Resolución de problemas de operaciones del plano de control en switches Catalyst 9000

Contenido

Introducción

Este documento describe cómo resolver problemas y validar el estado del plano de control en los switches de la familia Catalyst 9000 que ejecutan Cisco IOS® XE.

Antecedentes

La tarea principal de un switch es reenviar los paquetes lo más rápido posible. La mayoría de los paquetes se reenvían en hardware, pero ciertos tipos de tráfico deben ser manejados por la CPU del sistema. El tráfico que llega a la CPU se maneja lo más rápido posible. Se espera que se observe cierta cantidad de tráfico en la CPU, pero una sobreabundancia conduce a problemas operativos. La familia de switches Catalyst 9000 incorpora un sólido mecanismo de regulación del plano de control (CoPP) de forma predeterminada para evitar los problemas causados por la saturación del tráfico de la CPU.

En algunos casos prácticos surgen problemas inesperados en función del funcionamiento normal. La correlación entre causa y efecto no es obvia a veces, lo que hace que el problema sea difícil

de abordar. Este documento le proporciona herramientas para validar el estado del plano de control y proporciona un flujo de trabajo sobre cómo abordar los problemas que involucran el punt del plano de control o el trayecto de inyección. También proporciona varias situaciones comunes basadas en problemas observados sobre el terreno.

Tenga en cuenta que el trayecto de punt de la CPU es un recurso limitado. Los switches de reenvío de hardware modernos pueden gestionar un volumen de tráfico exponencialmente mayor. La familia de switches Catalyst 9000 admite aproximadamente 19 000 paquetes por segundo (pps) en conjunto en la CPU en un momento dado. Si se supera este umbral, el tráfico punteado se controla sin peso.

Terminology

- Controlador de motor de reenvío (FED): se trata del núcleo del switch Catalyst de Cisco y es responsable de toda la programación y retransmisión del hardware
- IOSd: Este es el daemon de Cisco IOS que se ejecuta en el kernel de Linux. Se ejecuta como un proceso de software dentro del núcleo
- Sistema de entrega de paquetes (PDS): arquitectura y proceso de cómo se entregan los paquetes desde y hacia los diversos subsistemas. Como ejemplo, controla cómo se entregan los paquetes desde la FED al IOSd y viceversa
- Plano de control (CP):El plano de control es un término genérico que se utiliza para agrupar las funciones y el tráfico que afectan a la CPU del switch Catalyst. Esto incluye el tráfico como el protocolo de árbol de extensión (STP), el protocolo de router en espera en caliente (HSRP) y los protocolos de routing destinados al switch o enviados desde el switch. Esto también incluye protocolos de capa de aplicación como Secure Shell (SSH) y protocolo simple de administración de red (SNMP) que debe gestionar la CPU
- Plano de datos (DP):normalmente el plano de datos incluye los ASIC de hardware y el tráfico que se reenvía sin la ayuda del plano de control
- Punt:Paquete de control de protocolo de entrada que se interceptó en el DP enviado al CP para procesarlo
- Inyección:paquete de protocolo generado por CP enviado a DP para salir de las interfaces de E/S
- LSMPI:Interfaz de Punt de Memoria Compartida de Linux

Catalyst 9000 CoPP

La base de la protección de la CPU en la familia de switches Catalyst 9000 es CoPP. Con CoPP, se aplica una política de calidad de servicio (QoS) generada por el sistema en la ruta de inserción/punt de la CPU. El tráfico enlazado a la CPU se agrupa en muchas clases diferentes y, posteriormente, se asigna a través de los controladores de políticas de hardware individuales asociados a la CPU. Los reguladores de tráfico evitan la saturación excesiva de la CPU por una clase de tráfico determinada.

Implementación de CoPP

El tráfico enlazado a la CPU se clasifica en colas. Estas colas/clases están definidas por el

sistema y no son configurables por el usuario. Los reguladores se configuran en el hardware. La familia Catalyst 9000 admite 32 controladores de políticas de hardware para 32 colas.

Los valores específicos difieren de una plataforma a otra. En general, hay 32 colas definidas por el sistema. Estas colas se relacionan con los class-maps, que se relacionan con los índices del regulador. Los índices del regulador tienen una velocidad predeterminada del regulador. Esta velocidad la puede configurar el usuario, aunque los cambios en la política CoPP predeterminada aumentan la susceptibilidad a un impacto inesperado en el servicio.

Valores definidos por el sistema para CoPP

Nombres de mapas de clase	Policer Index (Policer No.)	Colas de CPU (Nº de cola)
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-topology-control	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTROL(15)

Nombres de mapas de clase	Policer Index (Policer No.)	Colas de CPU (Nº de cola)
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)
system-cpp-police-multicast-	WK_CPP_POLICE_MULTICAST_SNOOPING(15)	WK_CPU_Q_MCAST_END_STA_TION_SE

Nombres de mapas de clase	Policer Index (Policer No.)	Colas de CPU (Nº de cola)
end-station		
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
system-cpp-police-l2lvx-control	WK_CPP_L2_LVX_CONT_PACK(4)	WK_CPU_Q_L2_LVX_CONT_PACK(8)

Cada cola está relacionada con un tipo de tráfico o un conjunto de funciones determinado. Esta es una lista exhaustiva:

Colas de CPU y funciones asociadas

Colas de CPU (Nº de cola)	Característica(s)
WK_CPU_Q_DOT1X_AUTH(0)	Autenticación basada en puertos IEEE 802.1x
WK_CPU_Q_L2_CONTROL(1)	Protocolo de concentración de enlaces dinámico (DTP) VLAN Trunking Protocol (VTP) Port Aggregation Protocol (PAgP) Protocolo de señalización de información del cliente (CISP)

Colas de CPU (Nº de cola)	Característica(s)
	Protocolo de retransmisión de sesión de mensajes Protocolo de registro de VLAN múltiple (MVRP) Red móvil metropolitana (MMN) Protocolo de descubrimiento de nivel de enlace (LLDP) Unidirectional Link Detection (UDLD) Protocolo de control de agregación de enlaces (LACP) Protocolo de detección de Cisco (CDP) Spanning Tree Protocol (STP)
WK_CPU_Q_FORUS_TRAFFIC(2)	Host como Telnet, Pingv4 y Pingv6, y SNMP Detección de keepalive/loopback Protocolo de intercambio de claves por Internet (IKE) (IPSec)
WK_CPU_Q_ICMP_GEN(3)	ICMP: destino inalcanzable ICMP-TTL caducado
WK_CPU_Q_ROUTING_CONTROL(4)	Protocolo de información de enrutamiento versión 1 (RIPv1) RIPv2 Protocolo de routing de gateway interior (IGRP) Border Gateway Protocol (BGP) PIM-UDP Virtual Router Redundancy Protocol

Colas de CPU (Nº de cola)	Característica(s)
	(VRRP) Protocolo de router con espera en caliente versión 1 (HSRPv1) HSRPv2 Gateway Load Balancing Protocol (GLBP) Label Distribution Protocol (LDP) Protocolo de comunicación de caché web (WCCP) Protocolo de información de routing de última generación (RIPng) Abrir primero la ruta más corta (OSPF) Abrir primero la ruta de acceso más corta versión 3 (OSPFv3) Protocolo de routing de gateway interior mejorado (EIGRP) Protocolo de routing de gateway interior mejorado versión 6 (EIGRPv6) DHCPv6 Multidifusión independiente de protocolo (PIM) Multidifusión independiente del protocolo versión 6 (PIMv6) Protocolo de router con espera en caliente de última generación (HSRPng) control IPv6 keepalive de encapsulación de routing genérico (GRE) Punt de traducción de direcciones de red (NAT) Intermediate System-to-Intermediate

Colas de CPU (Nº de cola)	Característica(s)
	System (IS-IS)
WK_CPU_Q_FORUS_ADDR_RESOLUTION(5)	Protocolo de resolución de direcciones (ARP) Anuncio de vecino IPv6 y solicitud de vecino
WK_CPU_Q_ICMP_REDIRECT(6)	Redirección del protocolo de mensajes de control de Internet (ICMP)
WK_CPU_Q_INTER_FED_TRAFFIC(7)	Inserción de dominio de puente de capa 2 para comunicación interna.
WK_CPU_Q_L2_LVX_CONT_PACK(8)	Paquete de ID de Exchange (XID)
WK_CPU_Q_EWLC_CONTROL(9)	Controlador inalámbrico integrado (eWLC) [control y aprovisionamiento de puntos de acceso inalámbricos (CAPWAP) (UDP 5246)]
WK_CPU_Q_EWLC_DATA(10)	Paquete de datos eWLC (DATOS CAPWAP, UDP 5247)
WK_CPU_Q_L2_LVX_DATA_PACK(11)	Paquete de unidifusión desconocido impulsado para la solicitud de mapa.
WK_CPU_Q_BROADCAST(12)	Todos los tipos de difusión
WK_CPU_Q_OPENFLOW(13)	Desbordamiento de caché de aprendizaje (capa 2 + capa 3)
WK_CPU_Q_CONTROLLER_PUNT(14)	Datos: lista de control de acceso (ACL) completa Datos: opciones de IPv4 Datos: salto por salto IPv6

Colas de CPU (Nº de cola)	Característica(s)
	Datos: falta de recursos/captura de todos Datos - Reenvío de ruta inversa (RPF) incompleto Recolectar paquete
WK_CPU_Q_TOPOLOGY_CONTROL(15)	Spanning Tree Protocol (STP) Resilient Ethernet Protocol (REP) Protocolo de árbol de extensión compartido (SSTP)
WK_CPU_Q_PROTO_SNOOPING(16)	Detección de protocolo de resolución de direcciones (ARP) para Dynamic ARP Inspection (DAI)
WK_CPU_Q_DHCP_SNOOPING(17)	Snooping DHCP
WK_CPU_Q_TRANSIT_TRAFFIC(18)	Esto se utiliza para los paquetes impulsados por NAT, que deben manejarse en la trayectoria del software.
WK_CPU_Q_RPF_FAILED(19)	Datos - error de mRPF (RPF multidifusión)
WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	Control de protocolo de administración de grupos de Internet (IGMP)/detección de escucha de multidifusión (MLD)
WK_CPU_Q_LOGGING(21)	Registro de lista de control de acceso (ACL)
WK_CPU_Q_PUNT_WEBAUTH(22)	Autenticación Web
WK_CPU_Q_HIGH_RATE_APP(23)	Difusión
WK_CPU_Q_EXCEPTION(24)	indicación IKE

Colas de CPU (Nº de cola)	Característica(s)
	Infracción de aprendizaje de IP Infracción de seguridad del puerto IP Infracción de dirección estática de IP Comprobación del alcance de IPv6 Excepción del protocolo de copia remota (RCP) Error de RPF unidifusión
WK_CPU_Q_SYSTEM_CRITICAL(25)	Señalización de medios/ARP de proxy inalámbrico
WK_CPU_Q_NFL_SAMPLED_DATA(26)	Datos de ejemplo de Netflow y Media Services Proxy (MSP)
WK_CPU_Q_LOW_LATENCY(27)	Detección de reenvío bidireccional (BFD), protocolo de tiempo de precisión (PTP)
WK_CPU_Q_EGR_EXCEPTION(28)	Excepción de resolución de salida
WK_CPU_Q_STACKWISE_VIRTUAL_CONTROL(29)	Protocolos de apilamiento lateral frontal, concretamente SVL
WK_CPU_Q_MCAST_DATA(30)	Datos: creación (S,G) Datos: uniones locales Datos - Registro PIM Datos - Conmutación SPT Datos - Multidifusión
WK_CPU_Q_GOLD_PKT(31)	Oro

Política predeterminada

De forma predeterminada, la política CoPP generada por el sistema se aplica a la ruta de punt/inyección. La política predeterminada se puede ver mediante comandos comunes basados en MQC. También se puede ver en la configuración del switch. La única política que se permite aplicar en el ingreso o egreso de la CPU/plano de control es la política definida por el sistema.

Utilice "show policy-map control-plane" para ver la política aplicada al plano de control:

```
<#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

Las velocidades del regulador de CoPP pueden ser configuradas por el usuario. Los usuarios también tienen la capacidad de deshabilitar colas.

En este ejemplo se muestra cómo ajustar un valor de regulador individual. En este ejemplo, la clase ajustada es "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
```

En este ejemplo se muestra cómo deshabilitar una cola por completo. Tenga cuidado al desactivar las colas, ya que esto podría provocar una posible saturación excesiva de la 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
```

Troubleshoot

Metodología

La utilización de la CPU se ve afectada por dos actividades básicas: procesos e interrupciones. Los procesos son actividades estructuradas que realiza la CPU mientras que la interrupción se refiere a paquetes interceptados en el plano de datos y enviados a la CPU para la acción. En conjunto, estas actividades abarcan la utilización total de la CPU. Dado que CoPP está habilitado de forma predeterminada, el impacto de un servicio no se correlaciona necesariamente con una alta utilización de la CPU. Si la CoPP realiza su trabajo, la utilización de la CPU no se ve afectada en gran medida. Es importante tener en cuenta el uso general de la CPU, pero el uso general no

cuenta toda la historia. Los comandos show y las utilidades de esta sección se utilizan para evaluar rápidamente el estado de la CPU e identificar detalles relevantes sobre el tráfico destinado a la CPU.

Pautas:

- Determine si el problema está relacionado con el plano de control. La mayor parte del tráfico de tránsito se reenvía en hardware. Solo ciertos tipos de tráfico y ciertos escenarios involucran la CPU y el plano de control, así que tenga esto en cuenta durante toda la investigación.
- Comprender la base de utilización. Es importante comprender el aspecto de la utilización normal para poder identificar las desviaciones con respecto a la norma.
- Valide la utilización general tanto de los procesos como de las interrupciones. Identifique cualquier proceso que ocupe volúmenes inesperados de ciclos de CPU. Si la utilización se sitúa fuera del intervalo esperado, esto puede ser motivo de preocupación. Es importante comprender la utilización media de un sistema, de modo que se reconozcan las desviaciones fuera de la norma. Tenga en cuenta que la utilización por sí sola no es una imagen completa del estado del plano de control.
- Determine si hay caídas que se incrementan de forma activa en CoPP. Las caídas de CoPP no siempre son indicativas de un problema, pero si resuelve un problema relacionado con una clase de tráfico que se controla de forma activa, esto es un fuerte indicador de relevancia.

‘Comandos show útiles’

El switch permite una supervisión rápida del estado de la CPU y las estadísticas de CoPP. También existe una CLI útil para determinar rápidamente el punto de entrada del tráfico enlazado a la CPU.

Determinación de la utilización general e histórica

- "Mostrar procesos de cpu ordenados" se utiliza para ver el uso general de la CPU. El argumento "sorted" ordena la salida del proceso en función del porcentaje de uso. Los procesos que utilizan más recursos de CPU se encuentran en la parte superior del resultado. La utilización debida a interrupciones también se proporciona como porcentaje.

<#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" proporciona un gráfico histórico del uso de la CPU durante los últimos 60 segundos, 5 minutos y 72 horas.

<#root>

```
Catalyst-9600#
show processes cpu history

    9997777766666888886666677777777778888877777666669999998888866
<<<--- The numbers at the top of each column represent the highest value seen throughout the time perio

    2225555599999444444444000008888888881111177777333335555500
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%
```

```
    9998989899998989989989989889999989889898899999999899999999
    4318230911026353162351292837713365748928096040142309011335 11
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
    6656565666465556666556565756545565677375555675745455457759575546485767 57
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%
```

Comprobación de políticas del plano de control

- Utilice "show platform hardware fed <switch> active qos queue stats internal cpu policer" para ver las estadísticas de CoPP agregadas y la información adicional sobre la estructura de cola/policer. Esta salida proporciona una vista histórica de las estadísticas del regulador desde el último reinicio del plano de control. Estos contadores también se pueden borrar manualmente. Generalmente, la evidencia de caídas del plano de control por parte del regulador apunta a un problema con la cola/clase asociada, pero asegúrese de que las caídas se incrementan activamente mientras ocurre el problema. Ejecute el comando varias veces para observar el aumento de los valores de descarte de cola.

<#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

level-1 level-2 level-2

QId	PlcIdx	PlcIdx	Queue Name	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/

Recopilar información sobre el tráfico impulsado

Estos comandos se utilizan para recopilar información sobre el tráfico dirigido a la CPU, incluido el tipo de tráfico y los puntos físicos de entrada.

- "Show platform software fed <switch> active punt cpuq all" o "Show platform software fed <switch> active punt cpuq <0-31 Queue ID>" se pueden utilizar para ver estadísticas relacionadas con todo o con una cola 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
-----------------------------------------
```

- Utilice "show platform software fed <switch> active punt cause summary" para obtener una visión rápida de todos los diferentes tipos de tráfico que se han visto en la CPU. Tenga en cuenta que sólo se muestran las causas distintas de cero.

<#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
-------------------------------------------------------------------------------
```

- Utilice el comando "show platform software fed <switch> active punt rates interfaces" para ver rápidamente las interfaces que el tráfico dirigido a la CPU ingresa al sistema. Este comando sólo muestra interfaces con una cola de entrada distinta de cero.

<#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

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

- Utilice "show platform software fed <switch> active punt rates interfaces <IF-ID>" para obtener detalles y ver las colas individuales de la interfaz. Este comando muestra estadísticas agregadas y se puede utilizar para ver la actividad histórica de la cola de

entrada y si se ha controlado el tráfico.

<#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 no	Queue Name	Recv Total	Recv Rate	Drop Total	Drop 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

Inspeccionar tráfico enlazado a CPU

La familia de switches Catalyst 9000 ofrece utilidades para supervisar y ver el tráfico dirigido a la CPU. Utilice estas herramientas para comprender qué tráfico se dirige activamente a la CPU.

Captura de paquetes integrada (EPC)

El EPC en el plano de control se puede realizar en cualquier dirección (o en ambas). Para el tráfico impulsado, capture el tráfico entrante. El EPC en el plano de control se puede guardar en el búfer o en un archivo.

<#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
```

Los resultados de la captura se pueden ver en resultados breves o detallados.

<#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
```

Los resultados de la captura se pueden escribir directamente en el archivo o exportarse desde el búfer.

```
<#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 paquetes de CPU FED

La familia de switches Catalyst 9000 admite una utilidad de depuración que permite una mejor visibilidad de los paquetes hacia y desde la 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)
```

El contenido del búfer tiene opciones breves y detalladas para la salida.

<#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
```

Hay muchos filtros de visualización disponibles para su uso. Se admiten los filtros de visualización más comunes de 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>
```

Los filtros también se pueden aplicar 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"
```

Escenarios de ejemplo

Pérdida intermitente de ICMP (ping) a IP local

El tráfico que se reenvía a una IP local en un switch se coloca en la cola de los foros (literalmente, "para nosotros"). Ver el incremento en la cola de CoPP de Forus se relaciona con los paquetes descartados destinados al switch local. Esto es relativamente sencillo y fácil de conceptualizar.

En algunas condiciones, sin embargo, podría haber pérdidas en el tráfico con destino local que no se correlacionan de forma clara con las caídas de Forus.

Con un flujo de tráfico limitado a la CPU suficiente, la trayectoria de punt se sobresaturará más allá de la capacidad de CoPP para priorizar qué tráfico se controla. El tráfico se controla de forma "silenciosa" según el orden de entrada y salida.

En este escenario, se observa evidencia de políticas de plano de control en grandes volúmenes, pero el tipo de tráfico de interés (Forus en este ejemplo) no aumenta necesariamente de forma activa.

En resumen, si hay un volumen excepcionalmente alto de tráfico destinado a la CPU, evidenciado por la regulación de CoPP activa y demostrado con una captura de paquetes o depuración de punt FED, podría haber una pérdida que no se alinee con la cola que está resolviendo. En este escenario, determine por qué hay una cantidad excesiva de tráfico destinado a la CPU y tome medidas para aliviar la carga en el plano de control.

Redireccionamiento ICMP alto y funcionamiento DHCP lento

La CoPP en el switch Catalyst de la serie 9000 se organiza en 32 colas de hardware. Esas 32 colas de hardware se alinean con 20 índices de regulación individuales. Cada índice de regulador se correlaciona con una o más colas de hardware.

Funcionalmente, esto significa que varias clases de tráfico comparten un índice de regulador y están sujetas a un valor de regulador agregado común.

Un problema común observado en los switches con los agentes de retransmisión DHCP habilitados implica una respuesta DHCP lenta. Los clientes pueden obtener las IP de forma

esporádica, pero se necesitan varios intentos para completarlas y algunos clientes agotan el tiempo de espera.

La cola de redirección ICMP y la cola de difusión comparten un índice de regulador, por lo que un gran volumen de tráfico que se recibe y se enruta desde la misma interfaz virtual de switch (SVI) afecta a las aplicaciones que dependen del tráfico de difusión. Esto es especialmente notable cuando el switch actúa como un agente relay.

Este documento ofrece una explicación detallada del concepto y cómo mitigar: Troubleshooting DHCP Issues on Catalyst 9000 DHCP Relay Agents

Recursos adicionales

Resolución de problemas de DHCP lento o intermitente en los agentes de retransmisión DHCP de Catalyst 9000

Configuración de la captura de paquetes de CPU FED en switches Catalyst 9000

Switches Catalyst 9300: Configuración de políticas del plano de control

Configuración de la captura de paquetes: Guía de configuración de la administración de redes, Cisco IOS XE Bengaluru 17.6.x (switches Catalyst 9300)

Funcionamiento y solución de problemas de detección DHCP en switches Catalyst 9000

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).