Catalyst 9000 스위치의 컨트롤 플레인 작업 문제 해결

목차

소개

이 문서에서는 Cisco IOS® XE를 실행하는 Catalyst 9000 제품군 스위치에서 컨트롤 플레인 상태를 트러블슈팅하고 검증하는 방법에 대해 설명합니다.

배경 정보

스위치의 기본 작업은 가능한 한 빨리 패킷을 전달하는 것입니다. 대부분의 패킷은 하드웨어에서 전달되지만 특정 트래픽 유형은 시스템 CPU에서 처리해야 합니다. CPU에 도착하는 트래픽은 최대한 빨리 처리됩니다. CPU에서 일정한 양의 트래픽이 표시될 것으로 예상되지만 과도한 트래픽이 발생할 경우 운영 문제가 발생합니다. Catalyst 9000 스위치 제품군에는 기본적으로 강력한 CoPP(Control Plane Policing) 메커니즘이 통합되어 있어 CPU의 트래픽 과포화로 인한 문제를 방지합니다.

정상적인 작동의 기능으로서 특정한 사용 사례에서 예기치 못한 문제가 발생한다. 원인과 결과의 상관관계가 명확하지 않은 경우도 있어 문제를 접근하기 어렵다. 이 문서에서는 컨트롤 플레인 상태를 확인할 수 있는 도구를 제공하며, 컨트롤 플레인 펀트 또는 삽입 경로와 관련된 문제에 접근하

는 방법에 대한 워크플로를 제공합니다. 또한 현장에서 발견된 문제를 기반으로 몇 가지 일반적인 시나리오를 제공합니다.

CPU 펀트 경로는 제한된 리소스입니다. 최신 하드웨어 포워딩 스위치는 기하급수적으로 더 많은 양의 트래픽을 처리할 수 있습니다. Catalyst 9000 스위치 제품군은 지정된 시간에 CPU에서 총 약 19,000pps(packets-per-second)를 지원합니다. 이 임계값을 초과하며, 펀트된 트래픽은 가중치 없이 폴리싱됩니다.

용어

- FED(Forwarding Engine Driver): Cisco Catalyst 스위치의 핵심으로서 모든 하드웨어 프로그래밍/포워딩을 담당합니다
- IOSd: Linux 커널에서 실행되는 Cisco IOS 데몬입니다. 커널 내에서 소프트웨어 프로세스의 형태로 실행됩니다
- PDS(Packet Delivery System): 다양한 서브시스템에 패킷이 전달되고 그 내부에서 패킷이 전달되는 아키텍처 및 프로세스입니다. 예를 들면, FED에서 IOSd로 또는 그 반대로 패킷이 전달되는 방법을 제어합니다
- 컨트롤 플레인(CP): 컨트롤 플레인은 Catalyst 스위치의 CPU와 관련된 기능과 트래픽을 그룹화하는 데 사용되는 일반 용어입니다. 여기에는 STP(Spanning Tree Protocol), HSRP(Hot Standby Router Protocol), 스위치를 대상으로 하거나 스위치에서 전송되는 라우팅 프로토콜과 같은 트래픽이 포함됩니다. 여기에는 SSH(Secure Shell), SNMP(Simple Network Management Protocol) 등 CPU에서 처리해야 하는 애플리케이션 레이어 프로토콜도 포함됩니다
- 데이터 플레인(DP): 일반적으로 데이터 플레인은 제어 플레인의 지원 없이 전달되는 하드웨어 ASIC 및 트래픽을 포함합니다
- Punt:처리를 위해 CP로 전송된 DP에서 가로챈 인그레스 프로토콜 제어 패킷
- 삽입:IP 생성 프로토콜 패킷이 IO 인터페이스에서 이그레스(egress) 아웃하기 위해 DP로 전송됨
- LSMPI:Linux 공유 메모리 Punt 인터페이스

Catalyst 9000 CoPP

Catalyst 9000 스위치 제품군의 CPU 보호의 기반은 CoPP입니다. CoPP를 사용하면 시스템에서 생성된 QoS(Quality of Service) 정책이 CPU 주입/주입 경로에 적용됩니다. CPU 바운드 트래픽은 여러 가지 다른 클래스로 그룹화되고, CPU와 연결된 개별 하드웨어 폴리서 간에 매핑됩니다. 폴리서는 특정 트래픽 클래스에 의한 CPU의 과포화(oversaturation)를 방지합니다.

CoPP 구현

CPU 바운드 트래픽은 대기열로 분류됩니다. 이러한 대기열/클래스는 시스템에서 정의되며 사용자가 구성할 수 없습니다. 폴리서는 하드웨어로 구성됩니다. Catalyst 9000 제품군은 32개의 대기열에 대해 32개의 하드웨어 폴리서를 지원합니다.

특정 값은 플랫폼마다 다릅니다. 일반적으로 시스템 정의 대기열은 32개입니다. 이러한 큐는 폴리서 인덱스와 관련된 클래스 맵과 관련이 있습니다. 폴리서 인덱스에는 기본 폴리서 속도가 있습니다. 이 속도는 사용자가 구성할 수 있지만 기본 CoPP 정책을 변경하면 예기치 않은 서비스에 영향

을 미칠 수 있습니다.

CoPP에 대한 시스템 정의 값

클래스 맵 이름	폴리서 인덱스(폴리서 번호)	CPU 큐(큐 번호)
system-cpp-police-data	WK_CPP_POLICE_DATA(0)	WK_CPU_Q_ICMP_GEN(3) WK_CPU_Q_BROADCAST(12) WK_CPU_Q_ICMP_REDIRECT(6)
system-cpp-police-l2-제어	WK_CPP_POLICE_L2_ 컨트롤(1)	WK_CPU_Q_L2_CONTROL(1)
system-cpp-police-routing-control	WK_CPP_POLICE_ROUTING_CONTROL(2)	WK_CPU_Q_ROUTING_CONTROL(4) WK_CPU_Q_LOW_LATENCY(27)
system-cpp-police-control-낮은 우선순위	WK_CPP_POLICE_CO NTROL_LOW_PRI(3)	WK_CPU_Q_GENERAL_PUNT(25)
system-cpp-police-punt-webauth	WK_CPP_POLICE_PU NT_WEBAUTH(7)	WK_CPU_Q_PUNT_WEBAUTH(22)
system-cpp-police- 토폴로지 제어	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTROL(15)
system-cpp-police- 멀티캐스트	WK_CPP_POLICE_MULTICAST(9)	WK_CPU_Q_TRANSIT_TRAFFIC(18) WK_CPU_Q_MCAST_DATA(30)
system-cpp-police-sys-data	WK_CPP_POLICE_SYS _DATA(10)	WK_CPU_Q_LEARNING_CACHE_OVFL(13 WK_CPU_Q_CRYPTO_CONTROL(23)

클래스 맵 이름	폴리서 인덱스(폴리서 번호)	CPU 큐(큐 번호)
		WK_CPU_Q_EXCEPTION(24) WK_CPU_Q_EGR_EXCEPTION(28) WK_CPU_Q_NFL_SAMPLED_DATA(26) WK_CPU_Q_GOLD_PKT(31) WK_CPU_Q_RPF_FAILED(19)
system-cpp-police-dot1x-auth	WK_CPP_POLICE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH(0)
system-cpp-police- 프로토콜 스누핑	WK_CPP_POLICE_PR(12)	WK_CPU_Q_PROTO_SNOOPING(16)
system-cpp-police-sw-forward	WK_CPP_POLICE_SW_FWD (13)	WK_CPU_Q_SW_FORWARDING_Q(14) WK_CPU_Q_LOGGING(21) WK_CPU_Q_L2_LVX_DATA_PACK(11)
system-cpp-police-forus	WK_CPP_POLICE_FORUS(14)	WK_CPU_Q_FORUS_ADDR_RESOLUTIO WK_CPU_Q_FORUS_TRAFFIC(2)
system-cpp-police- 멀티캐스트 엔드 스테이션	WK_CPP_POLICE_MULTICAST_SNOOPING(15)	WK_CPU_Q_MCAST_END_STATION_SERVICE(20)
system-cpp-default	WK_CPP_POLICE_DEFAULT_POLICER(16)	WK_CPU_Q_DHCP_SNOOPING(17) WK_CPU_Q_UNUSED(7) WK_CPU_Q_EWLC_CONTROL(9)

클래스 맵 이름	폴리서 인덱스(폴리서 번호)	CPU 큐(큐 번호)
		WK_CPU_Q_EWLC_DATA(10)
system-cpp-police-stackwise-virt-control	WK_CPP_STACKWISE_VIRTUAL_CONTROL(5)	WK_CPU_Q_STACKWISE_VIRTUAL_CON
system-cpp-police-l2lvx-control	WK_CPP_ L2_LVX_CONT_PACK(4)	WK_CPU_Q_L2_LVX_CONT_PACK(8)

각 대기열은 트래픽 유형 또는 특정 기능 집합과 관련이 있습니다. 이 목록이 완전한 목록은 아닙니다:

CPU 대기열 및 관련 기능

CPU 큐(큐 번호)	기능
WK_CPU_Q_DOT1X_AUTH(0)	IEEE 802.1x 포트 기반 인증
WK_CPU_Q_L2_CONTROL(1)	DTP(Dynamic Trunking Protocol) VTP(VLAN Trunking Protocol) PAgP(Port Aggregation Protocol) CISP(Client Information Signaling Protocol) 메시지 세션 릴레이 프로토콜 MVRP(Multiple VLAN Registration Protocol) MMN(Metropolitan Mobile Network) LLDP(Link Level Discovery Protocol) UDLD(UniDirectional Link Detection) LACP(Link Aggregation Control Protocol)

CPU 큐(큐 번호)	기능
	CDP(Cisco Discovery Protocol) STP(Spanning Tree Protocol)
WK_CPU_Q_FORUS_TRAFFIC(2)	Telnet, Pingv4 및 Pingv6, SNMP와 같은 호스트 Keepalive/루프백 탐지 IPSec(Initiate-Internet Key Exchange) 프로토콜
WK_CPU_Q_ICMP_GEN(3)	ICMP - 대상에 연결할 수 없음 ICMP-TTL이 만료됨
WK_CPU_Q_ROUTING_CONTROL(4)	RIPv1(Routing Information Protocol version 1) RIPv2 IGRP(Interior Gateway Routing Protocol) BGP(Border Gateway Protocol) PIM-UDP VRRP(Virtual Router Redundancy Protocol) HSRPv1(Hot Standby Router Protocol version 1) HSRPv2 GLBP(Gateway Load Balancing Protocol) LDP(Label Distribution Protocol) WCCP(Web Cache Communication Protocol) RIPng(Routing Information Protocol) 차세대 OSPF(Open Shortest Path First) Open Shortest Path First 버전 3(OSPFv3) EIGRP(Enhanced Interior Gateway Routing Protocol)

CPU 큐(큐 번호)	기능
	EIGRPv6(Enhanced Interior Gateway Routing Protocol version 6) DHCPv6 PIM(Protocol Independent Multicast) PIMv6(Protocol Independent Multicast version 6) HSRPng(Hot Standby Router Protocol next generation) IPv6 제어 GRE(Generic Routing Encapsulation) 킵얼라이브 NAT(Network Address Translation) 펀트 IS-IS(Intermediate System-to-Intermediate System)
WK_CPU_Q_FORUS_ADDR_RESOLUTION(5)	ARP(Address Resolution Protocol) IPv6 네이버 광고 및 네이버 요청
WK_CPU_Q_ICMP_REDIRECT(6)	ICMP(Internet Control Message Protocol) 재전송
WK_CPU_Q_INTER_FED_TRAFFIC(7)	내부 통신을 위한 레이어 2 브리지 도메인 삽입
WK_CPU_Q_L2_LVX_CONT_PACK(8)	Exchange ID(XID) 패킷
WK_CPU_Q_EWLC_CONTROL(9)	eWLC(Embedded Wireless Controller) [CAPWAP(Control and Provisioning of Wireless Access Points) (UDP 5246)]
WK_CPU_Q_EWLC_DATA(10)	eWLC 데이터 패킷(CAPWAP DATA, UDP 5247)

CPU 큐(큐 번호)	기능
WK_CPU_Q_L2_LVX_DATA_PACK(11)	맵 요청에 대해 펀트된 알 수 없는 유니캐스트 패킷입니다.
WK_CPU_Q_BROADCAST(12)	모든 유형의 브로드캐스트
WK_CPU_Q_OPENFLOW(13)	학습 캐시 오버플로(레이어 2 + 레이어 3)
WK_CPU_Q_CONTROLLER_PUNT(14)	데이터 - ACL(Access Control List) 전체 데이터 - IPv4 옵션 데이터 - IPv6 홉별 데이터 - 리소스 부족/모두 포착 데이터 - RPF(Reverse Path Forwarding) 불완전 군더더기 패킷
WK_CPU_Q_TOPOLOGY_CONTROL(15)	STP(Spanning Tree Protocol) REP(Resilient Ethernet Protocol) SSTP(Shared Spanning Tree Protocol)
WK_CPU_Q_PROTO_SNOOPING(16)	DAI(Dynamic ARP Inspection)를 위한 ARP(Address Resolution Protocol) 스누핑
WK_CPU_Q_DHCP_SNOOPING(17)	DHCP 스누핑
WK_CPU_Q_TRANSIT_TRAFFIC(18)	이는 소프트웨어 경로에서 처리해야 하는 NAT에서 펀칭하는 패킷에 사용됩니다.
WK_CPU_Q_RPF_FAILED(19)	데이터 - mRPF(멀티캐스트 RPF) 실패
WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	IGMP(Internet Group Management Protocol)/MLD(Multicast Listener Discovery) 제어

CPU 큐(큐 번호)	기능
WK_CPU_Q_LOGGING(21)	ACL(Access Control List) 로깅
WK_CPU_Q_PUNT_WEBAUTH(22)	웹 인증
WK_CPU_Q_HIGH_RATE_APP(23)	브로드캐스트
WK_CPU_Q_EXCEPTION(24)	IKE 표시 IP 학습 위반 IP 포트 보안 위반 IP 고정 주소 위반 IPv6 범위 확인 RCP(Remote Copy Protocol) 예외 유니캐스트 RPF 실패
WK_CPU_Q_SYSTEM_CRITICAL(25)	미디어 신호 처리/무선 프록시 ARP
WK_CPU_Q_NFL_SAMPLED_DATA(26)	Netflow 샘플 데이터 및 MSP(미디어 서비스 프록시)
WK_CPU_Q_LOW_LATENCY(27)	BFD(Bidirectional Forwarding Detection), PTP(Precision Time Protocol)
WK_CPU_Q_EGR_EXCEPTION(28)	이그레스 해결 예외
WK_CPU_Q_STACKWISE_VIRTUAL_CONTROL(29)	전면 스태킹 프로토콜, 즉 SVL
WK_CPU_Q_MCAST_DATA(30)	데이터 - (S,G) 생성 데이터 - 로컬 조인 데이터 - PIM 등록 데이터 - SPT 전환

CPU 큐(큐 번호)	기능
	데이터 - 멀티캐스트
WK_CPU_Q_GOLD_PKT(31)	골드

기본 정책

기본적으로 시스템 생성 CoPP 정책은 펀트/삽입 경로에 적용됩니다. 기본 정책은 일반적인 MQC 기반 명령을 사용하여 볼 수 있습니다. 스위치 컨피그레이션에서도 볼 수 있습니다. CPU/컨트롤 플레인의 인그레스 또는 이그레스에 적용할 수 있는 유일한 정책은 시스템 정의 정책입니다.

"show policy-map control-plane"을 사용하여 제어 평면에 적용된 정책을 봅니다.

```
<#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 조정

CoPP 폴리서 속도는 사용자가 구성할 수 있습니다. 사용자는 대기열을 비활성화할 수도 있습니다.

이 예에서는 개별 폴리서 값을 조정하는 방법을 보여 줍니다. 이 예에서 조정된 클래스는 "system-cpp-police-protocol-snooping"입니다.

<#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
```

이 예에서는 큐를 완전히 비활성화하는 방법을 보여 줍니다. 큐가 비활성화되면 CPU의 과포화 상태가 될 수 있으므로 주의하십시오.

<#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
```

문제 해결

방법론

CPU 사용률은 프로세스와 중단이라는 두 가지 기본 활동의 영향을 받습니다. 프로세스는 CPU가 수행하는 구조화된 활동이며, 중단은 데이터 플레인에서 가로채기되어 작업을 위해 CPU로 전송되는 패킷을 가리킵니다. 이러한 활동은 모두 CPU의 총 사용률을 구성합니다. CoPP는 기본적으로 활성화되어 있으므로 서비스 영향은 높은 CPU 사용률과 상관관계가 없습니다. CoPP가 작업을 수행할 경우 CPU 사용률은 크게 영향을 받지 않습니다. CPU의 전체적인 활용도를 고려하는 것도 중요하지만, 전체적인 활용도가 전체 스토리를 말해주지는 않는다. 이 섹션의 show 명령 및 유틸리티는 CPU의 상태를 신속하게 평가하고 CPU 바인딩 트래픽에 대한 관련 세부 정보를 식별하는 데 사용됩니다.

지침:

- 문제가 컨트롤 플레인과 관련이 있는지 확인합니다. 대부분의 전송 트래픽은 하드웨어에서 전달됩니다. 특정 트래픽 유형 및 특정 시나리오에만 CPU와 컨트롤 플레인이 포함되므로 조사 내내 이 점을 염두에 두십시오.
- 사용 기준을 파악합니다. 정상 활용이 어떻게 보이는지 이해하는 것이 중요하기 때문에 규범으로부터의 편차를 확인할 수 있다.
- 프로세스 및 중단 모두에 대한 전체 활용도를 검증합니다. 예기치 않은 양의 CPU 사이클을 사용하는 프로세스를 식별합니다. 사용률이 예상 범위를 벗어나는 경우 이는 잠재적으로 문제가 될 수 있습니다. 어떤 시스템에 대한 평균 활용도를 파악하는 것이 중요한데, 이를 통해 규범 밖의 편차가 인식되게 된다. 사용률만으로는 컨트롤 플레인 상태를 완벽하게 파악할 수 없습니다.
- CoPP에서 드롭이 활발하게 증가하고 있는지 확인합니다. CoPP 드롭이 항상 문제를 나타내는 것은 아니지만, 능동적으로 폴리싱된 트래픽 클래스와 관련된 문제를 트러블슈팅할 경우 이는 관련성을 나타내는 강력한 지표입니다.

유용한 Show 명령

이 스위치를 사용하면 CPU 상태 및 CoPP 통계를 빠르게 관리할 수 있습니다. 또한 CPU 바운드 트래픽의 인그레스 지점을 신속하게 확인하는 데 유용한 CLI가 있습니다.

전체 및 기록 사용률 결정

- 전체 CPU 사용률을 보려면 "프로세스 cpu를 정렬하여 표시"를 사용합니다. "sorted" 인수는 사용률을 기준으로 프로세스 출력을 정렬합니다. 더 많은 CPU 리소스를 사용하는 프로세스가 출력의 맨 위에 있습니다. 인터럽트로 인한 사용률도 백분율로 제공됩니다.

<#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
```

- "프로세스 CPU 기록 표시"는 지난 60초, 5분 및 72시간 동안의 CPU 사용률에 대한 기록 그래프를 제공합니다.

<#root>

```
Catalyst-9600#
show processes cpu history

    99977777666668888866666777777777888887777766666999998888866
<<<--- The numbers at the top of each column represent the highest value seen throughout the time perio

    22255555999994444444440000088888888811111777773333355555500
It is read top-down. "9" over "2" in this example means "92%" for example.

100
 90  *** ***** **********
 80  ******** ***** ********** **********
 70  ****************** ************************************
 60  ************************************************************
 50  ************************************************************
 40  ************************************************************
 30  ************************************************************
 20  ************************************************************
 10  ************************************************************
```

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

```
0....5....1....1....2....2....3....3....4....4....5....5....6
```

In this example, utilization spiked to 92% in the last 5 seconds.

```
          0    5    0    5    0    5    0    5    0    5    0
          CPU% per second (last 60 seconds)
         * = maximum CPU% # = average CPU%
```

```
    9998989899998989989989989889999989889898899999999899999999
    4318230911026353162351292837713365748928096040142309011335 11
100 ** *
90  ***** **************************************************
80  ************#***#**#***####*#######**#***#***#**********
70  ############################################################
```

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

```
60  ############################################################
```

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

```
50  ############################################################
```

the maximum utilization spiked to 94%.

```
40  ############################################################
30  ############################################################
20  ############################################################
10  ############################################################
0....5....1....1....2....2....3....3....4....4....5....5....6
          0    5    0    5    0    5    0    5    0    5    0
          CPU% per minute (last 60 minutes)
         * = maximum CPU% # = average CPU%
```

```
    9999999999999999999999999999999999999999999999999999999999999999999999999999
    6656565666465556666556565756545565677375555675745455457759575546485767 57
100 ********** ****************** ******* ********* * ** ********* * *****
90  ************************************************************************
80  ************************************************************************
70  ########################################################################
60  ########################################################################
50  ########################################################################
40  ########################################################################
30  ########################################################################
20  ########################################################################
10  ########################################################################
0....5....1....1....2....2....3....3....4....4....5....5....6....6....7..
          0    5    0    5    0    5    0    5    0    5    0    5    0
               CPU% per hour (last 72 hours)
              * = maximum CPU% # = average CPU%
```

컨트롤 플레인 정책 확인

- "show platform hardware fed <switch> active qos queue stats internal cpu policer"를 사용하여 큐/폴리서 구조에 대한 집계 CoPP 통계 및 추가 정보를 봅니다. 이 출력은 제어 평면의 마지막 재설정 이후 폴리서 통계의 기록 보기를 제공합니다. 이러한 카운터는 수동으로 정리할 수도 있습니다. 일반적으로 폴리서가 제어 평면을 삭제하는 경우 관련된 대기열/클래스의 문제가 나타나지만 문제가 발생하는 동안 삭제는 활발하게 증가해야 합니다. Queue Drop 값이 증가하는지 관찰하려면 명령을 여러 번 실행합니다.

<#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    Policer Accept   Policer Accept  Policer Drop  Policer Drop
  Index         Bytes            Frames        Bytes          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/

펀트된 트래픽에 대한 정보 수집

이 명령은 트래픽 유형 및 인그레스 물리적 지점을 포함하여 CPU에 적용되는 트래픽에 대한 정보를 수집하는 데 사용됩니다.

- "Show platform software fed <switch> active punt cpuq all" 또는 "Show platform software fed <switch> active punt cpuq <0-31 Queue ID>"를 사용하여 전체 또는 특정 CPU 대기열과 관련된 통계를 볼 수 있습니다.

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

- CPU에서 볼 수 있는 모든 다른 트래픽 유형을 간략하게 살펴보려면 "show platform software

fed <switch> active punt cause summary"를 사용하십시오. 0이 아닌 원인만 표시됩니다.

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

- CPU 바운드 트래픽이 시스템을 인그레스하는 인터페이스를 빠르게 보려면 "show platform software fed <switch> active punt rate interfaces" 명령을 사용합니다. 이 명령은 0이 아닌 입력 대기열이 있는 인터페이스만 표시합니다.

<#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

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

- 인터페이스의 개별 대기열을 드릴다운하고 보려면 "show platform software fed <switch> active punt rate interfaces <IF-ID>"를 사용합니다. 이 명령은 집계 통계를 보여주며, 트래픽이 폴리싱된 경우 기록 입력 대기열 활동을 보는 데 사용할 수 있습니다.

<#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

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

CPU 바운드 트래픽 검사

Catalyst 9000 스위치 제품군은 CPU 바운드 트래픽을 모니터링하고 볼 수 있는 유틸리티를 제공합니다. 이러한 툴을 사용하여 어떤 트래픽이 CPU에 능동적으로 적용되는지 파악합니다.

EPC(Embedded Packet Capture)

제어 평면의 EPC는 어느 방향(또는 둘 다)으로도 수행할 수 있습니다. 펀트된 트래픽의 경우 인바운드를 캡처합니다. 컨트롤 플레인의 EPC를 버퍼 또는 파일에 저장할 수 있습니다.

<#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
```

캡처 결과는 간략한 출력 또는 자세한 출력으로 볼 수 있습니다.

<#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: 0000000000000000000000000000
    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
```

캡처 결과는 파일에 직접 기록하거나 버퍼에서 내보낼 수 있습니다.

<#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 패킷 캡처

Catalyst 9000 스위치 제품군은 CPU를 오가는 패킷의 향상된 가시성을 허용하는 디버그 유틸리티를 지원합니다.

```
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)
```

버퍼 내용에는 출력에 대한 간략하고 자세한 옵션이 있습니다.

<#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
```

많은 디스플레이 필터를 사용할 수 있습니다. 대부분의 일반적인 Wireshark 디스플레이 필터가 지원됩니다.

<#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>
```

필터를 캡처 필터로 적용할 수도 있습니다.

<#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"
```

일반적인 시나리오

로컬 IP에 대한 간헐적 ICMP(Ping) 손실

스위치의 로컬 IP로 전달되는 트래픽은 Forus(문자 그대로 "for us") 대기열에 기록됩니다. Forus CoPP 대기열이 증가하는 것은 로컬 스위치로 향하는 삭제된 패킷과 관련이 있습니다. 이는 비교적 직진적이고 개념화가 용이하다.

그러나 어떤 상황에서는 Forus 삭제와 깔끔하게 상관관계가 없는 로컬로 목적지가 지정된 트래픽에 손실이 발생할 수 있습니다.

충분한 CPU 바운드 트래픽 플로우가 이루어지면, CoPP가 어떤 트래픽을 폴리싱할지 우선 순위를 지정할 수 없을 정도로 펀트 경로가 과포화 상태가 됩니다. 트래픽은 선입 선출 방식으로 '조용히' 폴리싱됩니다.

이 시나리오에서는 대량의 컨트롤 플레인 정책에 대한 증거가 확인되지만, 관심 트래픽 유형(이 예에서는 Forus)이 반드시 증가하지는 않습니다.

요약하면, 활성 CoPP 폴리싱에서 입증되고 패킷 캡처 또는 FED punt 디버그로 입증된 매우 많은 양의 CPU 바운드 트래픽이 있을 경우, 문제 해결 중인 대기열에 맞지 않는 손실이 발생할 수 있습니다. 이 시나리오에서는 과도한 양의 CPU 바인딩 트래픽이 있는 이유를 확인하고 컨트롤 플레인의 부담을 덜어주기 위한 조치를 취합니다.

높은 ICMP 리디렉션 및 DHCP 작업 부진

Catalyst 9000 Series 스위치의 CoPP는 32개의 하드웨어 대기열로 구성됩니다. 이러한 32개의 하드웨어 대기열은 20개의 개별 폴리서 인덱스에 맞춰집니다. 각 폴리서 인덱스는 하나 이상의 하드웨어 대기열과 상호 연관됩니다.

기능적으로는 여러 트래픽 클래스가 하나의 폴리서 인덱스를 공유하며 공통의 종합 폴리서 값에 종속됨을 의미합니다.

DHCP 릴레이 에이전트가 활성화된 스위치에서 나타나는 일반적인 문제는 느린 DHCP 응답과 관련이 있습니다. 클라이언트는 산발적으로 IP를 얻을 수 있지만 완료하는 데 몇 가지 시도가 필요하고 일부 클라이언트는 시간이 초과됩니다.

 ICMP 리디렉션 대기열과 브로드캐스트 대기열은 폴리서 인덱스를 공유하므로, 동일한 SVI(Switch Virtual Interface)에서 수신 및 라우팅되는 많은 양의 트래픽은 브로드캐스트 트래픽에 의존하는 애플리케이션에 영향을 줍니다. 이는 스위치가 릴레이 에이전트 역할을 하는 경우 특히 두드러집니다.

이 문서에서는 Catalyst 9000 DHCP 릴레이 에이전트의 개념 및 완화 방법에 대한 자세한 설명을 제공합니다.

추가 리소스

Catalyst 9000 DHCP 릴레이 에이전트의 느린 DHCP 또는 간헐적인 DHCP 문제 해결

Catalyst 9000 스위치에서 FED CPU 패킷 캡처 구성

Catalyst 9300 스위치: 컨트롤 플레인 정책 구성

패킷 캡처 구성 - 네트워크 관리 컨피그레이션 가이드, Cisco IOS XE Bengaluru 17.6.x(Catalyst 9300 스위치)

Catalyst 9000 스위치에서 DHCP 스누핑 운영 및 문제 해결

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.