排除Catalyst 9000交換機上的控制平面操作故障

目錄

簡介

背景資訊

- 技術

Catalyst 9000 CoPP

- CoPP實施
- 預設原則
- 調整CoPP

疑難排解

- 方法
- 實用的Show命令
  - 確定總體利用率和歷史利用率
  - 檢查控制層面策略
  - 收集有關傳送流量的資訊
  - 檢查CPU繫結的流量

常見案例

- 對本地IP的間斷ICMP(Ping)丟失
- 高ICMP重定向和DHCP運行緩慢

其他資源

簡介

本文檔介紹如何對運行Cisco IOS® XE的Catalyst 9000系列交換機上的控制層面運行狀況進行故障排除和驗證。

背景資訊

交換機的主要工作是儘快轉發資料包。大多數資料包在硬體中轉發，但某些型別的流量必須由系統CPU處理。到達CPU的流量會儘快得到處理。預計在CPU上會看到一定量的流量，但是過多的流量會導致操作問題。Catalyst 9000系列交換器預設結合強大的控制層面管制(CoPP)機制，以防止CPU的流量過度飽和所產生的問題。

在某些使用情形中，由於正常操作，會出現意想不到的問題。因果關係有時並不明顯，使問題難以解決。本文檔提供了驗證控制平面運行狀況的工具，並提供了有關如何處理涉及控制平面點焊或插入路徑問題的工作流。它還根據現場出現的問題提供了幾種常見方案。

請記住，CPU傳送路徑是有限的資源。現代硬體轉發交換機可以處理呈指數級成長的流量。Catalyst 9000系列交換機在任何給定時間的CPU上支援約19,000個每秒資料包(pps)。超出此閾值，則傳送的流量會受到無權值管制。

技術

- 轉發引擎驅動程式(FED)：這是Cisco Catalyst交換機的核心，負責所有硬體程式設計/轉發
- IOSd：這是在Linux核心上運行的Cisco IOS守護程式。它作為核心中的軟體進程運行
- 封包交付系統(PDS)：這是將封包傳輸到各個子系統或從各個子系統傳出的架構和程式。例如，它控制資料包如何從FED傳送到IOSd，反之亦然
- 控制平面(CP)：控制平面是一個通用術語，用於將涉及Catalyst交換機CPU的功能和流量組合在一起。這包括目的地為交換器或從交換器傳送的流量，例如跨距樹狀目錄通訊協定(STP)、熱待命路由器通訊協定(HSRP)和路由通訊協定。這也包括必須由CPU處理的應用層協定，如安全外殼(SSH)和簡單網路管理協定(SNMP)
- 資料層面(DP)：通常，資料層麵包含硬體ASIC和轉發流量，而無需控制層面的協助
- 傳送：在傳送到CP以處理的DP上截獲的入口協定控制資料包
- 插入：CP生成的協定資料包傳送到DP以從IO介面輸出
- LSMPI：Linux共用記憶體傳送介面

Catalyst 9000 CoPP

CoPP是Catalyst 9000系列交換機上CPU保護的基礎。使用CoPP時，系統生成的服務品質(QoS)策略應用於CPU傳送/插入路徑。CPU繫結流量分為許多不同的類，並隨後對映到與CPU關聯的各個硬體監察器。監察器可防止特定流量類別導致CPU過飽和。

CoPP實施

CPU密集型流量被歸類為隊列。這些隊列/類是系統定義的，不可由使用者配置。監察器是在硬體中配置的。Catalyst 9000系列支援32個隊列的32個硬體監察器。

具體值因平台而異。一般而言，有32個系統定義的佇列。這些隊列與類對映相關，類對映與監察器索引相關。監察器索引具有預設監察器速率。此速率可由使用者配置，不過對預設CoPP策略的更改會增加對意外服務影響的敏感性。

CoPP的系統定義值

類對映名稱	管制器索引（管制器編號）	CPU佇列（佇列編號）
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-控制	WK_CPP_POLICE_L2_ CONTROL(1)	WK_CPU_Q_L2_CONTROL(1)

類對映名稱	管制器索引（管制器編號）	CPU佇列（佇列編號）
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-低優先順序	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)
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-資料	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)

類對映名稱	管制器索引（管制器編號）	CPU佇列（佇列編號）
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-end-station	WK_CPP_POLICE_MULTICAST_SNOOPING(15)	WK_CPU_Q_MCAST_END_STA_SERVICE
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)

每個隊列與流量型別或特定功能集相關。此清單並非詳盡無遺：

CPU佇列和相關功能

CPU佇列（佇列編號）	功能
WK_CPU_Q_DOT1X_AUTH(0)	IEEE 802.1x連線埠型驗證
WK_CPU_Q_L2_CONTROL(1)	動態Trunk通訊協定(DTP) VLAN中繼線通訊協定(VTP) 連線埠彙總通訊協定(PAgP) 使用者端資訊訊號通訊協定(CISP) 消息會話中繼協定 多個VLAN註冊通訊協定(MVRP) 都會行動網路(MMN) 連結層級探索通訊協定(LLDP) 單向連結偵測(UDLD) 連結彙總控制通訊協定(LACP) 思科探索通訊協定(CDP) 生成樹通訊協定(STP)
WK_CPU_Q_FORUS_TRAFFIC(2)	主機，例如Telnet、Pingv4和Pingv6以及SNMP Keepalive/環回檢測 Initiate-Internet Key Exchange (IKE)協定(IPSec)
WK_CPU_Q_ICMP_GEN(3)	ICMP -無法到達目的地 ICMP-TTL已過期
WK_CPU_Q_ROUTING_CONTROL(4)	路由資訊通訊協定第1版(RIPv1) RIPv2 內部閘道路由通訊協定 (IGRP)

CPU佇列（佇列編號）	功能
	邊界閘道通訊協定(BGP) PIM-UDP 虛擬路由器備援通訊協定(VRRP) 熱待命路由器通訊協定第1版(HSRPv1) HSRPv2 閘道負載平衡通訊協定(GLBP) 標籤發佈通訊協定(LDP) 網路快取通訊協定(WCCP) 路由資訊協定下一代(RIPng) 開放最短路徑優先(OSPF) 開放最短路徑優先版本3(OSPFv3) 增強型內部閘道路由通訊協定(EIGRP) 增強型內部閘道路由通訊協定第6版(EIGRPv6) DHCPv6 通訊協定無關多點傳送(PIM) 通訊協定無關多點傳送版本6 (PIMv6) 熱待命路由器通訊協定下一代(HSRPng) IPv6控制 通用路由封裝(GRE) Keepalive 網路位址翻譯(NAT)傳送 中間系統到中間系統(IS IS)
WK_CPU_Q_FORUS_ADDR_RESOLUTION(5)	位址解析通訊協定(ARP) IPv6鄰居通告和鄰居請求
WK_CPU_Q_ICMP_REDIRECT(6)	網際網路控制訊息通訊協定(ICMP)重新導向

CPU佇列（佇列編號）	功能
WK_CPU_Q_INTER_FED_TRAFFIC(7)	用於內部通訊的第2層網橋域。
WK_CPU_Q_L2_LVX_CONT_PACK(8)	交換ID (XID)封包
WK_CPU_Q_EWLC_CONTROL(9)	嵌入式無線控制器(eWLC) [無線存取點的控制與布建(CAPWAP) (UDP 5246)]
WK_CPU_Q_EWLC_DATA(10)	eWLC資料包（CAPWAP資料，UDP 5247）
WK_CPU_Q_L2_LVX_DATA_PACK(11)	為對映請求傳送了未知的單播資料包。
WK_CPU_Q_BROADCAST(12)	所有型別的廣播
WK_CPU_Q_OPENFLOW(13)	學習快取溢位（第2層+第3層）
WK_CPU_Q_CONTROLLER_PUNT(14)	資料-存取控制清單(ACL)已滿 資料- IPv4選項 資料- IPv6逐跳 資料-資源不足/全部捕獲 資料-反向路徑轉送(RPF)未完成 收集資料包
WK_CPU_Q_TOPOLOGY_CONTROL(15)	生成樹通訊協定(STP) 彈性乙太網路通訊協定(REP) 共用跨距樹狀目錄通訊協定(SSTP)
WK_CPU_Q_PROTO_SNOOPING(16)	動態ARP檢測(DAI)的地址解析協定(ARP)監聽
WK_CPU_Q_DHCP_SNOOPING(17)	DHCP監聽
WK_CPU_Q_TRANSIT_TRAFFIC(18)	這用於由NAT傳送的資料包，這些資料包需要

CPU佇列（佇列編號）	功能
	在軟體路徑中處理。
WK_CPU_Q_RPF_FAILED(19)	資料- mRPF （組播RPF）失敗
WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	網際網路群組管理通訊協定(IGMP)/多點傳送接聽程式探索(MLD)控制
WK_CPU_Q_LOGGING(21)	存取控制清單(ACL)記錄
WK_CPU_Q_PUNT_WEBAUTH(22)	Web驗證
WK_CPU_Q_HIGH_RATE_APP(23)	廣播
WK_CPU_Q_EXCEPTION(24)	IKE指示 IP學習違規 IP連線埠資安違規 IP靜態位址違規 IPv6範圍檢查 遠端複製協定(RCP)異常 單播RPF失敗
WK_CPU_Q_SYSTEM_CRITICAL(25)	媒體訊號/無線代理ARP
WK_CPU_Q_NFL_SAMPLED_DATA(26)	Netflow抽樣資料和媒體服務代理(MSP)
WK_CPU_Q_LOW_LATENCY(27)	雙向轉發檢測(BFD)、精確時間協定(PTP)
WK_CPU_Q_EGR_EXCEPTION(28)	出口解析異常
WK_CPU_Q_STACKWISE_VIRTUAL_CONTROL(29)	前端堆疊協定，即SVL

CPU佇列（佇列編號）	功能
WK_CPU_Q_MCAST_DATA(30)	資料- (S，G)建立 資料-本地聯接 資料- PIM註冊 資料- SPT切換 資料-組播
WK_CPU_Q_GOLD_PKT(31)	金牌

預設原則

預設情況下，系統生成的CoPP策略應用於傳送/插入路徑。使用常見的基於MQC的命令可以檢視預設策略。它也可以在交換機配置中檢視。唯一允許在CPU/控制平面的入口或出口應用的策略是系統定義的策略。

使用show policy-map control-plane檢視應用於控制平面的策略：

```
<#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
```

調整CoPP

CoPP監察器速率可由使用者配置。使用者還可以停用隊列。

此示例演示如何調整單個監察器值。在本示例中，調整的類是「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
```

此示例演示如何完全停用隊列。停用隊列時請小心，因為這樣可能會導致CPU過飽和。

```
<#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
```

疑難排解

方法

CPU使用率受兩個基本活動（進程和中斷）的影響。進程是CPU執行的結構化活動，而中斷是指在資料平面上截獲並傳送到CPU以供操作的資料包。這些活動共同構成了CPU的總使用率。由於預設情況下啟用CoPP，服務影響並不一定與高CPU利用率相關。如果CoPP完成工作，CPU利用率不會受到很大影響。考慮CPU的整體利用率很重要，但總體利用率並不能反映整個情況。本部分中的show命令和實用程式用於快速評估CPU的運行狀況和辨識有關計算密集型流量的相關詳細資訊。

準則：

- 確定問題是否與控制平面有關。大多數中轉流量都透過硬體轉發。只有某些流量型別和某些場景涉及CPU和控制平面，因此在整個調查過程中請記住這一點。
- 瞭解您的利用率基線。瞭解正常使用率的情況很重要，這樣才能確定與標準值的偏差。
- 驗證進程和中斷的總體利用率。確定佔用意外量CPU週期的所有進程。如果利用率落在預期範圍之外，這可能會導致問題。瞭解系統的平均利用率很重要，這樣才能辨識標準之外的偏差。請記住，僅使用率並不能全面反映控制平面的運行狀況。
- 確定CoPP中是否存在主動遞增的丟包數。CoPP丟包數並不總是表示存在問題，但如果您對與主動監察的流量類相關的問題進行故障排除，則這是一個表明相關性的強有力指標。

實用的Show命令

該交換機可快速監控CPU運行狀況和CoPP統計資訊。還有實用的CLI，可快速確定計算密集型流量的入口點。

確定總體利用率和歷史利用率

- Show processes cpu sorted用於檢視整體CPU使用率。「sorted」引數根據使用百分比對進程輸出進行排序。使用更多CPU資源的進程位於輸出頂部。因中斷造成的使用率也以百分比表示。

```
<#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」提供過去60秒、5分鐘和72小時的CPU使用率歷史圖表。

<#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%
```

```
    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%
```

檢查控制層面策略

- 使用show platform hardware fed <switch> active qos queue stats internal cpu policer可以檢視聚合CoPP統計資訊和有關隊列/監察器結構的其他資訊。此輸出提供了自上次重置控制平面以來監察器統計資訊的歷史檢視。這些計數器也可以手動清除。 通常，由監察器提供的控制平面丟棄證據會指向相關隊列/類的問題，但確保在問題發生時丟棄主動增加。多次運行該命令以觀察增加隊列丟棄值的情況。

<#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/
```

收集有關傳送流量的資訊

這些命令用於收集有關傳送到CPU的流量的資訊，包括流量的型別和入口的物理點。

- Show platform software fed <switch> active punt cpuq all或「Show platform software fed <switch> active punt cpuq <0-31 Queue ID>」可用於檢視與所有或特定CPU隊列有關的統計資訊。

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

- 使用show platform software fed <switch> active punt cause summary快速檢視CPU中發現的所有不同流量型別。請注意，僅顯示非零原因。

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

- 使用命令show platform software fed <switch> active punt rates interfaces"可快速檢視傳入系統的介面CPU的流量。此命令僅顯示具有非零輸入隊列的介面。

<#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

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

- 使用「show platform software fed <switch> active punt rates interfaces <IF-ID>」深入檢視和檢視介面的單個隊列。此命令顯示聚合統計資訊，並且可用於檢視歷史輸入隊列活動以及流量是否已受到管制。

<#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

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

檢查CPU繫結的流量

Catalyst 9000系列交換機提供用於監控和檢視計算密集型流量的實用程式。使用這些工具瞭解哪些流量被主動傳送到CPU。

內嵌式封包擷取(EPC)

控制平面上的EPC可在任一方向（或兩者）上完成。對於傳送的流量，捕獲入站。控制平面上的EPC可以儲存到緩衝區或檔案。

<#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
```

擷取結果可透過簡短或詳細的輸出來檢視。

<#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
```

擷取結果可以直接寫入檔案，或從緩衝區匯出。

```
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#
```

FED CPU資料包捕獲

Catalyst 9000系列交換機支援調試實用程式，該實用程式允許增強與CPU之間的資料包的可視性。

```
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)
```

緩衝區內容包含用於輸出的簡短和詳細選項。

<#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
```

有許多顯示過濾器可供使用。支援最常見的Wireshark顯示過濾器。

<#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>
```

過濾器也可以作為捕獲過濾器應用。

<#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"
```

常見案例

對本地IP的間斷ICMP(Ping)丟失

轉發到交換機上的本地IP的流量在Forus（字面意思是「for us」）隊列中傳送。Forus CoPP隊列中的增加與發往本地交換機的丟棄資料包有關。這相對直截了當，且易於概念化。

但在某些情況下，可能會丟失與Forus丟棄不完全相關的本地目標流量。

如果流量足夠多的CPU流量，則傳送路徑會變得過度飽和，超出了CoPP優先處理被管制流量的能力。流量以先入先出方式進行「靜默式」管制。

在此場景中，可以看到大量控制平面策略的證據，但是感興趣的流量型別（在本示例中為Forus）不一定會主動增加。

總之，如果存在異常高的CPU流量（透過活動的CoPP管制進行證明並透過資料包捕獲或FED傳送調試進行演示），則可能會出現與您正在故障排除的隊列不一致的丟失。在此場景中，確定超出CPU流量的原因，並採取措施減輕控制平面的負擔。

高ICMP重定向和DHCP運行緩慢

Catalyst 9000系列交換機上的CoPP可組織為32個硬體隊列。這32個硬體隊列與20個單獨的監察器索引一致。 每個監察器索引都與一個或多個硬體隊列相關。

從功能上講，這意味著多個流量類共用一個監察器索引，並受一個通用聚合監察器值的約束。

在啟用了DHCP中繼代理的交換機上發現的一個常見問題是DHCP響應緩慢。客戶端可以偶爾獲得IP，但需要多次嘗試才能完成，並且某些客戶端會超時。

ICMP重新導向佇列和廣播佇列共用一個管制器索引，因此在同一交換器虛擬介面(SVI)上接收和路由輸出的大量流量會影響依賴廣播流量的應用程式。 當交換機充當中繼代理時，這一點尤其明顯。

本文檔深入解釋了這一概念以及如何緩解：排除Catalyst 9000 DHCP中繼代理上的DHCP故障

其他資源

對Catalyst 9000 DHCP中繼代理上的慢速或間歇性DHCP進行故障排除

在Catalyst 9000交換機上配置FED CPU資料包捕獲

Catalyst 9300交換機：配置控制層面策略

配置資料包捕獲-網路管理配置指南，Cisco IOS XE Bengaluru 17.6.x（Catalyst 9300交換機）

操作Catalyst 9000交換機上的DHCP監聽並排除故障

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。