9300#

show monitor capture CONTROL parameter <-- Check to ensure parameters are as expected.

   monitor capture CONTROL control-plane IN
   monitor capture CONTROL match any
   monitor capture CONTROL buffer size 10 circular
C9300#

monitor capture CONTROL start <-- Starts the capture.

Started capture point : CONTROL
C9300#

monitor capture CONTROL stop <-- Stops the capture.

Capture statistics collected at software:
    Capture duration - 5 seconds
    Packets received - 39
    Packets dropped - 0
    Packets oversized - 0

Bytes dropped in asic - 0

Capture buffer will exists till exported or cleared

Stopped capture point : CONTROL
```

Os resultados da captura podem ser exibidos em uma saída breve ou detalhada.

<#root>

```
C9300#

show monitor capture CONTROL buffer brief

Starting the packet display ........ Press Ctrl + Shift + 6 to exit

    1   0.000000 5c:5a:c7:61:4c:5f -> 00:00:04:00:0e:00 ARP 64 192.168.10.1 is at 5c:5a:c7:61:4c:5f
    2   0.030643 00:00:00:00:00:00 -> 00:06:df:f7:20:01 0x0000 30 Ethernet II
    3   0.200016 5c:5a:c7:61:4c:5f -> 00:00:04:00:0e:00 ARP 64 192.168.10.1 is at 5c:5a:c7:61:4c:5f
    4   0.400081 5c:5a:c7:61:4c:5f -> 00:00:04:00:0e:00 ARP 64 192.168.10.1 is at 5c:5a:c7:61:4c:5f
    5   0.599962 5c:5a:c7:61:4c:5f -> 00:00:04:00:0e:00 ARP 64 192.168.10.1 is at 5c:5a:c7:61:4c:5f
    6   0.800067 5c:5a:c7:61:4c:5f -> 00:00:04:00:0e:00 ARP 64 192.168.10.1 is at 5c:5a:c7:61:4c:5f
    7   0.812456 00:1b:0d:a5:e2:a5 -> 01:80:c2:00:00:00 STP 60 RST. Root = 0/10/00:1b:53:bb:91:00  Cost
    8   0.829809 10.122.163.3 -> 224.0.0.2    HSRP 92 Hello (state Active)
    9   0.981313 10.122.163.2 -> 224.0.0.13   PIMv2 72 Hello
   10   1.004747 5c:5a:c7:61:4c:5f -> 00:00:04:00:0e:00 ARP 64 192.168.10.1 is at 5c:5a:c7:61:4c:5f
   11   1.200082 5c:5a:c7:61:4c:5f -> 00:00:04:00:0e:00 ARP 64 192.168.10.1 is at 5c:5a:c7:61:4c:5f
   12   1.399987 5c:5a:c7:61:4c:5f -> 00:00:04:00:0e:00 ARP 64 192.168.10.1 is at 5c:5a:c7:61:4c:5f
   13   1.599944 5c:5a:c7:61:4c:5f -> 00:00:04:00:0e:00 ARP 64 192.168.10.1 is at 5c:5a:c7:61:4c:5f
<snip>

C9300#
```

show monitor capture CONTROL buffer detail | begin Frame 7

```
Frame 7: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface /tmp/epc_ws/wif_to_ts_p
    Interface id: 0 (/tmp/epc_ws/wif_to_ts_pipe)
        Interface name: /tmp/epc_ws/wif_to_ts_pipe
    Encapsulation type: Ethernet (1)
    Arrival Time: May  3, 2023 23:58:11.727432000 UTC
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1683158291.727432000 seconds
    [Time delta from previous captured frame: 0.012389000 seconds]
    [Time delta from previous displayed frame: 0.012389000 seconds]
    [Time since reference or first frame: 0.812456000 seconds]
    Frame Number: 7
    Frame Length: 60 bytes (480 bits)
    Capture Length: 60 bytes (480 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:llc:stp]
IEEE 802.3 Ethernet
    Destination: 01:80:c2:00:00:00 (01:80:c2:00:00:00)
        Address: 01:80:c2:00:00:00 (01:80:c2:00:00:00)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    Source: 00:1b:0d:a5:e2:a5 (00:1b:0d:a5:e2:a5)
        Address: 00:1b:0d:a5:e2:a5 (00:1b:0d:a5:e2:a5)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Length: 39
    Padding: 00000000000000
Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
        0100 001. = SAP: Spanning Tree BPDU
        .... ...0 = IG Bit: Individual
    SSAP: Spanning Tree BPDU (0x42)
        0100 001. = SAP: Spanning Tree BPDU
        .... ...0 = CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x3)
Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Rapid Spanning Tree (2)
    BPDU Type: Rapid/Multiple Spanning Tree (0x02)
    BPDU flags: 0x3c, Forwarding, Learning, Port Role: Designated
        0... .... = Topology Change Acknowledgment: No
        .0.. .... = Agreement: No
        ..1. .... = Forwarding: Yes
        ...1 .... = Learning: Yes
        .... 11.. = Port Role: Designated (3)
        .... ..0. = Proposal: No
        .... ...0 = Topology Change: No
    Root Identifier: 0 / 10 / 00:1b:53:bb:91:00
        Root Bridge Priority: 0
        Root Bridge System ID Extension: 10
        Root Bridge System ID: 00:1b:53:bb:91:00 (00:1b:53:bb:91:00)
    Root Path Cost: 19
    Bridge Identifier: 32768 / 10 / 00:1b:0d:a5:e2:80
        Bridge Priority: 32768
        Bridge System ID Extension: 10
        Bridge System ID: 00:1b:0d:a5:e2:80 (00:1b:0d:a5:e2:80)
    Port identifier: 0x8025
    Message Age: 1
    Max Age: 20
```

```
    Hello Time: 2
    Forward Delay: 15
    Version 1 Length: 0

C9300#

monitor capture CONTROL buffer display-filter "frame.number==9" detailed <-- Most Wireshark display filt

Starting the packet display ........ Press Ctrl + Shift + 6 to exit

Frame 9: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface /tmp/epc_ws/wif_to_ts_p
    Interface id: 0 (/tmp/epc_ws/wif_to_ts_pipe)
        Interface name: /tmp/epc_ws/wif_to_ts_pipe
    Encapsulation type: Ethernet (1)
    Arrival Time: May  4, 2023 00:07:44.912567000 UTC
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1683158864.912567000 seconds
    [Time delta from previous captured frame: 0.123942000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 1.399996000 seconds]
    Frame Number: 9
    Frame Length: 64 bytes (512 bits)
    Capture Length: 64 bytes (512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:vlan:ethertype:arp]
Ethernet II, Src: 5c:5a:c7:61:4c:5f (5c:5a:c7:61:4c:5f), Dst: 00:00:04:00:0e:00 (00:00:04:00:0e:00)
    Destination: 00:00:04:00:0e:00 (00:00:04:00:0e:00)
        Address: 00:00:04:00:0e:00 (00:00:04:00:0e:00)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: 5c:5a:c7:61:4c:5f (5c:5a:c7:61:4c:5f)
        Address: 5c:5a:c7:61:4c:5f (5c:5a:c7:61:4c:5f)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 10
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0000 1010 = ID: 10
    Type: ARP (0x0806)
    Padding: 00000000000000000000000000000000
    Trailer: 00000000
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 5c:5a:c7:61:4c:5f (5c:5a:c7:61:4c:5f)
    Sender IP address: 192.168.10.1
    Target MAC address: 00:00:04:00:0e:00 (00:00:04:00:0e:00)
    Target IP address: 192.168.10.25
```

Os resultados da captura podem ser gravados diretamente no arquivo ou exportados do buffer.

<#root>

```
C9300#

monitor capture CONTROL export location flash:control.pcap <-- Exports the current buffer to file. Exter

Export Started Successfully

Export completed for capture point CONTROL
C9300#
C9300#

dir flash: | in control.pcap

475231  -rw-             3972   May 4 2023 00:00:38 +00:00  control.pcap
C9300#
```

Captura de Pacotes de CPU FED

A família de switches Catalyst 9000 suporta um utilitário de depuração que permite visibilidade aprimorada de pacotes de e para a CPU.

```
C9300#debug platform software fed switch active punt packet-capture ?
  buffer        Configure packet capture buffer
  clear-filter  Clear punt PCAP filter
  set-filter    Specify wireshark like filter (Punt PCAP)
  start         Start punt packet capturing
  stop          Stop punt packet capturing

C9300#$re fed switch active punt packet-capture buffer limit 16384
Punt PCAP buffer configure: one-time with buffer size 16384...done

C9300#show platform software fed switch active punt packet-capture status
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 0 packets. Capture capacity : 16384 packets

C9300#debug platform software fed switch active punt packet-capture start
Punt packet capturing started.

C9300#debug platform software fed switch active punt packet-capture stop
Punt packet capturing stopped. Captured 55 packet(s)
```

O conteúdo do buffer tem opções breves e detalhadas para a saída.

<#root>

```
C9300#

show platform software fed switch active punt packet-capture brief

Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 55 packets. Capture capacity : 16384 packets

------ Punt Packet Number: 1, Timestamp: 2023/05/04 00:17:41.709 ------
 interface : physical: TenGigabitEthernet1/0/2[if-id: 0x0000000a], pal: TenGigabitEthernet1/0/2 [if-id:
 metadata  : cause: 109 [snoop packets], sub-cause: 1, q-no: 16, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 0000.0400.0e00, src mac: 5c5a.c761.4c5f
```

```
 ether hdr : vlan: 10, ethertype: 0x8100

------ Punt Packet Number: 2, Timestamp: 2023/05/04 00:17:41.909 ------
 interface : physical: TenGigabitEthernet1/0/2[if-id: 0x0000000a], pal: TenGigabitEthernet1/0/2 [if-id:
 metadata  : cause: 109 [snoop packets], sub-cause: 1, q-no: 16, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 0000.0400.0e00, src mac: 5c5a.c761.4c5f
 ether hdr : vlan: 10, ethertype: 0x8100

------ Punt Packet Number: 3, Timestamp: 2023/05/04 00:17:42.109 ------
 interface : physical: TenGigabitEthernet1/0/2[if-id: 0x0000000a], pal: TenGigabitEthernet1/0/2 [if-id:
 metadata  : cause: 109 [snoop packets], sub-cause: 1, q-no: 16, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 0000.0400.0e00, src mac: 5c5a.c761.4c5f
 ether hdr : vlan: 10, ethertype: 0x8100

------ Punt Packet Number: 4, Timestamp: 2023/05/04 00:17:42.309 ------
 interface : physical: TenGigabitEthernet1/0/2[if-id: 0x0000000a], pal: TenGigabitEthernet1/0/2 [if-id:
 metadata  : cause: 109 [snoop packets], sub-cause: 1, q-no: 16, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 0000.0400.0e00, src mac: 5c5a.c761.4c5f
 ether hdr : vlan: 10, ethertype: 0x8100

------ Punt Packet Number: 5, Timestamp: 2023/05/04 00:17:42.509 ------
 interface : physical: TenGigabitEthernet1/0/2[if-id: 0x0000000a], pal: TenGigabitEthernet1/0/2 [if-id:
 metadata  : cause: 109 [snoop packets], sub-cause: 1, q-no: 16, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 0000.0400.0e00, src mac: 5c5a.c761.4c5f
 ether hdr : vlan: 10, ethertype: 0x8100

C9300#
```

show platform software fed switch active punt packet-capture detailed <-- Detailed provides the same inf

```
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 55 packets. Capture capacity : 16384 packets

------ Punt Packet Number: 1, Timestamp: 2023/05/04 00:17:41.709 ------
 interface : physical: TenGigabitEthernet1/0/2[if-id: 0x0000000a], pal: TenGigabitEthernet1/0/2 [if-id:
 metadata  : cause: 109 [snoop packets], sub-cause: 1, q-no: 16, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 0000.0400.0e00, src mac: 5c5a.c761.4c5f
 ether hdr : vlan: 10, ethertype: 0x8100

 Packet Data Hex-Dump (length: 68 bytes) :
   000004000E005C5A  C7614C5F8100000A  0806000108000604  00025C5AC7614C5F
   C0A80A0100000400  0E00C0A80A190000  0000000000000000  0000000000000000
   E9F1C9F3

 Doppler Frame Descriptor :
   fdFormat                  = 0x4           systemTtl                 = 0xe
   loadBalHash1              = 0x20          loadBalHash2              = 0xc
   spanSessionMap            = 0             forwardingMode            = 0
   destModIndex              = 0             skipIdIndex               = 0
   srcGpn                    = 0x2           qosLabel                  = 0x83
   srcCos                    = 0             ingressTranslatedVlan     = 0x7
   bpdu                      = 0             spanHistory               = 0
   sgt                       = 0             fpeFirstHeaderType        = 0
   srcVlan                   = 0xa           rcpServiceId              = 0x1
   wccpSkip                  = 0             srcPortLeIndex            = 0x1
   cryptoProtocol            = 0             debugTagId                = 0
   vrfId                     = 0             saIndex                   = 0
   pendingAfdLabel           = 0             destClient                = 0x1
   appId                     = 0             finalStationIndex         = 0x74
   decryptSuccess            = 0             encryptSuccess            = 0
   rcpMiscResults            = 0             stackedFdPresent          = 0
   spanDirection             = 0             egressRedirect            = 0
   redirectIndex             = 0             exceptionLabel            = 0
```

```
   destGpn                   = 0             inlineFd                  = 0x1
   suppressRefPtrUpdate      = 0             suppressRewriteSideEfects = 0
   cmi2                      = 0             currentRi                 = 0x1
   currentDi                 = 0x527b        dropIpUnreachable         = 0
   srcZoneId                 = 0             srcAsicId                 = 0
   originalDi                = 0             originalRi                = 0
   srcL3IfIndex              = 0x27          dstL3IfIndex              = 0
   dstVlan                   = 0             frameLength               = 0x44
   fdCrc                     = 0x97          tunnelSpokeId             = 0
   isPtp                     = 0             ieee1588TimeStampValid    = 0
   ieee1588TimeStamp55_48    = 0             lvxSourceRlocIpAddress    = 0
   sgtCachingNeeded          = 0

 Doppler Frame Descriptor Hex-Dump :
   0000000044004E04  000B40977B520000  0000000000000100  000000070A000000
   0000000001000010  0000000074000100  0000000027830200  0000000000000000
```

Muitos filtros de exibição estão disponíveis para uso. Há suporte para os filtros de exibição mais comuns do Wireshark.

<#root>

```
C9300#

show platform software fed switch active punt packet-capture display-filter-help

FED Punject specific filters :
   1. fed.cause              FED punt or inject cause
   2. fed.linktype           FED linktype
   3. fed.pal_if_id          FED platform interface ID
   4. fed.phy_if_id          FED physical interface ID
   5. fed.queue              FED Doppler hardware queue
   6. fed.subcause           FED punt or inject sub cause
Generic filters supported :
   7. arp                    Is this an ARP packet
   8. bootp                  DHCP packets [Macro]
   9. cdp                    Is this a CDP packet
  10. eth                    Does the packet have an Ethernet header
  11. eth.addr               Ethernet source or destination MAC address
  12. eth.dst                Ethernet destination MAC address
  13. eth.ig                 IG bit of ethernet destination address (broadcast/multicast)
  14. eth.src                Ethernet source MAC address
  15. eth.type               Ethernet type
  16. gre                    Is this a GRE packet
  17. icmp                   Is this a ICMP packet
  18. icmp.code              ICMP code
  19. icmp.type              ICMP type
  20. icmpv6                 Is this a ICMPv6 packet
  21. icmpv6.code            ICMPv6 code
  22. icmpv6.type            ICMPv6 type
  23. ip                     Does the packet have an IPv4 header
  24. ip.addr                IPv4 source or destination IP address
  25. ip.dst                 IPv4 destination IP address
  26. ip.flags.df            IPv4 dont fragment flag
  27. ip.flags.mf            IPv4 more fragments flag
  28. ip.frag_offset         IPv4 fragment offset
  29. ip.proto               Protocol used in datagram
  30. ip.src                 IPv4 source IP address
```

```
  31. ip.ttl                IPv4 time to live
  32. ipv6                  Does the packet have an IPv4 header
  33. ipv6.addr             IPv6 source or destination IP address
  34. ipv6.dst              IPv6 destination IP address
  35. ipv6.hlim             IPv6 hop limit
  36. ipv6.nxt              IPv6 next header
  37. ipv6.plen             IPv6 payload length
  38. ipv6.src              IPv6 source IP address
  39. stp                   Is this a STP packet
  40. tcp                   Does the packet have a TCP header
  41. tcp.dstport           TCP destination port
  42. tcp.port              TCP source OR destination port
  43. tcp.srcport           TCP source port
  44. udp                   Does the packet have a UDP header
  45. udp.dstport           UDP destination port
  46. udp.port              UDP source OR destination port
  47. udp.srcport           UDP source port
  48. vlan.id               Vlan ID (dot1q or qinq only)
  49. vxlan                 Is this a VXLAN packet


C9300#
```

```
show platform software fed switch active punt packet-capture display-filter arp brief

Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 55 packets. Capture capacity : 16384 packets

------ Punt Packet Number: 1, Timestamp: 2023/05/04 00:17:41.709 ------
 interface : physical: TenGigabitEthernet1/0/2[if-id: 0x0000000a], pal: TenGigabitEthernet1/0/2 [if-id:
 metadata  : cause: 109 [snoop packets], sub-cause: 1, q-no: 16, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 0000.0400.0e00, src mac: 5c5a.c761.4c5f
 ether hdr : vlan: 10, ethertype: 0x8100

------ Punt Packet Number: 2, Timestamp: 2023/05/04 00:17:41.909 ------
 interface : physical: TenGigabitEthernet1/0/2[if-id: 0x0000000a], pal: TenGigabitEthernet1/0/2 [if-id:
 metadata  : cause: 109 [snoop packets], sub-cause: 1, q-no: 16, linktype: MCP_LINK_TYPE_IP [1]
 ether hdr : dest mac: 0000.0400.0e00, src mac: 5c5a.c761.4c5f
 ether hdr : vlan: 10, ethertype: 0x8100
<snip>
```

Os filtros também podem ser aplicados como filtros de captura.

<#root>

```
C9300#

show platform software fed switch active punt packet-capture set-filter arp <-- Most common Wireshark fi

Filter setup successful. Captured packets will be cleared

C9300#$e fed switch active punt packet-capture status
Punt packet capturing: disabled. Buffer wrapping: disabled
Total captured so far: 0 packets. Capture capacity : 16384 packets
Capture filter : "arp"
```

Cenários comuns

Perda intermitente de ICMP (ping) para o IP local

O tráfego encaminhado para um IP local em um switch é colocado na fila Forus (literalmente "para nós"). Ver incrementos na fila de CoPP de fóruns relaciona-se a pacotes descartados destinados ao switch local. Isso é relativamente direto e fácil de conceitualizar.

Em algumas condições, no entanto, pode haver perda para o tráfego destinado localmente que não se correlaciona perfeitamente com quedas de Forus.

Com fluxo de tráfego limitado pela CPU suficiente, o caminho de punt fica saturado além da capacidade de CoPP para priorizar qual tráfego é policiado. O tráfego é policiado "silenciosamente", primeiro a entrar, primeiro a sair.

Neste cenário, é vista evidência de policiamento de plano de controle em alto volume, mas o tipo de tráfego de interesse (Forus neste exemplo) não aumenta necessariamente.

Em resumo, se houver um volume excepcionalmente alto de tráfego vinculado à CPU, evidenciado por policiamento de CoPP ativo e demonstrado com uma captura de pacotes ou depuração de punt de FED, poderá haver perda que não se alinhe à fila que você está solucionando. Neste cenário, determine por que há uma quantidade excessiva de tráfego vinculado à CPU e tome medidas para aliviar a carga no plano de controle.

Redirecionamentos ICMP altos e operação DHCP lenta

O CoPP no switch da série Catalyst 9000 é organizado em 32 filas de hardware. Essas 32 filas de hardware se alinham a 20 índices de vigilante individuais. Cada índice de vigilante correlaciona-se com uma ou mais filas de hardware.

Funcionalmente, isso significa que várias classes de tráfego compartilham um índice de vigilante e estão sujeitas a um valor de vigilante agregado comum.

Um problema comum visto em switches com agentes de retransmissão DHCP ativados envolve uma resposta DHCP lenta. Os clientes podem obter IPs esporadicamente, mas são necessárias várias tentativas para concluir e alguns clientes atingem o tempo limite.

A fila de redirecionamento de ICMP e a fila de Broadcast compartilham um índice de vigilante, de modo que um alto volume de tráfego que é recebido e roteado para fora da mesma Interface Virtual do Switch (SVI) impacta os aplicativos que dependem do tráfego de broadcast. Isso é especialmente perceptível quando o switch atua como um agente de retransmissão.

Este documento oferece uma explicação detalhada do conceito e como mitigar: Troubleshooting de Problemas de DHCP nos Catalyst 9000 DHCP Relay Agents

Outros recursos

Identificar e Solucionar Problemas de DHCP Lento ou Intermitente em Agentes de Retransmissão DHCP do Catalyst 9000

Configurar a captura de pacotes de CPU FED nos Switches Catalyst 9000

Switches Catalyst 9300: configuração da vigilância do plano de controle

Configurando a captura de pacotes - Guia de configuração de gerenciamento de rede, Cisco IOS XE Bengaluru 17.6.x (Switches Catalyst 9300)

Operar e solucionar problemas de rastreamento de DHCP em switches Catalyst 9000

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.