Fehlerbehebung beim Betrieb der Kontrollebene auf Catalyst Switches der Serie 9000

Inhalt

Einleitung

In diesem Dokument wird die Fehlerbehebung und Validierung des Zustands der Kontrollebene auf Catalyst Switches der Serie 9000 mit Cisco IOS® XE beschrieben.

Hintergrundinformationen

Die primäre Aufgabe eines Switches ist die möglichst schnelle Weiterleitung von Paketen. Die meisten Pakete werden in der Hardware weitergeleitet, aber bestimmte Arten von Datenverkehr müssen von der System-CPU verarbeitet werden. An der CPU ankommender Datenverkehr wird so schnell wie möglich verarbeitet. Es wird erwartet, dass die CPU eine bestimmte Menge an Datenverkehr verarbeitet, aber eine Überkapazität führt zu Betriebsproblemen. Die Catalyst Switches der Serie 9000 verfügen standardmäßig über ein robustes CoPP-Verfahren (Control Plane Policing), um Probleme zu vermeiden, die durch eine Übersättigung des CPU-Datenverkehrs verursacht werden.

Unerwartete Probleme treten in bestimmten Anwendungsfällen in Abhängigkeit vom

Normalbetrieb auf. Der Zusammenhang zwischen Ursache und Wirkung ist manchmal nicht offensichtlich, was die Herangehensweise an das Problem erschwert. In diesem Dokument finden Sie Tools zur Überprüfung des Zustands der Kontrollebene sowie einen Workflow zum Umgang mit Problemen, die das Einsetzen oder Einschleusen des Pfades der Kontrollebene betreffen. Darüber hinaus werden mehrere gängige Szenarien basierend auf den in der Praxis festgestellten Problemen vorgestellt.

Beachten Sie, dass der Pfad für CPU-Punt eine begrenzte Ressource ist. Moderne Hardware-Forwarding-Switches können ein exponentiell höheres Datenverkehrsvolumen verarbeiten. Die Catalyst Switches der Serie 9000 unterstützen jeweils ca. 19.000 Pakete pro Sekunde (pps) an der CPU. Überschreiten Sie diesen Schwellenwert, und der blockierte Datenverkehr wird ohne Gewichtung überwacht.

Terminologie

- Forwarding Engine Driver (FED): Dieser Treiber bildet das Herzstück des Cisco Catalyst Switches und ist für die gesamte Hardware-Programmierung/-Weiterleitung zuständig.
- IOSd: Dies ist der Cisco IOS-Daemon, der auf dem Linux-Kernel ausgeführt wird. Es wird als ein Software-Prozess innerhalb des Kernels ausgeführt
- Packet Delivery System (PDS): Dies ist die Architektur und der Prozess für die Zustellung von Paketen zu und von den verschiedenen Subsystemen. So wird beispielsweise gesteuert, wie Pakete vom FED an das IOSd und umgekehrt übermittelt werden
- Kontrollebene (CP): Die Kontrollebene ist ein allgemeiner Begriff, der verwendet wird, um die Funktionen und den Datenverkehr, die die CPU des Catalyst Switches betreffen, zusammenzufassen. Dazu gehören Datenverkehr wie Spanning Tree Protocol (STP), Hot Standby Router Protocol (HSRP) und Routing-Protokolle, die für den Switch bestimmt sind oder vom Switch gesendet werden. Dazu gehören auch Protokolle auf Anwendungsebene wie Secure Shell (SSH) und Simple Network Management Protocol (SNMP), die von der CPU verarbeitet werden müssen
- Datenebene (DP): Die Datenebene umfasst in der Regel die Hardware-ASICs und den Datenverkehr, der ohne Unterstützung durch die Kontrollebene weitergeleitet wird.
- Punt:Eingangsprotokoll-Steuerungspaket, das auf DP abgefangen wurde und zur Verarbeitung an den CP gesendet wurde
- Inject (Einspeisen): Von CP generiertes Protokollpaket wird an DP gesendet, um an E/A-Schnittstelle(n) auszutreten
- LSMPI:Punt-Schnittstelle für gemeinsamen Linux-Speicher

CoPP bei Catalyst 9000

Die Grundlage für den CPU-Schutz auf den Catalyst Switches der Serie 9000 ist CoPP. Bei CoPP wird eine vom System generierte QoS-Richtlinie auf den Einfügepfad der CPU angewendet. CPU-gebundener Datenverkehr wird in viele verschiedene Klassen eingeteilt und anschließend den einzelnen Hardware-Richtlinien zugeordnet, die der CPU zugeordnet sind. Die Richtlinien verhindern eine Übersättigung der CPU durch eine bestimmte Datenverkehrsklasse.

CoPP-Implementierung

Der CPU-gebundene Datenverkehr wird in Warteschlangen klassifiziert. Diese Warteschlangen/Klassen sind vom System definiert und können nicht vom Benutzer konfiguriert werden. Policers werden in der Hardware konfiguriert. Die Catalyst Serie 9000 unterstützt 32 Hardware-Richtlinien für 32 Warteschlangen.

Bestimmte Werte unterscheiden sich von Plattform zu Plattform. Im Allgemeinen gibt es 32 systemdefinierte Warteschlangen. Diese Warteschlangen beziehen sich auf Klassenzuordnungen, die sich auf Policer-Indizes beziehen. Die Policer-Indizes weisen eine Standardkontrollrate auf. Diese Rate kann vom Benutzer konfiguriert werden, Änderungen an der CoPP-Standardrichtlinie erhöhen jedoch die Anfälligkeit für unerwartete Auswirkungen auf Services.

Systemdefinierte Werte für CoPP

Klassenzuordnungsnamen	Policer-Index (Policer-Nr.)	CPU-Warteschlangen (W
system-cpp- polizeidaten	WK_CPP_POLICE_DATA(0)	WK_CPU_Q_ICMP_GEN(3 WK_CPU_Q_BROADCAST WK_CPU_Q_ICMP_REDIR
system-cpp-police-l2-control	WK_CPP_POLICE_L2_ CONTROL(1)	WK_CPU_Q_L2_CONTROL
system-cpp-police-routing-control	WK_CPP_POLICE_ROUTING_CONTROL(2)	WK_CPU_Q_ROUTING_CO WK_CPU_Q_LOW_LATEN
system-cpp-police-control-low-priority	WK_CPP_POLICE_CO STEUERUNG_LOW_PRI(3)	WK_CPU_Q_GENERAL_PU
system-cpp-police-punt-webauth	WK_CPP_POLICE_PU NT_WEBAUTH(7)	WK_CPU_Q_PUNT_WEBA
system-cpp-police-Topologiesteuerung	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_
system-cpp-police-multicast	WK_CPP_POLICE_MULTICAST(9)	WK_CPU_Q_TRANSIT_TR WK_CPU_Q_MCAST_DATA
system-cpp-police-sys-data	WK_CPP_POLICE_SYS_DATA(10)	WK_CPU_Q_LEARNING_C

Klassenzuordnungsnamen	Policer-Index (Policer-Nr.)	CPU-Warteschlangen (W
		WK_CPU_Q_CRYPTO_CO WK_CPU_Q_EXCEPTION(WK_CPU_Q_EGR_EXCEP WK_CPU_Q_NFL_SAMPLE WK_CPU_Q_GOLD_PKT(3 WK_CPU_Q_RPF_FAILED(
system-cpp-police-dot1x-auth	WK_CPP_POLICE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH
system-cpp-police-protocol-snooping	WK_CPP_POLICE_PR(12)	WK_CPU_Q_PROTO_SNO
system-cpp-police-sw-forward	WK_CPP_POLICE_SW_FWD (13)	WK_CPU_Q_SW_FORWAR WK_CPU_Q_LOGGING(21 WK_CPU_Q_L2_LVX_DATA
system-cpp-police-forus	WK_CPP_POLICE_FORUS(14)	WK_CPU_Q_FORUS_ADD WK_CPU_Q_FORUS_TRA
system-cpp-police-Multicast-Endstation	WK_CPP_POLICE_MULTICAST_SNOOPING(15)	WK_CPU_Q_MCAST_END TION_SERVICE(20)
system-cpp-default	WK_CPP_POLICE_DEFAULT_POLICER(16)	WK_CPU_Q_DHCP_SNOO WK_CPU_Q_UNUSED(7) WK_CPU_Q_EWLC_CONT WK_CPU_Q_EWLC_DATA
system-cpp-police-stackwise-virt-control	WK_CPP_STACKWISE_VIRTUAL_CONTROL(5)	WK_CPU_Q_STACKWISE_ (29)
system-cpp-police-l2lvx-	WK_CPP_ L2_LVX_CONT_PACK(4)	WK_CPU_Q_L2_LVX_CON

Klassenzuordnungsnamen	Policer-Index (Policer-Nr.)	CPU-Warteschlangen (W
control		

Jede Warteschlange bezieht sich auf einen Datenverkehrstyp oder bestimmte Funktionen. Diese Liste ist nicht vollständig:

CPU-Warteschlangen und zugehörige Funktionen

CPU-Warteschlangen (Warteschlangennr.)	Funktion/en
WK_CPU_Q_DOT1X_AUTH(0)	IEEE 802.1x Portbasierte Authentifizierung
WK_CPU_Q_L2_CONTROL(1)	Dynamic Trunking Protocol (DTP) VLAN Trunking Protocol (VTP) Port Aggregation Protocol (PAgP) Client Information Signaling Protocol (CISP) Relaisprotokoll für Nachrichtensitzung Multiple VLAN Registration Protocol (MVRP) Metropolitan Mobile Network (MMN) LLDP (Link Level Discovery Protocol) Unidirectional Link Detection (UDLD) Link Aggregation Control Protocol (LACP) Cisco Discovery Protocol (CDP) Spanning Tree Protocol (STP)
WK_CPU_Q_FORUS_TRAFFIC(2)	Hosts wie Telnet, Pingv4 und Pingv6 und SNMP Keepalive-/Loopback-Erkennung Initiate-Internet Key Exchange (IKE)-Protokoll (IPSec)

CPU-Warteschlangen (Warteschlangennr.)	Funktion/en
WK_CPU_Q_ICMP_GEN(3)	ICMP - Ziel nicht erreichbar ICMP-TTL abgelaufen
WK_CPU_Q_ROUTING_CONTROL(4)	Routing Information Protocol Version 1 (RIPv1) RIPv2 Interior Gateway Routing Protocol (IGRP) Border Gateway Protocol (BGP) PIM-UDP Virtual Router Redundancy Protocol (VRRP) Hot Standby Router Protocol Version 1 (HSRPv1) HSRPv2 Gateway Load Balancing Protocol (GLBP) Label Distribution Protocol (LDP) Web Cache Communication Protocol (WCCP) Routing Information Protocol Next Generation (RIPng) Open Shortest Path First (OSPF) Open Shortest Path First Version 3 (OSPFv3) EIGRP (Enhanced Interior Gateway Routing Protocol) Enhanced Interior Gateway Routing Protocol Version 6 (EIGRPv6) DHCPv6 Protocol Independent Multicast (PIM)

CPU-Warteschlangen (Warteschlangennr.)	Funktion/en
	Protocol Independent Multicast Version 6 (PIMv6) Hot Standby Router Protocol Next Generation (HSRPng) IPv6-Kontrolle Generic Routing Encapsulation (GRE)-Keepalive Network Address Translation (NAT) Punt Intermediate System-to-Intermediate System (IS-IS)
WK_CPU_Q_FORUS_ADDR_RESOLUTION(5)	Address Resolution Protocol (ARP) IPv6-Nachbar-Werbung und Nachbar-Ausschreibung
WK_CPU_Q_ICMP_REDIRECT(6)	Internet Control Message Protocol (ICMP)-Umleitung
WK_CPU_Q_INTER_FED_TRAFFIC(7)	Layer-2-Bridge-Domäneninjektion für interne Kommunikation.
WK_CPU_Q_L2_LVX_CONT_PACK(8)	Exchange ID (XID)-Paket
WK_CPU_Q_EWLC_CONTROL(9)	Integrierter Wireless Controller (eWLC) [Control and Provisioning of Wireless Access Points (CAPWAP) (UDP 5246)]
WK_CPU_Q_EWLC_DATA(10)	eWLC-Datenpaket (CAPWAP DATA, UDP 5247)
WK_CPU_Q_L2_LVX_DATA_PACK(11)	Unbekanntes Unicast-Paket für Kartenanforderung analysiert.
WK_CPU_Q_BROADCAST(12)	Alle Arten von Broadcast

CPU-Warteschlangen (Warteschlangennr.)	Funktion/en
WK_CPU_Q_OPENFLOW(13)	Lerncache-Überlauf (Layer 2 + Layer 3)
WK_CPU_Q_CONTROLLER_PUNT(14)	Daten - Zugriffskontrollliste (ACL) voll Daten - IPv4-Optionen Daten - IPv6 Hop-by-Hop Daten - keine Ressourcen/keine Erfassung Daten - Reverse Path Forwarding (RPF) unvollständig Glean Packet
WK_CPU_Q_TOPOLOGY_CONTROL(15)	Spanning Tree Protocol (STP) Resilient Ethernet Protocol (REP) Shared Spanning Tree Protocol (SSTP)
WK_CPU_Q_PROTO_SNOOPING(16)	Address Resolution Protocol (ARP) Snooping für Dynamic ARP Inspection (DAI)
WK_CPU_Q_DHCP_SNOOPING(17)	DHCP-Snooping
WK_CPU_Q_TRANSIT_TRAFFIC(18)	Dies wird für Pakete verwendet, die von NAT analysiert werden und im Softwarepfad verarbeitet werden müssen.
WK_CPU_Q_RPF_FAILED(19)	Daten - mRPF (Multicast RPF) fehlgeschlagen
WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	Internet Group Management Protocol (IGMP)-/Multicast Listener Discovery (MLD)-Steuerung
WK_CPU_Q_LOGGING(21)	Protokollierung von Zugriffskontrolllisten (ACL)

CPU-Warteschlangen (Warteschlangennr.)	Funktion/en
WK_CPU_Q_PUNT_WEBAUTH(22)	Webauthentifizierung
WK_CPU_Q_HIGH_RATE_APP(23)	Senden
WK_CPU_Q_EXCEPTION(24)	IKE-Anzeige IP-Lernverletzung IP-Port-Sicherheitsverletzung IP-Adressverletzung IPv6-Bereichsprüfung Remote Copy Protocol (RCP)-Ausnahme Unicast-RPF fehlgeschlagen
WK_CPU_Q_SYSTEM_CRITICAL(25)	Mediensignalisierung/Wireless-Proxy-ARP
WK_CPU_Q_NFL_SAMPLED_DATA(26)	Von NetFlow erfasste Daten und Media Services Proxy (MSP)
WK_CPU_Q_LOW_LATENCY(27)	Bidirectional Forwarding Detection (BFD), Precision Time Protocol (PTP)
WK_CPU_Q_EGR_EXCEPTION(28)	Ausnahme bei Egress-Auflösung
WK_CPU_Q_STACKWISE_VIRTUAL_CONTROL(29)	Front-Side-Stacking-Protokolle, insbesondere SVL
WK_CPU_Q_MCAST_DATA(30)	Daten - (S,G) Erstellung Daten - lokale Joins Daten - PIM-Registrierung Daten - SPT-Switchover Daten - Multicast

CPU-Warteschlangen (Warteschlangennr.)	Funktion/en
WK_CPU_Q_GOLD_PKT(31)	Gold

Standardrichtlinie

Standardmäßig wird die vom System generierte CoPP-Richtlinie auf den Einfügepfad angewendet. Die Standardrichtlinie kann mithilfe allgemeiner MQC-basierter Befehle angezeigt werden. Er ist auch in der Switch-Konfiguration zu sehen. Die einzige Richtlinie, die beim Ein- oder Ausgang der CPU/Kontrollebene angewendet werden darf, ist die vom System definierte Richtlinie.

Verwenden Sie "show policy-map control-plane", um die auf die Kontrollebene angewendete Richtlinie anzuzeigen:

<#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 anpassen

Die CoPP-Policer-Raten können vom Benutzer konfiguriert werden. Benutzer können auch Warteschlangen deaktivieren.

In diesem Beispiel wird veranschaulicht, wie ein einzelner Policer-Wert angepasst wird. In diesem Beispiel lautet die angepasste Klasse "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 diesem Beispiel wird veranschaulicht, wie eine Warteschlange vollständig deaktiviert wird. Gehen Sie beim Deaktivieren von Warteschlangen vorsichtig vor, da dies zu einer möglichen Überlastung der CPU führen kann.

<#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
```

Fehlerbehebung

Methodik

Die CPU-Auslastung wird durch zwei grundlegende Aktivitäten beeinträchtigt: Prozesse und Unterbrechung. Prozesse sind strukturierte Aktivitäten, die die CPU ausführt, während sich die Unterbrechung auf Pakete bezieht, die auf dem Datenflugzeug abgefangen und zur Aktion an die CPU gesendet werden. Zusammen umfassen diese Aktivitäten die Gesamtnutzung der CPU. Da CoPP standardmäßig aktiviert ist, sind die Auswirkungen auf Services nicht notwendigerweise mit einer hohen CPU-Auslastung verbunden. Wenn CoPP seine Aufgabe erfüllt, hat dies keine großen Auswirkungen auf die CPU-Auslastung. Es ist wichtig, die Gesamtnutzung der CPU zu berücksichtigen, aber die Gesamtnutzung erzählt nicht die ganze Geschichte. Die Befehle und Dienstprogramme show in diesem Abschnitt dienen dazu, den Zustand der CPU schnell zu bewerten und relevante Details zum CPU-gebundenen Datenverkehr zu identifizieren.

Richtlinien:

- Prüfen Sie, ob sich das Problem auf die Kontrollebene bezieht. Der Großteil des Datenverkehrs wird in der Hardware weitergeleitet. Nur bestimmte Datenverkehrstypen und bestimmte Szenarien betreffen die CPU und die Steuerungsebene. Denken Sie daher bei der Untersuchung immer daran.
- Grundlegende Informationen zur Nutzung Es ist wichtig zu wissen, wie die normale Auslastung aussieht, damit Abweichungen von der Norm identifiziert werden können.
- Validierung der Gesamtnutzung für Prozesse und Unterbrechungen Identifizieren Sie alle Prozesse, die unerwartete CPU-Zyklen beanspruchen. Fällt die Auslastung außerhalb des erwarteten Bereichs, ist dies potenziell Besorgnis erregend. Es ist wichtig, die durchschnittliche Auslastung eines Systems zu kennen, damit Abweichungen außerhalb der Norm erkannt werden. Beachten Sie, dass die Auslastung allein kein vollständiges Bild des Zustands der Kontrollebene darstellt.
- Stellen Sie fest, ob in CoPP eine inkrementelle Verwerfung stattfindet. CoPP-Verwerfungen weisen nicht immer auf ein Problem hin. Wenn Sie jedoch ein Problem beheben, das mit einer Datenverkehrsklasse zusammenhängt, für die eine aktive Richtlinie festgelegt wurde, ist dies ein guter Indikator für die Relevanz.

Nützliche Show-Befehle

Der Switch bietet einen schnellen Überblick über die CPU-Status und CoPP-Statistiken. Außerdem gibt es eine nützliche CLI, um den Eingangspunkt des CPU-gebundenen Datenverkehrs schnell zu bestimmen.

Bestimmen der Gesamt- und Verlaufsnutzung

- "Prozesse sortiert anzeigen" wird verwendet, um die CPU-Auslastung insgesamt anzuzeigen. Das Argument "sortiert" sortiert die Prozessausgabe basierend auf dem Nutzungsprozentsatz. Prozesse, die mehr CPU-Ressourcen benötigen, stehen an der Spitze der Ausgabe. Die Auslastung aufgrund von Unterbrechungen wird ebenfalls als Prozentsatz

angegeben.

<#root>

```
Catalyst-9600#

show processes cpu sorted

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

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

                                                                                92% refers to the c

                                                                                The 13% value refer

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

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

344   547030523  607054509        901  38.13% 30.61% 29.32%   0  SISF Switcher Th
345   394700227  615024099        641  31.18% 22.68% 21.66%   0  SISF Main Thread
 98   112308516  119818535        937   4.12%  4.76%  5.09%   0  Crimson flush tr
247    47096761   92250875        510   2.42%  2.21%  2.18%   0  Spanning Tree
123    35303496  679878082         51   1.85%  1.88%  1.84%   0  IOSXE-RP Punt Se
234         955       1758        543   1.61%  0.71%  0.23%   3  SSH Process
547     5360168    5484910        977   1.04%  0.46%  0.44%   0  DHCPD Receive
229    27381066  963726156         28   1.04%  1.34%  1.23%   0 IP Input
 79    13183805  108951712        121   0.48%  0.55%  0.55%   0 IOSD ipc task
  9     1073134     315186       3404   0.40%  0.06%  0.03%   0 Check heaps
 37    11099063  147506419         75   0.40%  0.54%  0.52%   0 ARP Input
312     2986160  240782059         12   0.24%  0.12%  0.14%   0 DAI Packet Proce
<snip>
565           0          1          0   0.00%  0.00%  0.00%   0 LICENSE AGENT
566          14       1210         11   0.00%  0.00%  0.00%   0 DHCPD Timer
567          40         45        888   0.00%  0.00%  0.00%   0 OVLD SPA Backgro
568          12       2342          5   0.00%  0.00%  0.00%   0 DHCPD Database
569           0         12          0   0.00%  0.00%  0.00%   0 SpanTree Flush
571           0          1          0   0.00%  0.00%  0.00%   0 EM Action CNS
572         681     140276          4   0.00%  0.00%  0.00%   0 Inline power inc
```

- "Show processes cpu history" (Prozesse-CPU-Verlauf anzeigen) liefert ein historisches Diagramm der CPU-Auslastung über die letzten 60 Sekunden, 5 Minuten und 72 Stunden.

<#root>

```
Catalyst-9600#

show processes cpu history

    999777776666688888666667777777777888887777766666999998888866
```

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

```
    22255555999994444444444000008888888881111177777333335555500
```

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%
```

```
    99989898999989899899899898988999998988989889999999899999999
    43182309110263531623512928377133657489280960401423090113351 1
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%
```

Überprüfen des Control Plane Policing

- Verwenden Sie "show platform hardware fed <switch> active qos queue stats internal cpu policer", um aggregierte CoPP-Statistiken und zusätzliche Informationen zur Warteschlangen-/Policer-Struktur anzuzeigen. Diese Ausgabe bietet eine historische Ansicht der Policer-Statistiken seit dem letzten Zurücksetzen der Kontrollebene. Diese Zähler können auch manuell gelöscht werden. Im Allgemeinen weist der Hinweis auf Verwerfen der Kontrollebene durch die Richtlinie auf ein Problem mit der zugehörigen Warteschlange/Klasse hin, stellt jedoch sicher, dass die Verwerfungen aktiv zunehmen, während das Problem auftritt. Führen Sie den Befehl mehrmals aus, um zu beobachten, ob die Werte für das Ablegen von Warteschlangen erhöht werden.

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

               Second Level Policer Config
=====================================================================
    level-1 level-2                              level-2
QId PlcIdx  PlcIdx  Queue Name                   Enabled
---------------------------------------------------------------------
0    11      21      DOT1X Auth                   Yes
1    1       20      L2 Control                   Yes
2    14      21      Forus traffic                Yes
3    0       21      ICMP GEN                     Yes
4    2       20      Routing Control              Yes
5    14      21      Forus Address resolution     Yes
6    0       21      ICMP Redirect                Yes
7    16      -       Inter FED Traffic            No
8    4       21      L2 LVX Cont Pack             Yes
9    19      -       EWLC Control                 No
10   16      -       EWLC Data                    No
11   13      21      L2 LVX Data Pack             Yes
12   0       21      BROADCAST                    Yes
13   10      21      Openflow                     Yes
14   13      21      Sw forwarding                Yes
15   8       20      Topology Control             Yes
16   12      21      Proto Snooping               Yes
17   6       -       DHCP Snooping                No
18   13      21      Transit Traffic              Yes
19   10      21      RPF Failed                   Yes
20   15      21      MCAST END STATION            Yes
21   13      21      LOGGING                      Yes
22   7       21      Punt Webauth                 Yes
23   18      -       High Rate App                No
24   10      21      Exception                    Yes
25   3       -       System Critical              No
26   10      21      NFL SAMPLED DATA             Yes
27   2       20      Low Latency                  Yes
28   10      21      EGR Exception                Yes
29   5       -       Stackwise Virtual OOB        No
30   9       21      MCAST Data                   Yes
31   3       -       Gold Pkt                     No

                        CPP Classes to queue map

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

PlcIdx CPP Class                                :  Queues
---------------------------------------------------------------------------------------------
0      system-cpp-police-data                   :  ICMP GEN/ BROADCAST/ ICMP Redirect/
10     system-cpp-police-sys-data               :  Openflow/ Exception/ EGR Exception/ NFL SAMPLED DATA/
13     system-cpp-police-sw-forward             :  Sw forwarding/ LOGGING/ L2 LVX Data Pack/ Transit Tra
```

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

Sammeln von Informationen zu unterbrochenem Datenverkehr

Mit diesen Befehlen werden Informationen über den an die CPU gesendeten Datenverkehr gesammelt, einschließlich der Art des Datenverkehrs und der physischen Eingangspunkte.

- "Show platform software fed <switch> active punt cpuq all" oder "Show platform software fed <switch> active punt cpuq <0-31 Queue ID>" können verwendet werden, um Statistiken zu allen oder zu einer bestimmten CPU-Warteschlange anzuzeigen.

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

- Verwenden Sie "show platform software fed <switch> active punt Cause summary", um einen kurzen Überblick über die verschiedenen Datenverkehrstypen zu erhalten, die auf der CPU zu beobachten sind. Beachten Sie, dass nur Nicht-Null-Ursachen angezeigt werden.

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

- Verwenden Sie den Befehl "show platform software fed <switch> active punt rates interfaces", um die Schnittstellen schnell anzuzeigen, an die CPU-gebundener Datenverkehr im System eingeht. Dieser Befehl zeigt nur Schnittstellen mit einer Eingabewarteschlange ungleich null an.

<#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

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

- Verwenden Sie "show platform software fed <switch> active punt rates interfaces <IF-ID>", um Details anzuzeigen und die einzelnen Warteschlangen der Schnittstelle anzuzeigen. Dieser Befehl zeigt aggregierte Statistiken an und kann verwendet werden, um historische Eingangswarteschlangenaktivitäten anzuzeigen und zu überprüfen, ob der Datenverkehr geregelt wurde.

<#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

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

Prüfung von CPU-gebundenem Datenverkehr

Die Catalyst Switches der Serie 9000 bieten Dienstprogramme zur Überwachung und Anzeige von CPU-gebundenem Datenverkehr. Verwenden Sie diese Tools, um zu erfahren, welcher Datenverkehr aktiv an die CPU geleitet wird.

Embedded Packet Capture (EPC)

EPC auf der Kontrollebene kann in beide Richtungen (oder beides) durchgeführt werden. Erfassen Sie eingehenden Datenverkehr für einen begrenzten Zeitraum. EPC auf der Kontrollebene kann als Puffer oder in einer Datei gespeichert werden.

<#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
```

Die Erfassungsergebnisse können entweder in einer kurzen oder in einer detaillierten Ausgabe angezeigt werden.

<#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: 000000000000000000000000000000
    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
```

Die Erfassungsergebnisse können entweder direkt in eine Datei geschrieben oder aus einem Puffer exportiert werden.

```
<#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-Paketerfassung

Die Catalyst Switches der Serie 9000 unterstützen ein Debug-Dienstprogramm, das eine verbesserte Transparenz von Paketen zur und von der CPU ermöglicht.

```
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)
```

Pufferinhalte haben kurze und detaillierte Ausgabeoptionen.

<#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
```

Es stehen zahlreiche Anzeigefilter zur Verfügung. Die gängigsten Wireshark-Anzeigefilter werden

unterstützt.

<#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>
```

Filter können auch als Erfassungsfilter angewendet werden.

```
<#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"
```

Gängige Szenarien

Intermittierender ICMP-Verlust (Ping) an lokale IP

Datenverkehr, der an eine lokale IP-Adresse auf einem Switch weitergeleitet wird, wird in die Warteschlange für Foren (wörtlich "für uns") weitergeleitet. Die Erkennung der Inkrementierung in der Forus CoPP-Warteschlange bezieht sich auf verworfene Pakete, die für den lokalen Switch bestimmt sind. Dies ist relativ einfach und leicht zu konzeptualisieren.

Unter bestimmten Bedingungen kann es jedoch zu Verlusten bei lokal bestimmtem Datenverkehr kommen, die nicht sauber mit Forus-Drops korrelieren.

Bei ausreichendem Datenverkehrsfluss über die CPU ist der Point-Pfad überlastet, sodass CoPP nicht mehr priorisieren kann, für welchen Datenverkehr eine Richtlinie gilt. Der Datenverkehr wird "lautlos" nach dem "first-in" und "first-out"-Prinzip überwacht.

In diesem Szenario wird ein großer Teil des Datenverkehrs durch Richtlinien auf Steuerungsebene geregelt, der interessierende Datenverkehrstyp (in diesem Beispiel "Forus") steigt jedoch nicht unbedingt aktiv an.

Zusammenfassend lässt sich sagen, dass bei einem außergewöhnlich hohen Volumen an CPU-gebundenem Datenverkehr, der sowohl durch aktives CoPP-Policing belegt wird als auch durch eine Paketerfassung oder FED-Fehlerbehebung belegt wird, Verluste auftreten können, die nicht mit der Warteschlange übereinstimmen, für die Sie eine Fehlerbehebung durchführen. Ermitteln Sie in diesem Szenario die Ursache für einen übermäßigen CPU-gebundenen Datenverkehr, und ergreifen Sie Maßnahmen, um die Kontrollebene zu entlasten.

Umleitungen mit hohem ICMP-Wert und langsamer DHCP-Betrieb

Die CoPP auf dem Catalyst Switch der Serie 9000 ist in 32 Hardware-Warteschlangen eingeteilt. Diese 32 Hardware-Warteschlangen werden anhand von 20 Indizes für individuelle Richtlinien angepasst. Jeder Policer-Index korreliert mit einer oder mehreren Hardware-Warteschlangen.

In funktioneller Hinsicht bedeutet dies, dass mehrere Datenverkehrsklassen einen Policer-Index gemeinsam nutzen und einem gemeinsamen aggregierten Policer-Wert unterliegen.

Ein häufiges Problem, das bei aktivierten DHCP-Relay-Agenten auf Switches auftritt, ist die langsame DHCP-Antwort. Clients können IPs sporadisch abrufen, es sind jedoch mehrere Versuche erforderlich, die IPs abzuschließen, und bei einigen Clients dauert es eine Zeitüberschreitung.

Die ICMP-Umleitungswarteschlange und die Broadcast-Warteschlange verwenden gemeinsam einen Richtlinieindex, sodass sich ein hohes Datenverkehrsvolumen, das über dieselbe Switch Virtual Interface (SVI) empfangen und von dieser geroutet wird, auf Anwendungen auswirkt, die auf Broadcast-Datenverkehr angewiesen sind. Dies zeigt sich vor allem, wenn der Switch als Relay-Agent fungiert.

Dieses Dokument bietet eine ausführliche Erläuterung des Konzepts und Möglichkeiten zur Minimierung: Fehlerbehebung bei DHCP-Problemen mit DHCP Relay Agents der Catalyst Serie 9000

Zusätzliche Ressourcen

Fehlerbehebung bei langsamem oder zeitweiligem DHCP auf Catalyst 9000 DHCP Relay Agents

Konfigurieren der FED-CPU-Paketerfassung auf Catalyst 9000-Switches

Catalyst 9300-Switches: Konfigurieren des Control Plane Policing

Configuring Packet Capture - Network Management Configuration Guide, Cisco IOS XE Bengaluru 17.6.x (Catalyst 9300 Switches)

Betrieb und Fehlerbehebung bei DHCP-Snooping auf Catalyst Switches der Serie 9000

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.