Probleemoplossing voor besturingsplatformbewerkingen op Catalyst 9000 Switches

Inhoud

- Inleiding
- Achtergrondinformatie
  - Terminologie
- Catalyst 9000 CoP switch
  - CoPP-implementatie
  - Standaardbeleid
  - CoPP aanpassen
- Problemen oplossen
  - Methodologie
  - Handige opdrachten weergeven
    - Bepaal het algemene en historische gebruik
    - Toezicht op besturingsplane controleren
    - Verzamel informatie over bestraft verkeer
    - Inspecteer CPU-gebonden verkeer
- Gemeenschappelijke scenario's
  - Intermitterend ICMP-verlies (Ping) voor lokaal IP
  - Hoge ICMP-omleidingen en langzame DHCP-werking
- Aanvullende bronnen

Inleiding

Dit document beschrijft hoe u de gezondheid van het controlevliegtuig kunt oplossen en valideren op Catalyst 9000-Series switches waarop Cisco IOS® XE wordt uitgevoerd.

Achtergrondinformatie

Het belangrijkste werk van een switch is om pakketten zo snel mogelijk door te sturen. De meeste pakketten worden doorgestuurd in hardware, maar bepaalde typen verkeer moeten worden verwerkt door de systeem-CPU. Het verkeer dat bij de CPU aankomt, wordt zo snel mogelijk verwerkt. Verwacht wordt dat er een bepaalde hoeveelheid verkeer bij de CPU te zien is, maar een overvloed leidt tot operationele problemen. Catalyst 9000 reeks switches bevat standaard een robuust CoPP-mechanisme (Control Plane Policing) om problemen te voorkomen die worden veroorzaakt door oververzadiging van het verkeer op de CPU.

Onverwachte problemen ontstaan in bepaalde gebruiksgevallen als functie van de normale werking. De correlatie tussen oorzaak en gevolg is soms niet duidelijk, wat het probleem moeilijk te benaderen maakt. Dit document biedt u gereedschappen om de gezondheid van het besturingsplane te valideren en biedt een workflow voor het benaderen van problemen die betrekking hebben op het besturingsplantpunt of het injectiepad. Het bevat ook verschillende gemeenschappelijke scenario's die zijn gebaseerd op problemen die in het veld worden gezien.

Houd in gedachten dat het CPU puntpad een beperkte bron is. De moderne hardware-doorsturen switches kunnen exponentieel groter verkeersvolume verwerken. Catalyst 9000 reeks switches ondersteunt ongeveer 19.000 pakketten-per-seconde (pps) in totaal bij de CPU op een willekeurig moment. Overschrijd deze drempel, en gestraft verkeer wordt gecontroleerd zonder gewicht.

Terminologie

- Forwarding Engine Driver (FED): dit is het hart van de Cisco Catalyst switch en is verantwoordelijk voor alle hardwareprogrammering/doorsturen
- IOSd: Dit is de Cisco IOS daemon die op de Linux-kernel draait. Het wordt uitgevoerd als een softwareproces binnen de kernel
- Packet Delivery System (PDS): dit is de architectuur en het proces van hoe pakketten worden geleverd aan en van de verschillende subsystemen. Zo wordt bijvoorbeeld bepaald hoe pakketten van de FED aan IOSd worden geleverd en omgekeerd
- Besturingsplane (CP): Het besturingsplane is een generieke term die wordt gebruikt om de functies en het verkeer te groeperen die de CPU van de Catalyst Switch omvatten. Dit omvat verkeer zoals Spanning Tree Protocol (STP), Hot Standby Router Protocol (HSRP) en routeringsprotocollen die bestemd zijn voor de switch of die vanaf de switch worden verzonden. Dit omvat ook toepassingslaagprotocollen zoals Secure Shell (SSH) en Simple Network Management Protocol (SNMP) die door de CPU moeten worden verwerkt
- Gegevensvlak (DP):Het gegevensvlak omvat gewoonlijk de hardware-ASIC's en het verkeer dat wordt doorgestuurd zonder hulp van het besturingsplane
- Punt:Ingress-protocolcontrolepakket dat op DP is onderschept en naar de CP is verzonden om het te verwerken
- Injecteren:CP gegenereerd protocolpakket verzonden naar DP om uit te stappen op IO-interface(s)
- LSMPI:Linux gedeelde geheugen-puntinterface

Catalyst 9000 CoP switch

De basis van CPU-beveiliging op de Catalyst 9000-reeks switches is CoPP. Met CoPP wordt een QoS-beleid (Quality of Service) door het systeem gegenereerd op het CPU point/injectpad. CPU-gebonden verkeer wordt gegroepeerd in vele verschillende klassen en vervolgens toegewezen aan de afzonderlijke hardwarepolicers die aan de CPU zijn gekoppeld. De policers voorkomen oververzadiging van de CPU door een bepaalde verkeersklasse.

CoPP-implementatie

CPU-gebonden verkeer wordt geclassificeerd in wachtrijen. Deze wachtrijen/-klassen zijn door het

systeem gedefinieerd en kunnen niet door de gebruiker worden geconfigureerd. Politieagenten worden ingesteld op hardware. De Catalyst 9000 reeks ondersteunt 32 hardware policers voor 32 wachtrijen.

Specifieke waarden verschillen per platform. In het algemeen zijn er 32 systeemgedefinieerde wachtrijen. Deze wachtrijen hebben betrekking op class-maps, die betrekking hebben op politiemensen. De policer indices hebben een standaard policer rate. Dit tarief is gebruiker-configureerbaar, alhoewel veranderingen in het standaard CoPP beleid gevoeligheid voor een onverwacht de diensteffect verhogen.

Systeemgedefinieerde waarden voor CoPP

Namen voor klassekaarten	Policer Index (policer-nr.)	CPU-wachtrijen (wachtrij
system-cpp-politie-data	WK_CPP_POLITIE_DATA(0)	WK_CPU_Q_ICMP_GEN(3) WK_CPU_Q_BROADCAST(12) WK_CPU_Q_ICMP_REDIRECT(6)
system-cpp-Police-l2-controle	WK_CPP_POLITIE_L2_ CONTROLE(1)	WK_CPU_Q_L2_CONTROL(1)
System-CPP-Police-Routing-Control	WK_CPP_POLITIE_ROUTING_CONTROL(2)	WK_CPU_Q_ROUTING_CONTROL(WK_CPU_Q_LOW_LATENCY (27)
systeem-cpp-politie-controle-lage prioriteit	WK_CPP_POLITIE_CO NTROL_LOW_PRI(3)	WK_CPU_Q_ALGEMEEN_PUNT(25)
system-cpp-Police-punt-webauth	WK_CPP_POLITIE_PU NT_WEBAUTH(7)	WK_CPU_Q_PUNT_WEBAUTH(22)
systeem-cpp-politie-topologiebeheer	WK_CPP_POLITIE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTRO
systeem-cpp-politie-multicast	WK_CPP_POLITIE_MULTICAST(9)	WK_CPU_Q_TRANSIT_TRAFFIC(18 WK_CPU_Q_MCAST_DATA(30)
systeem-cpp-politie-sys-data	WK_CPP_POLITIE_SYS_DATA(10)	WK_CPU_Q_LEARNING_CACHE_O WK_CPU_Q_CRYPTO_CONTROL(2

Namen voor klassekaarten	Policer Index (policer-nr.)	CPU-wachtrijen (wachtrij
		WK_CPU_Q_UITZONDERING(24) WK_CPU_Q_EGR_EXCEPTION(28) WK_CPU_Q_NFL_STEEKPROEF_G WK_CPU_Q_GOLD_PKT(31) WK_CPU_Q_RPF_MISLUKT(19)
System-cpp-Police-dot1x-auth	WK_CPP_POLITIE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH(0)
System-CPP-Police-protocolsnooping	WK_CPP_POLITIE_PR(12)	WK_CPU_Q_PROTO_SNOOPING(1
systeem-cpp-politie-doorspoelen	WK_CPP_POLITIE_SW_FWD (13)	WK_CPU_Q_SW_FORWARDING_Q WK_CPU_Q_LOGGING(21) WK_CPU_Q_L2_LVX_DATA_PACK(
systeem-cpp-politie-forus	WK_CPP_POLITIE_FORUS(14)	WK_CPU_Q_FORUS_ADDR_Resolu WK_CPU_Q_FORUS_TRAFFIC(2)
system-cpp-Police-multicast-end-station	WK_CPP_POLICA_MULTICAST_SNOOPING(15)	WK_CPU_Q_MCAST_END_STA TION_SERVICE(20)
systeemeigen standaard	WK_CPP_POLITIE_DEFAULT_POLICER(16)	WK_CPU_Q_DHCP_SNOOPING(17) WK_CPU_Q_ONGEBRUIKT(7) WK_CPU_Q_EWLC_CONTROL(9) WK_CPU_Q_EWLC_DATA(10)
systeem-cpp-politie-stackwise-virt-control	WK_CPP_STACKWISE_VIRTUAL_CONTROL(5)	WK_CPU_Q_STACKWISE_VIRTUAL (29)
System-cpp-	WK_CPP_ L2_LVX_CONT_PACK(4)	WK_CPU_Q_L2_LVX_CONT_PACK(

Namen voor klassekaarten	Policer Index (policer-nr.)	CPU-wachtrijen (wachtrij
Police-l2lvx-Control		

Elke wachtrij heeft betrekking op een verkeerstype of een bepaalde reeks functies. Dit is geen uitputtende lijst:

CPU-wachtrijen en bijbehorende functies

CPU-wachtrijen (wachtrij nr.)	Functie(s)
WK_CPU_Q_DOT1X_AUTH(0)	IEEE 802.1x-poortgebaseerde verificatie
WK_CPU_Q_L2_CONTROL(1)	Dynamic Trunking Protocol (DTP) VLAN-trunkingprotocol (VTP) Poortaggregatieprotocol (PAgP) Clientinformatiesignaleringsprotocol (CISP) Relay-protocol voor berichtensessie MVRP (Multiple VLAN Registration Protocol) Metropolitan Mobile Network (MMN) LLDP-protocol (Link Level Discovery Protocol) UniDirectional Link Detection (UDLD) Link Aggregation Control Protocol (LACS) Cisco Discovery Protocol (CDP) STP-protocol (Spanning Tree Protocol)
WK_CPU_Q_FORUS_TRAFFIC(2)	Host zoals Telnet, Pingv4 en Pingv6 en SNMP Keepalive/loopback-detectie

CPU-wachtrijen (wachtrij nr.)	Functie(s)
	Initiate-Internet Key Exchange (IKE)-protocol (IPSec)
WK_CPU_Q_ICMP_GEN(3)	ICMP - onbereikbare bestemming ICMP-TTL verlopen
WK_CPU_Q_ROUTING_CONTROL(4)	Routing Information Protocol, versie 1 (RIPv1) RIPv2 Interior Gateway Routing Protocol (IGRP) BGP-protocol (border gateway protocol) PIM-UDP Virtual Router Redundancy Protocol (VRRP) Hot Standby Router Protocol, versie 1 (HSRPv1) HSRPv2 Gateway-taakverdeling (GLBP) Label Distribution Protocol (LDP) Web Cache Communication Protocol (WCCP) Routing Information Protocol, volgende generatie (RIPng) Open eerst het kortste pad (OSPF) Open Shortest Path First versie 3(OSPFv3) Enhanced Interior Gateway Routing Protocol (EIGRP) Enhanced Interior Gateway Routing Protocol, versie 6 (EIGRPv6)

CPU-wachtrijen (wachtrij nr.)	Functie(s)
	DHCPv6-software Protocolonafhankelijke multicast (PIM) Protocol Independent Multicast, versie 6 (PIMv6) Hot Standby Router Protocol, volgende generatie (HSRPng) IPv6-besturing Generic Routing Encapsulation (GRE) keepalive NAT-punt (Network Address Translation) Intermediate System-to-Intermediate System (IS-IS)
WK_CPU_Q_FORUS_ADDR_Resolution(5)	Adresoplossingsprotocol (ARP) IPv6-buuradvertentie en buurverzoek
WK_CPU_Q_ICMP_REDIRECT(6)	ICMP-omleiding (Internet Control Message Protocol)
WK_CPU_Q_INTER_FED_TRAFFIC(7)	Layer 2-brugdomein voor interne communicatie.
WK_CPU_Q_L2_LVX_CONT_PACK(8)	Exchange-id (XID)-pakket
WK_CPU_Q_EWLC_CONTROL(9)	Ingesloten draadloze controller (WLC) [Control and Provisioning of Wireless Access points (CAPWAP) (UDP 5246)]
WK_CPU_Q_EWLC_DATA(10)	WLC-gegevenspakket (CAPWAP DATA, UDP 5247)
WK_CPU_Q_L2_LVX_DATA_PACK(11)	Onbekend unicastpakket gestraft voor kaartverzoek.

CPU-wachtrijen (wachtrij nr.)	Functie(s)
WK_CPU_Q_BROADCAST(12)	Alle uitzendingstypen
WK_CPU_Q_OPENFLOW(13)	Leercacheoverflow (Layer 2 + Layer 3)
WK_CPU_Q_CONTROLLER_PUNT(14)	Gegevens - toegangscontrolelijst (ACL) Volledig Gegevens - IPv4-opties Gegevens - IPv6 hop-by-hop Gegevens - out-of-resources/catch all Gegevens - onvolledig omgekeerd Path Forwarding (RPF) Glean-pakket
WK_CPU_Q_TOPOLOGY_CONTROL(15)	STP-protocol (Spanning Tree Protocol) Resilient Ethernet Protocol (REP) Gedeeld Spanning Tree Protocol (SSTP)
WK_CPU_Q_PROTO_SNOOPING(16)	ARP-snooping (Address Resolution Protocol) voor dynamische ARP-inspectie (DAI)
WK_CPU_Q_DHCP_SNOOPING(17)	DHCP-controle
WK_CPU_Q_TRANSIT_TRAFFIC(18)	Dit wordt gebruikt voor pakketten die door NAT worden gepunteerd, die in de softwarepad moeten worden verwerkt.
WK_CPU_Q_RPF_MISLUKT(19)	Gegevens - mRPF (multicast RPF) mislukt
WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	IGMP- (Internet Group Management Protocol)/MLD-controle (Multicast Listener Discovery)

CPU-wachtrijen (wachtrij nr.)	Functie(s)
WK_CPU_Q_LOGGING(21)	Vastlegging toegangscontrolelijst (ACL)
WK_CPU_Q_PUNT_WEBAUTH(22)	Web verificatie
WK_CPU_Q_HIGH_RATE_APP(23)	uitzenden
WK_CPU_Q_UITZONDERING(24)	IKE-indicatie IP-leerschending IP-poortbeveiligingsschending Statische IP-adresschending IPv6-toepassingscontrole Remote Copy Protocol (RCP)-uitzondering Unicast RPF mislukt
WK_CPU_Q_SYSTEM_CRITICAL(25)	Media-signalering/draadloze proxy-ARP
WK_CPU_Q_NFL_STEEKPROEF_GEGEVENS(26)	NetFlow-proxy voor data- en mediaservices (MSP)
WK_CPU_Q_LOW_LATENCY(27)	BFD-detectie (Bidirectional Forwarding Detection), precisie-tijdprotocol (PTP)
WK_CPU_Q_EGR_EXCEPTION(28)	Uitgangs-resolutie-uitzondering
WK_CPU_Q_STACKWISE_VIRTUAL_CONTROL(29)	Stapelprotocollen voorzijde, namelijk SVL
WK_CPU_Q_MCAST_DATA(30)	Gegevens - (S,G) aanmaak Gegevens - lokale verbindingen Gegevens - PIM-registratie Data - NBP-overschakeling

CPU-wachtrijen (wachtrij nr.)	Functie(s)
	Gegevens - multicast
WK_CPU_Q_GOLD_PKT(31)	Goud

Standaardbeleid

Standaard wordt het door het systeem gegenereerde CoPP-beleid toegepast op het punt/injectiepad. Het standaardbeleid kan worden bekeken met behulp van gemeenschappelijke op MQC gebaseerde opdrachten. Het beeld kan ook worden bekeken in de switch. Het enige beleid dat mag worden toegepast op het in- of uitstappen van de CPU/besturingsplane is het door het systeem gedefinieerde beleid.

Gebruik "toon beleidskaart-besturingsplane" om het op het bedieningsvlak toegepaste beleid te bekijken:

<#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
```

CoPP aanpassen

CoPP-policersnelheden kunnen door de gebruiker worden geconfigureerd. Gebruikers hebben ook de mogelijkheid om wachtrijen uit te schakelen.

Dit voorbeeld laat zien hoe je een individuele policer waarde aanpast. In dit voorbeeld, de aangepaste klasse is "systeem-cpp-politie-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
```

Dit voorbeeld laat zien hoe je een wachtrij volledig moet uitschakelen. Gebruik voorzichtigheid bij het uitschakelen van wachtrijen, aangezien dit kan leiden tot mogelijke oververzadiging van de 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
```

Problemen oplossen

Methodologie

Het CPU-gebruik wordt beïnvloed door twee basisprocessen en -onderbrekingen. Processen zijn gestructureerde activiteiten die de CPU uitvoert terwijl interruptie verwijst naar pakketten die op het dataplane worden onderschept en naar de CPU worden gestuurd voor actie. Samen omvatten deze activiteiten het totale gebruik van de CPU. Aangezien CoPP standaard is ingeschakeld, is een service-effect niet noodzakelijkerwijs gecorreleerd met een hoog CPU-gebruik. Als CoPP zijn werk doet, wordt het gebruik van cpu niet zeer beïnvloed. Het is belangrijk om het algemene gebruik van de CPU te overwegen, maar het algemene gebruik vertelt niet het hele verhaal. De show commando's en hulpprogramma's in deze sectie worden gebruikt om snel de status van de CPU te beoordelen en relevante details te identificeren over CPU-gebonden verkeer.

Richtsnoeren:

- Bepaal of het probleem verband houdt met het bedieningsvlak. Het meeste transitoverkeer wordt doorgestuurd in hardware. Alleen bij bepaalde verkeerstypen en bepaalde scenario's zijn de CPU en de besturingsplane betrokken, dus houd dit tijdens het onderzoek in gedachten.
- Begrijp uw benuttingsbasislijn. Het is belangrijk om te begrijpen hoe normaal gebruik eruit ziet, zodat afwijkingen van de norm kunnen worden geïdentificeerd.
- Bevestig het algemene gebruik voor zowel processen als onderbrekingen. Identificeer processen die onverwachte volumes van CPU-cycli opnemen. Als het gebruik buiten het verwachte bereik valt, is dit mogelijk reden tot zorg. Het is belangrijk om het gemiddelde gebruik voor een systeem te begrijpen, zodat afwijkingen die niet aan de norm voldoen, worden erkend. Houd in gedachten dat gebruik alleen geen volledig beeld van de gezondheid van het controlevliegtuig is.
- Bepaal of er actief toenemende druppels in CoPP zijn. CoPP druppels zijn niet altijd indicatief voor een probleem, maar als u problemen oplossen met betrekking tot een verkeersklasse die actief wordt gecontroleerd, is dit een sterke indicator van relevantie.

Handige opdrachten weergeven

De switch maakt snel toezicht op CPU-gezondheids- en CoPP-statistieken mogelijk. Er is ook nuttige CLI om snel het ingangspunt van CPU-gebonden verkeer te bepalen.

Bepaal het algemene en historische gebruik

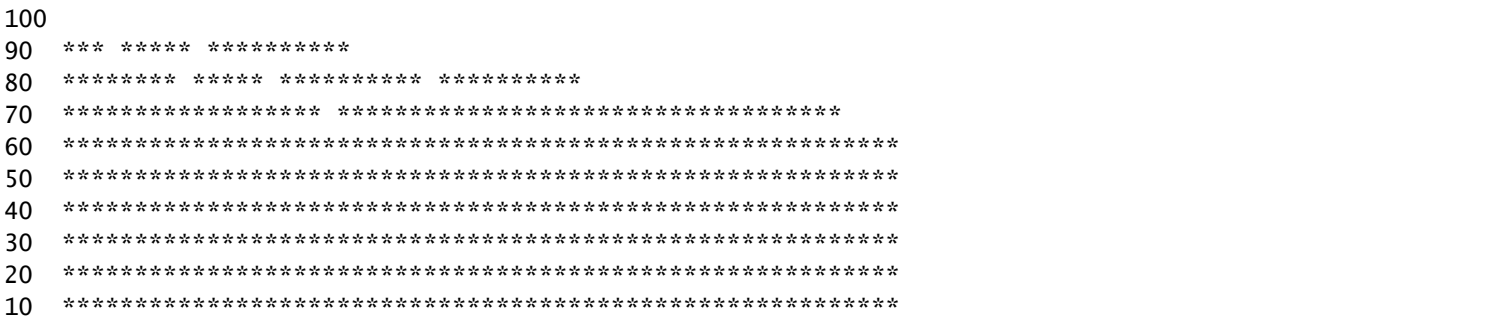
- "Toon processen cpu gesorteerd" wordt gebruikt om het algemene CPU gebruik te bekijken. Het "gesorteerde" argument sorteert procesoutput op basis van percentage van gebruik. Processen die gebruik maken van meer CPU bronnen bevinden zich aan de top van de output. Gebruik als gevolg van onderbrekingen wordt ook als percentage verstrekt.

```
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
```

- "Toon processen cpu geschiedenis" biedt een historische grafiek van CPU-gebruik over de laatste 60 seconden, 5 minuten en 72 uur.

<#root>

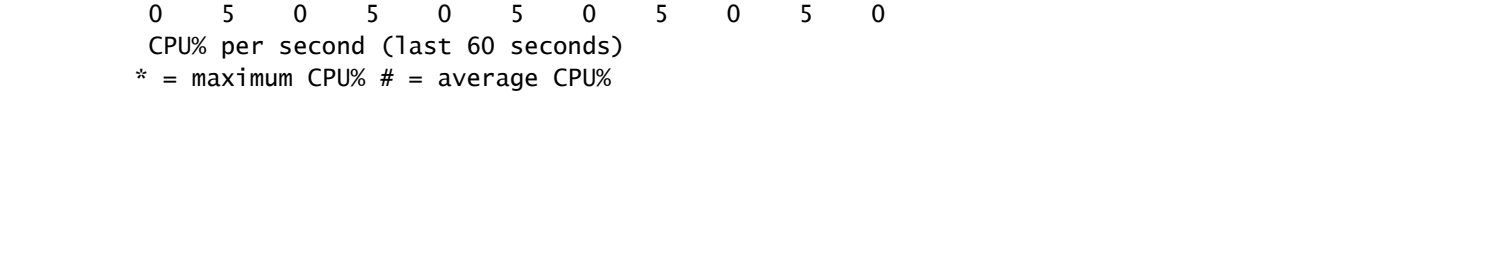
```
Catalyst-9600#
show processes cpu history

    9997777766666888886666677777777778888877777666669999998888866
<<<--- 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.
```

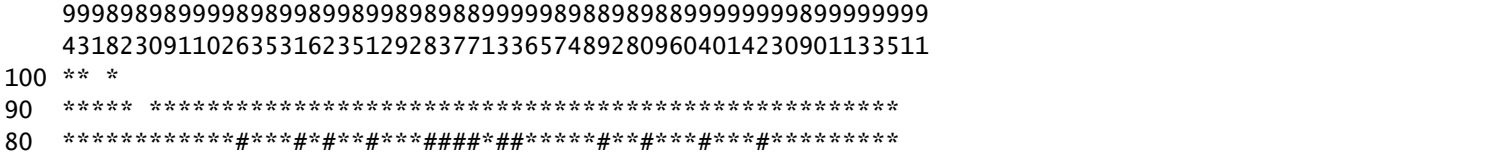


<<<--- The "*" represents the highest value during the given time period. This relates to a momentary sp

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.





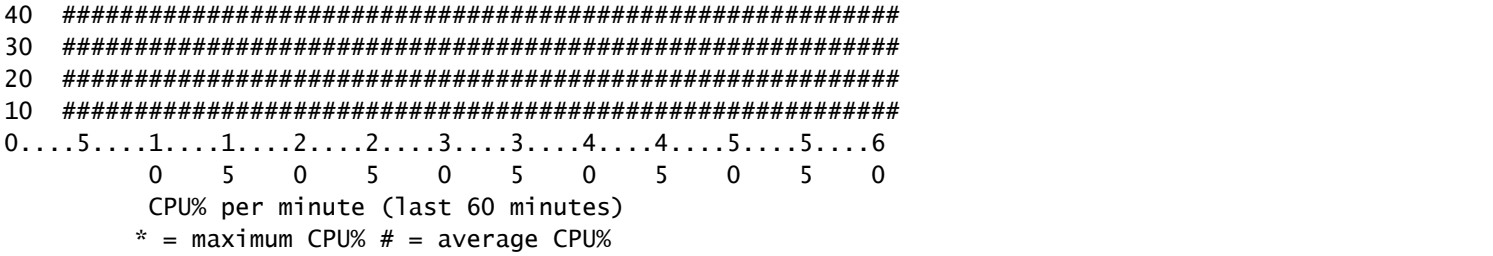
<<<--- The "#" represents the average utilization. This indicates sustained utilization.

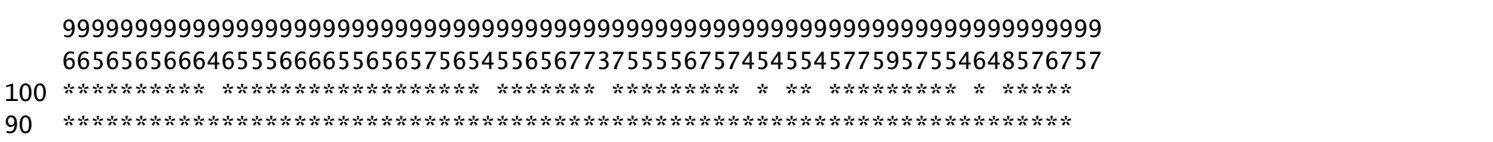
60 ##

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

50 ##

the maximum utilization spiked to 94%.





```
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%
```

Toezicht op besturingsplane controleren

- Gebruik "show platform hardware gevoed <switch> actieve qos wachtrij stats internal cpu policer" om geaggregeerde CoPP-statistieken en aanvullende informatie over wachtrij/policer structuur te bekijken. Deze uitgang geeft een historische weergave van politiestatistieken sinds de laatste reset van het besturingsplane. Deze tellers zijn ook handmatig leesbaar. Over het algemeen wijst het bewijs van regelvliegtuig-druppels door politieman op een probleem met de bijbehorende rij/klasse maar zorg ervoor dat druppels actief verhogen terwijl het probleem zich voordoet. Voer de opdracht meerdere malen uit om te observeren voor het verhogen van Queue Drop-waarden.

<#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/
```

Verzamel informatie over bestraft verkeer

Deze opdrachten worden gebruikt om informatie te verzamelen over verkeer dat naar de CPU is geprikt, waaronder het type verkeer en de fysieke ingangspunten.

- "Toon platformsoftware gevoed <switch> actief punt cpuq all" of "Toon platformsoftware gevoed <switch> actief punt cpuq <0-31 Queue ID>" kan worden gebruikt om statistieken te zien met betrekking tot alle of een specifieke CPU wachtrij.

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

- Gebruik "toon platformsoftware gevoed <switch> actief punt oorzaak samenvatting" voor een snelle blik op alle verschillende verkeerstypes die bij CPU zijn gezien. Merk op dat alleen non-zero oorzaken worden getoond.

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

- Gebruik de opdracht "toon platform software gevoed <switch> actieve punt rates interfaces" om snel de interfaces CPU-gebonden verkeer te bekijken betreedt het systeem. Deze opdracht toont alleen interfaces met een niet-nul invoerwachtrij.

<#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

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

- Gebruik "Toon platformsoftware gevoed <switch> actieve punttarieven interfaces <IF-ID>" om de individuele wachtrijen van de interface af te boren en te bekijken. Deze opdracht toont geaggregeerde statistieken en kan worden gebruikt om historische activiteit in de inputwachtrij te bekijken en als verkeer is bewaakt.

<#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

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

Inspecteer CPU-gebonden verkeer

Catalyst 9000 reeks switches biedt hulpprogramma's om CPU-gebonden verkeer te controleren en weer te geven. Gebruik deze gereedschappen om te begrijpen welk verkeer actief naar de CPU wordt gestraft.

Ingesloten pakketvastlegging (EPC)

De EPC op het bedieningsvlak kan in beide richtingen (of beide) worden uitgevoerd. Voor bestraft verkeer, vangt inkomende. EPC op het besturingsplane kan worden opgeslagen als buffer of bestand.

<#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
```

De opnameresultaten kunnen in korte of gedetailleerde uitvoer worden bekeken.

<#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
```

De opnameresultaten kunnen rechtstreeks naar het bestand worden geschreven of uit de buffer worden geëxporteerd.

<#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#
```

FED CPU pakketvastlegging

De Catalyst 9000-reeks switches ondersteunt een debug-hulpprogramma dat een verbeterde zichtbaarheid van pakketten van en naar de CPU mogelijk maakt.

```
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)
```

Bufferinhoud heeft korte en gedetailleerde opties voor uitvoer.

<#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
```

Veel weergavefilters zijn beschikbaar voor gebruik. De meest voorkomende Wireshark weergavefilters worden ondersteund.

<#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>
```

Filters kunnen ook als opnamefilters worden toegepast.

```
<#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"
```

Gemeenschappelijke scenario's

Intermitterend ICMP-verlies (Ping) voor lokaal IP

Verkeer dat wordt doorgestuurd naar een lokaal IP op een switch wordt gestraft in de Forus-wachtrij (letterlijk "voor ons"). De toename in de Forus CoPP-wachtrij zien heeft betrekking op gedropte pakketten die bestemd zijn voor de lokale switch. Dit is relatief simpel en gemakkelijk te conceptualiseren.

In sommige omstandigheden, echter, kan er verlies aan plaatselijk bestemd verkeer dat niet netjes correleert met Forus druppels.

Met voldoende cpu-gebonden verkeersstroom wordt het puntpad oververzadigd voorbij de mogelijkheid van CoPP om prioriteit te geven aan welk verkeer wordt gecontroleerd. Het verkeer wordt 'stilletjes' gecontroleerd op een first-in, first-out basis.

In dit scenario zijn er aanwijzingen dat de besturingsplanning in grote aantallen wordt uitgevoerd, maar het verkeerstype van belang (Forus in dit voorbeeld) neemt niet per se actief toe.

Samengevat, als er een uitzonderlijk hoog volume van CPU-gebonden verkeer is, zoals blijkt uit zowel actief CoPP-toezicht als aangetoond met een pakketopname of FED punt debug, zou er verlies kunnen zijn dat niet uitgelijnd is op de wachtrij die u probleemoplossing bent. Bepaal in dit scenario waarom er een buitensporige hoeveelheid CPU-gebonden verkeer is en neem maatregelen om de belasting op het besturingsplane te verlichten.

Hoge ICMP-omleidingen en langzame DHCP-werking

CoPP op de Catalyst 9000 Series switch is georganiseerd in 32 hardwarerijen. Deze 32 hardware wachtrijen worden uitgelijnd op 20 individuele policer indices. Elke policer index correleert met een of meer hardware wachtrijen.

Functioneel betekent dit dat meerdere verkeersklassen een policer index delen en onderworpen zijn aan een gemeenschappelijke samengestelde policer waarde.

Een gemeenschappelijk probleem dat op switches met toegelaten DHCP-relay-agents wordt gezien, heeft een trage DHCP-respons. Clients kunnen IP's sporadisch krijgen, maar het kost verschillende pogingen om te voltooien en sommige klanten tijd uit.

De ICMP-omleidingswachtrij en de Broadcast-wachtrij delen een policer-index, zodat een groot verkeersvolume dat wordt ontvangen op en gerouteerd vanuit dezelfde Switch Virtual Interface (SVI)-toepassingen beïnvloedt die afhankelijk zijn van uitzendverkeer. Dit is vooral opmerkelijk wanneer de switch als relay agent fungeert.

Dit document biedt een diepgaande uitleg van het concept en hoe u dit kunt verhelpen: DHCP-problemen oplossen op Catalyst 9000 DHCP Relay Agents

Aanvullende bronnen

Probleemoplossing voor langzame of intermitterende DHCP op Catalyst 9000 DHCP Relay-agents

Configureer FED CPU-pakketvastlegging op Catalyst 9000 Switches

Catalyst 9300 Switches: controle-vliegtuig configureren

Packet Capture configureren - Configuratiehandleiding voor netwerkbeheer, Cisco IOS XE Bengaluru 17.6.x (Catalyst 9300 Switches)

DHCP-controle en probleemoplossing op Catalyst 9000 Switches

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.