Risoluzione dei problemi relativi alle operazioni del Control Plane sugli switch Catalyst 9000

Sommario

Introduzione

In questo documento viene descritto come risolvere i problemi e convalidare lo stato del control plane sugli switch Catalyst della famiglia 9000 con Cisco IOS® XE.

Premesse

Il compito principale di uno switch è inoltrare i pacchetti nel più breve tempo possibile. La maggior parte dei pacchetti viene inoltrata nell'hardware, ma alcuni tipi di traffico devono essere gestiti dalla CPU del sistema. Il traffico che arriva alla CPU viene gestito il più rapidamente possibile. Si prevede che alla CPU venga rilevata una certa quantità di traffico, ma una sovrabbondanza genera problemi operativi. Per impostazione predefinita, la famiglia di switch Catalyst 9000 incorpora un robusto meccanismo CoPP (Control Plane Policing) per prevenire i problemi causati dalla sovrasaturazione del traffico della CPU.

In alcuni casi di utilizzo si verificano problemi imprevisti in funzione del normale funzionamento. La correlazione tra causa ed effetto a volte non è ovvia, il che rende il problema difficile da affrontare.

Questo documento fornisce gli strumenti per convalidare lo stato del control plane e fornisce un flusso di lavoro su come affrontare i problemi che coinvolgono il punt o il percorso di inserimento del control plane. Fornisce inoltre diversi scenari comuni basati sui problemi riscontrati sul campo.

Tenete presente che il percorso del punt della CPU è una risorsa limitata. I moderni switch di inoltro hardware possono gestire volumi di traffico sempre maggiori. La famiglia di switch Catalyst 9000 supporta circa 19.000 pacchetti al secondo (pps) aggregati alla CPU in un determinato momento. Supera questa soglia e il traffico puntato viene sorvegliato senza pesare.

Terminologia

- Driver del motore di inoltro (FED): questo è il cuore dello switch Cisco Catalyst ed è responsabile di tutta la programmazione/inoltro hardware
- IOSd: questo è il daemon Cisco IOS in esecuzione sul kernel Linux. Viene eseguito come processo software all'interno del kernel
- Packet Delivery System (PDS): si tratta dell'architettura e del processo con cui i pacchetti vengono consegnati da e verso i vari sottosistemi. Ad esempio, controlla il modo in cui i pacchetti vengono consegnati dalla FED all'IOSd e viceversa
- Control Plane (CP):il control plane è un termine generico utilizzato per raggruppare le funzioni e il traffico che interessano la CPU dello switch Catalyst. Ciò include il traffico, ad esempio il protocollo Spanning Tree Protocol (STP), il protocollo HSRP (Hot Standby Router Protocol) e i protocolli di routing destinati allo switch o inviati dallo switch. Sono inclusi anche protocolli a livello di applicazione come Secure Shell (SSH) e Simple Network Management Protocol (SNMP) che devono essere gestiti dalla CPU
- Data Plane (DP): in genere il piano dati comprende gli ASIC hardware e il traffico inoltrati senza l'assistenza del Control Plane
- Punt:Pacchetto di controllo del protocollo in entrata che ha intercettato il DP inviato al CP per elaborarlo
- Inject:Pacchetto di protocollo generato da CP inviato a DP per l'uscita su interfacce I/O
- LSMPI:Interfaccia Punt memoria condivisa Linux

Catalyst 9000 CoP

La base della protezione della CPU sulla famiglia di switch Catalyst 9000 è CoPP. Con CoPP, viene applicata una policy QoS (Quality of Service) generata dal sistema sul percorso punt/inserimento della CPU. Il traffico basato sulla CPU viene raggruppato in molte classi diverse e successivamente mappato sui singoli policy hardware associati alla CPU. I policer impediscono la sovracorsaturazione della CPU da parte di una particolare classe di traffico.

Implementazione CoPP

Il traffico basato sulla CPU viene classificato nelle code. Queste code/classi sono definite dal sistema e non possono essere configurate dall'utente. I criteri sono configurati nell'hardware. La famiglia Catalyst 9000 supporta 32 policer hardware per 32 code.

I valori specifici variano da piattaforma a piattaforma. In generale, esistono 32 code definite dal

sistema. Queste code si riferiscono a mappe di classi, che si riferiscono agli indici dei policer. Gli indici del policer hanno una frequenza del policer predefinita. Questo tasso è configurabile dall'utente, sebbene le modifiche apportate al criterio CoPP predefinito aumentino la vulnerabilità a un impatto imprevisto sul servizio.

Valori definiti dal sistema per CoPP

Mappe classi - Nomi	Indice Policer (n. Policer)	Code CPU (n. coda)
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(2)
system-cpp-Police-controllo-topologia	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)

Mappe classi - Nomi	Indice Policer (n. Policer)	Code CPU (n. coda)
system-cpp-Police-sys-data	WK_CPP_POLICE_SYS _DATI(10)	WK_CPU_Q_LEARNING_CACHE_OVFL(13 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(1)	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(1)
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_TION_SE
system-cpp-default	WK_CPP_POLICE_DEFAULT_POLICER(16)	WK_CPU_Q_DHCP_SNOOPING(17)

Mappe classi - Nomi	Indice Policer (n. Policer)	Code CPU (n. coda)
		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)

Ogni coda è correlata a un tipo di traffico o a un insieme di funzionalità specifico. L'elenco che segue non è esaustivo:

Code CPU e funzionalità associate

Code CPU (n. coda)	Funzionalità
WK_CPU_Q_DOT1X_AUTH(0)	Autenticazione basata sulla porta IEEE 802.1x
WK_CPU_Q_L2_CONTROL(1)	Protocollo DTP (Dynamic Trunking Protocol) Protocollo VLAN Trunking Protocol (VTP) Protocollo PAgP (Port Aggregation Protocol) Protocollo CISP (Client Information Signaling Protocol) Protocollo di inoltro sessione messaggi Protocollo MVRP (Multiple VLAN

Code CPU (n. coda)	Funzionalità
	Registration Protocol) MMN (Metropolitan Mobile Network) LLDP (Link Level Discovery Protocol) UDLD (UniDirectional Link Detection) Protocollo LACP (Link Aggregation Control Protocol) Protocollo CDP (Cisco Discovery Protocol) STP (Spanning Tree Protocol)
WK_CPU_Q_FORUS_TRAFFIC(2)	Host come Telnet, Pingv4 e Pingv6 e SNMP Rilevamento keepalive/loopback Protocollo IPSec (Initiate-Internet Key Exchange)
WK_CPU_Q_ICMP_GEN(3)	ICMP - destinazione non raggiungibile ICMP-TTL scaduto
WK_CPU_Q_ROUTING_CONTROL(4)	Protocollo RIPv1 (Routing Information Protocol versione 1) RIPv2 IGRP (Interior Gateway Routing Protocol) Border Gateway Protocol (BGP) PIM-UDP Protocollo VRRP (Virtual Router Redundancy Protocol) Protocollo HSRPv1 (Hot Standby Router Protocol versione 1) HSRPv2

Code CPU (n. coda)	Funzionalità
	Protocollo GLBP (Gateway Load Balancing Protocol) Protocollo LDP (Label Distribution Protocol) Protocollo WCCP (Web Cache Communication Protocol) Protocollo RIPng (Routing Information Protocol next-generation) OSPF (Open Shortest Path First) Open Shortest Path First versione 3 (OSPFv3) Protocollo EIGRP (Enhanced Interior Gateway Routing Protocol) Enhanced Interior Gateway Routing Protocol versione 6 (EIGRPv6) DHCPv6 PIM (Protocol Independent Multicast) PIMv6 (Protocol Independent Multicast versione 6) Protocollo HSRPng (Hot Standby Router Protocol next-generation) Controllo IPv6 GRE (Generic Routing Encapsulation) keepalive Punt Network Address Translation (NAT) IS-IS (Intermediate System-to-Intermediate System)
WK_CPU_Q_FORUS_ADDR_RESOLUTION(5)	Protocollo ARP (Address Resolution Protocol) Annuncio router adiacente IPv6 e richiesta router adiacenti

Code CPU (n. coda)	Funzionalità
WK_CPU_Q_ICMP_REDIRECT(6)	Reindirizzamento Internet Control Message Protocol (ICMP)
WK_CPU_Q_INTER_FED_TRAFFIC(7)	Inserimento del dominio bridge di layer 2 per la comunicazione interna.
WK_CPU_Q_L2_LVX_CONT_PACK(8)	Pacchetto ID Exchange (XID)
WK_CPU_Q_EWLC_CONTROL(9)	Embedded Wireless Controller (eWLC) [Control and Provisioning of Wireless Access Point (CAPWAP) (UDP 5246)]
WK_CPU_Q_EWLC_DATA(10)	pacchetto dati eWLC (CAPWAP DATA, UDP 5247)
WK_CPU_Q_L2_LVX_DATA_PACK(1)	Pacchetto unicast sconosciuto su cui è stato eseguito il push per la richiesta di mapping.
WK_CPU_Q_BROADCAST(12)	Tutti i tipi di trasmissione
WK_CPU_Q_OPENFLOW(13)	Overflow della cache di apprendimento (livello 2 + livello 3)
WK_CPU_Q_CONTROLLER_PUNT(14)	Dati - elenco di controllo di accesso (ACL) completo Dati - Opzioni IPv4 Dati - Hop-by-hop IPv6 Dati: risorse insufficienti/cattura di tutto Dati - Inoltro percorso inverso (RPF) incompleto Pacchetto verde
WK_CPU_Q_TOPOLOGY_CONTROL(15)	STP (Spanning Tree Protocol)

Code CPU (n. coda)	Funzionalità
	Protocollo REP (Resilient Ethernet Protocol) Protocollo SSTP (Shared Spanning Tree Protocol)
WK_CPU_Q_PROTO_SNOOPING(16)	Snooping ARP (Address Resolution Protocol) per l'ispezione ARP dinamica (DAI)
WK_CPU_Q_DHCP_SNOOPING(17)	snooping DHCP
WK_CPU_Q_TRANSIT_TRAFFIC(18)	Questa opzione viene usata per i pacchetti puntati da NAT, che devono essere gestiti nel percorso software.
WK_CPU_Q_RPF_FAILED(19)	Dati - mRPF (multicast RPF) non riuscito
WK_CPU_Q_MCAST_END_STATION _SERVIZIO(20)	Controllo IGMP (Internet Group Management Protocol) / MLD (Multicast Listener Discovery)
WK_CPU_Q_LOGGING(21)	Registrazione Access Control List (ACL)
WK_CPU_Q_PUNT_WEBAUTH(2)	Autenticazione Web
WK_CPU_Q_HIGH_RATE_APP(23)	Trasmissione
WK_CPU_Q_EXCEPTION(24)	Indicazione IKE Violazione di apprendimento IP Violazione della sicurezza delle porte IP Violazione dell'indirizzo IP statico Controllo ambito IPv6 Eccezione RCP (Remote Copy Protocol)

Code CPU (n. coda)	Funzionalità
	Errore RPF unicast
WK_CPU_Q_SYSTEM_CRITICAL(25)	Segnalazione supporti/ARP proxy wireless
WK_CPU_Q_NFL_SAMPLED_DATA(26)	Dati campionati da Netflow e MSP (Media Services Proxy)
WK_CPU_Q_LOW_LATENCY(27)	BFD (Bidirectional Forwarding Detection), PTP (Precision Time Protocol)
WK_CPU_Q_EGR_EXCEPTION(28)	Eccezione risoluzione in uscita
WK_CPU_Q_STACKWISE_VIRTUAL_CONTROL(29)	Protocolli di stacking sul lato anteriore, ossia SVL
WK_CPU_Q_MCAST_DATA(30)	Dati: creazione (S,G) Dati - join locali Dati - Registrazione PIM Dati - Switchover SPT Dati - Multicast
WK_CPU_Q_GOLD_PKT(31)	Oro

Criterio predefinito

Per impostazione predefinita, il criterio CoPP generato dal sistema viene applicato al percorso punt/inserimento. È possibile visualizzare il criterio predefinito utilizzando i comandi comuni basati su MQC. Può essere visualizzato anche nella configurazione dello switch. L'unica regola che può essere applicata in entrata o in uscita dalla CPU o dal control-plane è quella definita dal sistema.

Utilizzare "show policy-map control-plane" per visualizzare la policy applicata al control-plane:

<#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
```

Regola CoPP

Le velocità dei policer CoPP sono configurabili dall'utente. Gli utenti possono inoltre disattivare le code.

In questo esempio viene illustrato come regolare un singolo valore del policer. In questo esempio, la classe adeguata è "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
```

In questo esempio viene illustrato come disabilitare completamente una coda. procedere con cautela quando si disabilitano le code, in quanto ciò potrebbe causare una possibile sovracapacità della 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
```

Risoluzione dei problemi

Metodologia

L'utilizzo della CPU è influenzato da due attività di base: i processi e l'interruzione. I processi sono attività strutturate che la CPU esegue mentre l'interruzione fa riferimento a pacchetti intercettati sul dataplane e inviati alla CPU per l'azione. Insieme, queste attività costituiscono l'utilizzo totale della CPU. Poiché CoPP è abilitato per impostazione predefinita, l'impatto del servizio non è necessariamente correlato a un utilizzo elevato della CPU. Se CoPP esegue il proprio lavoro, l'utilizzo della CPU non subisce un impatto significativo. È importante considerare l'utilizzo complessivo della CPU, ma l'utilizzo complessivo non racconta l'intera storia. I comandi e le utilità show in questa sezione vengono usati per valutare rapidamente lo stato della CPU e per identificare i dettagli rilevanti sul traffico basato sulla CPU.

Linee guida:

- Determinare se il problema riguarda il control plane. La maggior parte del traffico di transito viene inoltrata nell'hardware. Solo alcuni tipi di traffico e alcuni scenari coinvolgono la CPU e il control-plane, quindi tenete presente questo nel corso dell'indagine.
- Comprendere le previsioni di utilizzo. È importante comprendere l'aspetto dell'utilizzo normale in modo da poter identificare eventuali deviazioni dalla norma.

- Convalidare l'utilizzo complessivo sia per i processi che per le interruzioni. Identificare tutti i processi che assorbono volumi imprevisti di cicli della CPU. Se l'utilizzo non rientra nell'intervallo previsto, è possibile che si verifichino problemi. È importante comprendere l'utilizzo medio di un sistema, in modo da riconoscere le deviazioni al di fuori della norma. Tenete presente che l'utilizzo da solo non è un quadro completo della salute dei control plane.
- Determinare se si verificano cali in aumento nel CoPP. Le cali del CoPP non sono sempre indicative di un problema, ma se si risolve un problema relativo a una classe di traffico attivamente monitorata, questo è un forte indicatore di rilevanza.

Comandi Show

Lo switch consente un rapido controllo dello stato della CPU e delle statistiche CoPP. La CLI è inoltre utile per determinare rapidamente il punto di entrata del traffico basato sulla CPU.

Determinare l'utilizzo complessivo e storico

- L'opzione "Mostra processi ordinati nella CPU" viene utilizzata per visualizzare l'utilizzo complessivo della CPU. L'argomento "ordinato" ordina l'output del processo in base alla percentuale di utilizzo. I processi che utilizzano più risorse CPU sono i primi dell'output. L'utilizzo a causa di interruzioni è fornito anche come percentuale.

<#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
```

- "Mostra cronologia CPU processi" fornisce un grafico cronologico dell'utilizzo della CPU negli ultimi 60 secondi, 5 minuti e 72 ore.

<#root>

```
Catalyst-9600#

show processes cpu history


    9997777766666888886666677777777788888777776666699999888886 6

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

    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%




    9998989899998989989989989898899999898898989999999899999999
    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%
```

Verifica Control Plane Policing

- Utilizzare "show platform hardware feed <switch> active qos queue stats internal cpu policer" per visualizzare le statistiche CoPP aggregate e ulteriori informazioni sulla struttura della coda/del policer. Questo output fornisce una visualizzazione storica delle statistiche del

policer dall'ultimo reset del control plane. Anche questi contatori sono cancellabili manualmente. In genere, la prova di cadute del control-plane da parte del policer indica un problema con la coda/classe associata, ma assicurarsi che le cadute incrementino attivamente mentre si verifica il problema. Eseguire il comando più volte per osservare l'aumento dei valori di rilascio della coda.

<#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/
```

Raccogliere informazioni sul traffico puntato

Questi comandi vengono utilizzati per raccogliere informazioni sul traffico puntato alla CPU, tra cui il tipo di traffico e i punti fisici di ingresso.

- È possibile utilizzare l'opzione "Show platform software fed <switch> active punt cpuq all" o "Show platform software fed <switch> active punt cpuq <0-31 Queue ID>" per visualizzare le statistiche relative a tutte o a una coda CPU specifica.

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

- Utilizzare "show platform software fed <switch> active punt cause summary" per esaminare rapidamente tutti i diversi tipi di traffico rilevati sulla CPU. Vengono visualizzate solo le cause diverse da zero.

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

- Utilizzare il comando "show platform software fed <switch> active punt rate interfaces" per visualizzare rapidamente le interfacce con traffico basato sulla CPU in entrata nel sistema. Questo comando visualizza solo le interfacce con una coda di input diversa da zero.

<#root>

```
C9300#

show platform software fed switch active punt rates interfaces

Punt Rate on Interfaces Statistics

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

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

- Utilizzare "show platform software fed <switch> active punt rates interfaces <IF-ID>" per espandere e visualizzare le singole code dell'interfaccia. Questo comando mostra le statistiche di aggregazione e può essere utilizzato per visualizzare l'attività cronologica della coda di input e se il traffico è stato controllato.

<#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

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

Ispeziona traffico associato alla CPU

La famiglia di switch Catalyst 9000 offre utility per il monitoraggio e la visualizzazione del traffico basato sulla CPU. Utilizzare questi strumenti per capire quale traffico viene indirizzato attivamente alla CPU.

EPC (Embedded Packet Capture)

L'EPC sul piano di controllo può essere eseguito in entrambe le direzioni (o in entrambe). Per il traffico puntuale, cattura in entrata. EPC sul piano di controllo può essere salvato nel buffer o nel file.

<#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
```

I risultati dell'acquisizione possono essere visualizzati in un output breve o dettagliato.

<#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
```

I risultati dell'acquisizione possono essere scritti direttamente nel file o esportati dal buffer.

<#root>

```
C9300#

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

Export Started Successfully

Export completed for capture point CONTROL
C9300#
C9300#

dir flash: | in control.pcap

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

Acquisizione pacchetti CPU FED

La famiglia di switch Catalyst 9000 supporta un'utility di debug che consente una maggiore visibilità dei pacchetti da e verso 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)
```

Il contenuto del buffer dispone di opzioni di output brevi e dettagliate.

<#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
```

Sono disponibili molti filtri di visualizzazione. Sono supportati i filtri di visualizzazione più comuni di 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>
```

I filtri possono essere applicati anche come filtri di acquisizione.

<#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"
```

Scenari comuni

Perdita ICMP (Ping) intermittente su IP locale

Il traffico che viene inoltrato a un IP locale su uno switch viene puntato nella coda Forus (letteralmente "per noi"). L'incremento rilevato nella coda Forus CoPP si riferisce ai pacchetti ignorati destinati allo switch locale. Si tratta di un'operazione relativamente semplice e concettuale.

In alcune condizioni, tuttavia, il traffico destinato localmente potrebbe non essere correttamente correlato alle cadute dei Forus.

Con un sufficiente flusso di traffico basato sulla CPU, il percorso punt diventa sovraccarico oltre la capacità del CoPP di stabilire la priorità del traffico da controllare. Il traffico viene sorvegliato "in modo silenzioso" su base "first-in" e "first-out".

In questo scenario, si vede la prova di policing control-plane in grandi volumi, ma il tipo di traffico di interesse (Forus in questo esempio) non necessariamente aumenta attivamente.

In sintesi, se il volume del traffico basato sulla CPU è eccezionalmente elevato, come dimostrato dal policing CoPP attivo e dimostrato con l'acquisizione di pacchetti o il debug del punt FED, è possibile che si verifichi una perdita non allineata alla coda su cui si sta eseguendo la risoluzione dei problemi. In questo scenario, determinare il motivo per cui è presente una quantità eccessiva di traffico basato sulla CPU e adottare misure per alleggerire il carico sul control plane.

Reindirizzamenti ICMP elevati e funzionamento DHCP lento

Il protocollo CoPP sugli switch Catalyst serie 9000 è organizzato in 32 code hardware. Queste 32 code hardware sono allineate a 20 indici dei singoli policer. Ogni indice del policer è correlato a una o più code hardware.

Dal punto di vista funzionale, questo significa che più classi di traffico condividono un indice di un policer e sono soggette a un valore di policer aggregato comune.

Un problema comune rilevato sugli switch con gli agenti di inoltro DHCP abilitati comporta una risposta DHCP lenta. I client sono in grado di ottenere gli indirizzi IP in modo sporadico, ma sono necessari diversi tentativi per completare l'operazione e alcuni client scadono.

La coda di reindirizzamento ICMP e la coda Broadcast condividono un indice policer, quindi un elevato volume di traffico ricevuto e indirizzato dalla stessa interfaccia virtuale dello switch (SVI) influisce sulle applicazioni che si basano sul traffico broadcast. Ciò è particolarmente evidente quando lo switch funge da agente di inoltro.

Questo documento offre una spiegazione dettagliata del concetto e del modo per mitigare i problemi: Risoluzione dei problemi DHCP sugli agenti di inoltro DHCP Catalyst 9000

Ulteriori risorse

Risoluzione dei problemi relativi a DHCP lenti o intermittenti sugli agenti di inoltro DHCP Catalyst 9000

Configurazione dell'acquisizione di pacchetti CPU FED sugli switch Catalyst 9000

Switch Catalyst 9300: configurazione di Control Plane Policing

Configurazione dell'acquisizione dei pacchetti - Guida alla configurazione della gestione della rete, Cisco IOS XE Bengaluru 17.6.x (switch Catalyst 9300)

Funzionamento e risoluzione dei problemi di snooping DHCP sugli switch Catalyst 9000

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).