

# Catalyst 4500 스위치의 높은 CPU 사용률 문제 해결

## 목차

---

### [소개](#)

#### [사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기 규칙](#)

### [배경 정보](#)

#### [Catalyst 4500 CPU 패킷 처리 아키텍처 이해](#)

#### [Catalyst 4500의 CPU 사용률이 높은 이유 식별](#)

[CPU 사용량 베이스라인](#)

[Catalyst 4500 스위치의 show processes cpu 명령어 이해](#)

[Catalyst 4500 스위치의 show platform health 명령어 이해](#)

#### [일반적인 높은 CPU 사용률 문제 해결](#)

##### [프로세스 전환 패킷으로 인한 높은 CPU 사용률](#)

[다수의 스페닝 트리 포트 인스턴스](#)

2단계: [show platform health](#) 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

3단계: CPU 바인딩 트래픽의 유형을 식별하기 위해 트래픽을 수신하는 CPU 큐를 확인합니다.

4단계: 근본 원인 파악

##### [ICMP 리디렉션. 동일한 인터페이스에서 패킷 라우팅](#)

1단계: [show processes cpucommand](#)를 사용하여 Cisco IOS 프로세스를 확인합니다.

2단계: [show platform health](#) 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

3단계: CPU 바인딩 트래픽의 유형을 식별하기 위해 트래픽을 수신하는 CPU 큐를 확인합니다.

4단계: 근본 원인 파악

##### [IPX 또는 AppleTalk 라우팅](#)

1단계: [show processes cpu](#) 명령을 사용하여 Cisco IOS 프로세스를 확인합니다.

2단계: [show platform health](#) 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

3단계: CPU 바인딩 트래픽의 유형을 식별하려면 트래픽을 수신하는 CPU 큐를 확인합니다.

4단계: 근본 원인 파악

##### [호스트 학습](#)

1단계: [show processes cpu](#) 명령을 사용하여 Cisco IOS 프로세스를 확인합니다.

2단계: [show platform health](#) 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

3단계: CPU 바인딩 트래픽의 유형을 식별하려면 트래픽을 수신하는 CPU 큐를 확인합니다.

4단계: 근본 원인 파악

##### [보안 ACL을 위한 하드웨어 리소스 부족\(TCAM\)](#)

1단계: [show processes cpu](#) 명령을 사용하여 Cisco IOS 프로세스를 확인합니다.

2단계: [show platform health](#) 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

3단계: CPU 바인딩 트래픽의 유형을 식별하려면 트래픽을 수신하는 CPU 큐를 확인합니다.

4단계: 문제 해결

##### [ACL의 로그 키워드](#)

1단계: [show processes cpu](#) 명령을 사용하여 Cisco IOS 프로세스를 확인합니다.

2단계: [show platform health](#) 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

3단계: CPU 바인딩 트래픽의 유형을 식별하려면 트래픽을 수신하는 CPU 큐를 확인합니다.

[4단계: 문제 해결](#)

[레이어 2 전달 루프](#)

[1단계: thehow processes cpu 명령을 사용하여 Cisco IOS 프로세스를 확인합니다.](#)

[2단계: show platform health 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.](#)

[3단계: CPU 바인딩 트래픽의 유형을 식별하기 위해 트래픽을 수신하는 CPU 큐를 확인합니다.](#)

[4단계: 근본 원인 파악 및 문제 해결](#)

[5단계: 고급 STP 기능 구현](#)

[CPU 사용률이 높은 기타 원인](#)

[과도한 링크 플랩](#)

[1단계: show processes cpu 명령을 사용하여 Cisco IOS 프로세스를 확인합니다.](#)

[2단계: show platform health 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.](#)

[3단계: 근본 원인 파악](#)

[FIB 일관성 검사로 인한 CPU 사용률 급증](#)

[K2FibAdjMan Host Move 프로세스에서 높은 CPU 사용률](#)

[RkiosPortMan Port Review 프로세스에서 높은 CPU 사용률](#)

[트렁크 포트를 사용하여 IP 전화기에 연결된 경우 높은 CPU 사용률](#)

[RSPAN 및 레이어 3 제어 패킷을 통한 높은 CPU 사용률](#)

[CPU로 향하는 트래픽을 분석하기 위한 문제 해결 도구](#)

[툴 1: SPAN으로 CPU 트래픽 모니터링—Cisco IOS Software 릴리스 12.1\(19\)EW 이상](#)

[툴 2: 내장 CPU 스니퍼—Cisco IOS Software 릴리스 12.2\(20\)EW 이상](#)

[툴 3: CPU에 트래픽을 전송하는 인터페이스 식별—Cisco IOS Software 릴리스 12.2\(20\)EW 이상](#)

[요약](#)

[관련 정보](#)

---

## 소개

이 문서에서는 CPU 패킷 처리 아키텍처에 대해 설명하고 Catalyst 4500 스위치에서 CPU 사용량이 많은 원인을 식별하는 방법을 보여 줍니다.

## 사전 요구 사항

### 요구 사항

이 문서에 대한 특정 요건이 없습니다.

### 사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Catalyst 4500 시리즈 스위치
- Catalyst 4948 시리즈 스위치

---



참고: 이 문서는 Cisco IOS® 소프트웨어 기반 스위치에만 적용됩니다.

---

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 표기 규칙

문서 규칙에 대한 자세한 내용은 [Cisco 기술 팀 표기 규칙](#)을 참조하십시오.

## 배경 정보

Catalyst 4948 스위치를 포함하는 Catalyst 4500 시리즈 스위치에는 CPU 바인딩 트래픽을 위한 정교한 패킷 처리 방법이 있습니다. 일반적으로 인식되는 문제는 이러한 스위치에서 높은 CPU 사용률입니다. 이 문서는 CPU 패킷 처리 아키텍처에 대한 세부 사항을 제공하고 이러한 스위치에서 높은 CPU 사용률의 원인을 식별하는 방법을 보여줍니다. 이 문서에는 Catalyst 4500 시리즈에서

CPU 사용률을 높이는 몇 가지 일반적인 네트워크 또는 컨피그레이션 시나리오도 나와 있습니다

CPU 패킷 처리 아키텍처를 살펴보고 높은 CPU 사용률의 문제를 해결하기 전에 하드웨어 기반 전달 스위치와 Cisco IOS 소프트웨어 기반 라우터가 CPU를 사용하는 다양한 방법을 이해해야 합니다. 높은 CPU 사용률이 디바이스의 리소스 고갈 및 충돌의 위협을 나타낸다는 것은 일반적인 오해입니다. 용량 문제는 Cisco IOS 라우터에서 높은 CPU 사용률의 증상 중 하나입니다. 그러나 용량 문제는 Catalyst 4500과 같은 하드웨어 기반 전달 스위치에서 CPU 사용률이 높아지는 현상이 아닙니다. Catalyst 4500은 하드웨어 ASIC(Application-Specific integrated Circuit)의 패킷을 전달하고 초당 최대 1억 2백만 Mpps의 트래픽 전달 속도에 도달하도록 설계되었습니다.

Catalyst 4500 CPU는 다음 기능을 수행합니다.

- 구성된 소프트웨어 프로토콜을 관리합니다. 예를 들면 다음과 같습니다.
  - STP(Spanning Tree Protocol)
  - 라우팅 프로토콜
  - CDP(Cisco Discovery Protocol)
  - PAgP(Port Aggregation Protocol)
  - VTP(VLAN Trunk Protocol)
  - DTP(Dynamic Trunking Protocol)
- 하드웨어 ASIC에 대한 컨피그레이션/동적 항목을 프로그램합니다. 예를 들면 다음과 같습니다.
  - Access control lists (ACLs)
  - CEF 항목
- 다양한 구성 요소를 내부적으로 관리합니다. 예를 들면 다음과 같습니다.
  - PoE(Power over Ethernet) 라인 카드
  - 전원 공급 장치
  - Fan tray
- 스위치에 대한 액세스를 관리합니다. 예를 들면 다음과 같습니다.
  - Telnet
  - Console
  - SNMP(Simple Network Management Protocol)
- 소프트웨어 경로를 통해 패킷을 전달합니다. 예를 들면 다음과 같습니다.
  - 소프트웨어 경로에서만 지원되는 IPX(Internet Packet Exchange) 라우팅 패킷

- MTU(Maximum Transmission Unit) 단편화

이 목록에 따르면 높은 CPU 사용률은 CPU에서 패킷을 수신하거나 처리하는 과정에서 발생할 수 있습니다. 프로세스를 위해 전송되는 일부 패킷은 네트워크 작업에 필수적일 수 있습니다. 이러한 필수 패킷의 예로는 스페닝 트리 토폴로지 컨피그레이션을 위한 BPDU(Bridge Protocol Data Unit)가 있습니다. 그러나 다른 패킷은 소프트웨어 전달 데이터 트래픽일 수 있습니다. 이러한 시나리오에서는 스위칭 ASIC가 처리를 위해 CPU로 패킷을 전송해야 합니다.

- CPU에 복사된 패킷이지만 원래 패킷은 하드웨어에서 전환됨.

호스트 MAC 주소 학습을 예로 들 수 있습니다.

- 처리를 위해 CPU로 전송되는 패킷

예를 들면 다음과 같습니다.

- 라우팅 프로토콜 업데이트
- Bpdu
- 의도적이거나 의도하지 않은 트래픽 플러드

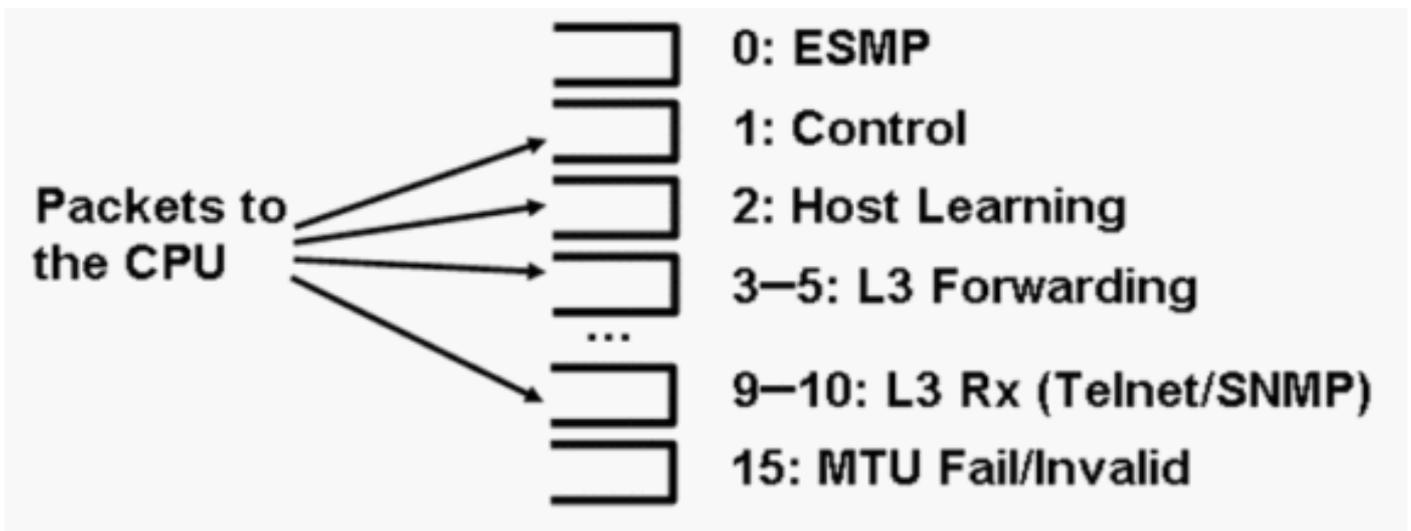
- 전달하기 위해 CPU로 전송되는 패킷

IPX 또는 AppleTalk 라우팅이 필요한 패킷을 예로 들 수 있습니다.

## Catalyst 4500 CPU 패킷 처리 아키텍처 이해

Catalyst 4500에는 CPU로 향하는 트래픽 유형 간 차별화를 위해 QoS(Quality of Service) 메커니즘이 내장되어 있습니다. 이 메커니즘은 L2(Layer 2)/L3(Layer 3)/L4(Layer 4) 패킷 정보를 기반으로 차별화합니다. 슈퍼바이저 패킷 엔진에는 다양한 유형의 패킷 또는 이벤트를 처리하기 위한 16개의 대기열이 있습니다. [그림 1](#)은 이러한 대기열을 보여줍니다. [표 1](#)에는 대기열 및 각 대기열에 있는 패킷 유형이 나와 있습니다. 16개 대기열은 Catalyst 4500이 패킷 유형 또는 우선순위에 따라 패킷을 대기열에 추가하도록 허용합니다.

그림 1 - Catalyst 4500에서 다중 CPU 대기열 사용



Catalyst 4500에서 여러 CPU 큐 사용

표 1 – Catalyst 4500 대기열 설명

대기열 번호	큐 이름	대기 중인 패킷
0	ESMP	라인 카드 ASIC 또는 기타 구성 요소 관리용 ESMP <sup>1</sup> 패킷(내부 관리 패킷)
1	제어	L2 컨트롤 플레인 패킷(예: STP, CDP, PAgP, LACP <sup>2</sup> 또는 UDLD <sup>3</sup> )
2	호스트 학습	L2 포워딩 테이블을 구축하기 위해 CPU에 복사된 알 수 없는 소스 MAC 주소가 있는 프레임
3, 4, 5	L3 Fwd 최고, L3 Fwd 높음/중간, L3 Fwd 낮음	소프트웨어에서 전달해야 하는 패킷(예: GRE <sup>4</sup> 터널) ARP <sup>5</sup> 가 대상 IP 주소에 대해 확인되지 않으면 패킷이 이 큐로 전송됩니다.
6, 7, 8	L2 Fwd 최상위, L2 Fwd 높음/중간, L2 Fwd 낮음	브리징의 결과로 전달되는 패킷 <ul style="list-style-type: none"> <li>• IPX 및 AppleTalk 라우팅 패킷과 같이 하드웨어에서 지원되지 않는 프로토콜은 CPU에 브리지됩니다.</li> <li>• ARP 요청 및 응답</li> <li>• 스위치 SVI<sup>6</sup>/L3 인터페이스의 대상 MAC 주소가 있는 패킷은 다음과 같은 이유로 하드웨어에서 패킷을 라우팅할 수 없는 경우 브리지됩니다. <ul style="list-style-type: none"> <li>◦ IP 헤더 옵션</li> <li>◦ 만료된 TTL<sup>7</sup></li> <li>◦ 비 ARPA 캡슐화</li> </ul> </li> </ul>
9, 10	L3 Rx 높음, L3 Rx 낮음	L3 컨트롤 플레인 트래픽(예: CPU IP 주소로 향하는 라우팅 프로토콜) 예시에는 텔넷, SNMP 및 SSH <sup>8</sup> 이 포함됩니다.
11	RPF 실패	RPF <sup>9</sup> 검사에 실패한 멀티캐스트 패킷입니다.

12	ACL Fwd(스누핑)	DHCP <sup>10</sup> 스누핑, 동적 ARP 검사 또는 IGMP <sup>11</sup> 스누핑 기능에서 처리되는 패킷
13	ACL 로그, 연결 불가	출력 ACL의 거부 또는 대상에 대한 경로 부족으로 인해 삭제된 패킷 또는 logkeyword로 ACE <sup>12</sup> 에 도달한 패킷 이러한 패킷에는 ICMP 도달 불가 메시지가 생성되어야 합니다.
14	ACL SW 처리	보안 ACL을 위한 TCAM <sup>13</sup> 과 같은 추가 ACL 하드웨어 리소스가 부족하여 CPU에 펀트된 패킷
15	MTU 실패/잘못됨	출력 인터페이스 MTU 크기가 패킷 크기보다 작아 프래그먼트화해야 하는 패킷

<sup>1</sup>ESMTP = 간단한 관리 프로토콜도 있습니다.

<sup>2</sup>LACP = Link Aggregation Control Protocol.

<sup>3</sup>UDLD = 단방향 링크 감지.

<sup>4</sup>GRE = 일반 라우팅 캡슐화

<sup>5</sup>ARP = Address Resolution Protocol.

<sup>6</sup>SVI = 스위치드 가상 인터페이스.

<sup>7</sup>TTL = Time to Live.

<sup>8</sup>SSH = Secure Shell Protocol.

<sup>9</sup>RPF = 역방향 경로 전달

<sup>10</sup>DHCP = Dynamic Host Configuration Protocol입니다.

<sup>11</sup>IGMP = 인터넷 그룹 관리 프로토콜.

<sup>12</sup>ACE = 액세스 제어 항목

<sup>13</sup>TCAM = TCAM(ternary content addressable memory)

이러한 대기열은 별도의 대기열입니다.

- L2 Fwd HighestL3 Fwd Highest
- L2 Fwd High/MediumorL3 Fwd High/Medium
- L2 Fwd 하한L3 Fwd 하한
- L3 Rx 높음L3 Rx 낮음

패킷은 IP ToS(Type of Service)의 차별화된 DSCP(Differentiated Services Code Point) 값인 QoS 레이블을 기준으로 이러한 대기열에 대기됩니다. 예를 들어, DSCP가 63인 패킷은 L3 Fwd Highestqueue에 대기됩니다. 이 16개 대기열에 대해 수신 및 삭제된 패킷을 thehow platform cpu

packet statistics allcommand의 출력에서 볼 수 있습니다. 이 명령어의 출력은 매우 깁니다. 0이 아닌 이벤트만 표시하려면 theshow platform cpu packet statistics 명령을 실행합니다. 대체 명령어는 show platform cpuport 명령어입니다. Cisco IOS Software Release 12.1(11)EW 또는 이전 버전을 실행하는 경우에만 show platform cpuport 명령을 사용합니다. 이 명령어는 이후 더 이상 사용하지 않습니다. 그러나 이 이전 명령은 Cisco IOS Software Release 12.2(20)EWA 이전 버전의 Cisco IOS Software Release에서 how tech-supportcommand의 일부였습니다.

모든 문제를 해결하려면 theshow platform cpu packet statistics 명령을 사용합니다.

<#root>

Switch#

show platform cpu packet statistics all

!--- Output suppressed.

Total packet queues 16

Packets Received by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Esm	0	0	0	0	0
Control	48	0	0	0	0
Host Learning	0	0	0	0	0
L3 Fwd High	0	0	0	0	0
L3 Fwd Medium	0	0	0	0	0
L3 Fwd Low	0	0	0	0	0
L2 Fwd High	0	0	0	0	0
L2 Fwd Medium	0	0	0	0	0
L2 Fwd Low	0	0	0	0	0
L3 Rx High	0	0	0	0	0
L3 Rx Low	0	0	0	0	0
RPF Failure	0	0	0	0	0
ACL fwd(snooping)	0	0	0	0	0
ACL log, unreachable	0	0	0	0	0
ACL sw processing	0	0	0	0	0
MTU Fail/Invalid	0	0	0	0	0

Packets Dropped by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Esm	0	0	0	0	0
Control	0	0	0	0	0
Host Learning	0	0	0	0	0
L3 Fwd High	0	0	0	0	0
L3 Fwd Medium	0	0	0	0	0
L3 Fwd Low	0	0	0	0	0
L2 Fwd High	0	0	0	0	0
L2 Fwd Medium	0	0	0	0	0
L2 Fwd Low	0	0	0	0	0
L3 Rx High	0	0	0	0	0
L3 Rx Low	0	0	0	0	0
RPF Failure	0	0	0	0	0
ACL fwd(snooping)	0	0	0	0	0
ACL log, unreachable	0	0	0	0	0

ACL sw processing	0	0	0	0	0
MTU Fail/Invalid	0	0	0	0	0

Catalyst 4500 CPU는 표 1에 나열된 다양한 대기열에 **가중치**를 할당합니다. CPU는 중요도 또는 유형, 트래픽 우선순위 또는 DSCP를 기준으로 가중치를 할당합니다. CPU는 대기열의 상대적 가중치를 기준으로 대기열에 서비스를 제공합니다. 예를 들어 BPDU와 같은 제어 패킷과 ICMP 에코 요청이 모두 보류 중인 경우 CPU가 먼저 제어 패킷을 서비스합니다. 우선순위가 낮거나 중요도가 낮은 트래픽의 과도한 양이 시스템을 처리하거나 관리할 수 있는 CPU의 기능을 고갈시키지 않습니다. 이 메커니즘을 사용하면 CPU 사용률이 높더라도 네트워크가 안정적으로 유지됩니다. 네트워크의 안정적인 상태 유지 기능은 반드시 이해해야 하는 중요한 정보입니다.

Catalyst 4500 CPU 패킷 처리에 대한 또 다른 매우 중요한 구현 세부 사항이 있습니다. CPU가 높은 우선순위 패킷 또는 프로세스를 이미 서비스했지만 특정 기간 동안 더 많은 예비 CPU 주기가 있는 경우, CPU는 낮은 우선순위 대기열 패킷을 서비스하거나 낮은 우선순위의 백그라운드 프로세스를 수행합니다. 우선순위가 낮은 패킷 처리 또는 백그라운드 프로세스의 결과로 CPU 사용률이 높으면 정상으로 간주됩니다. 왜냐하면 CPU는 항상 이용 가능한 모든 시간을 사용하려고 하기 때문입니다. 이러한 방식으로 CPU는 스위치의 안정성을 손상시키지 않으면서 스위치 및 네트워크의 최대 성능을 위해 노력합니다. 단일 시간 슬롯에 대해 CPU를 100% 사용하지 않는 한 Catalyst 4500에서는 CPU 사용률이 낮은 것으로 간주합니다.

Cisco IOS 소프트웨어 릴리스 12.2(25)EWA2 이상에서는 CPU 패킷 및 프로세스 처리 메커니즘과 계정을 개선했습니다. 따라서 Catalyst 4500 구축에서 이러한 릴리스를 사용하십시오.

## Catalyst 4500의 CPU 사용률이 높은 이유 식별

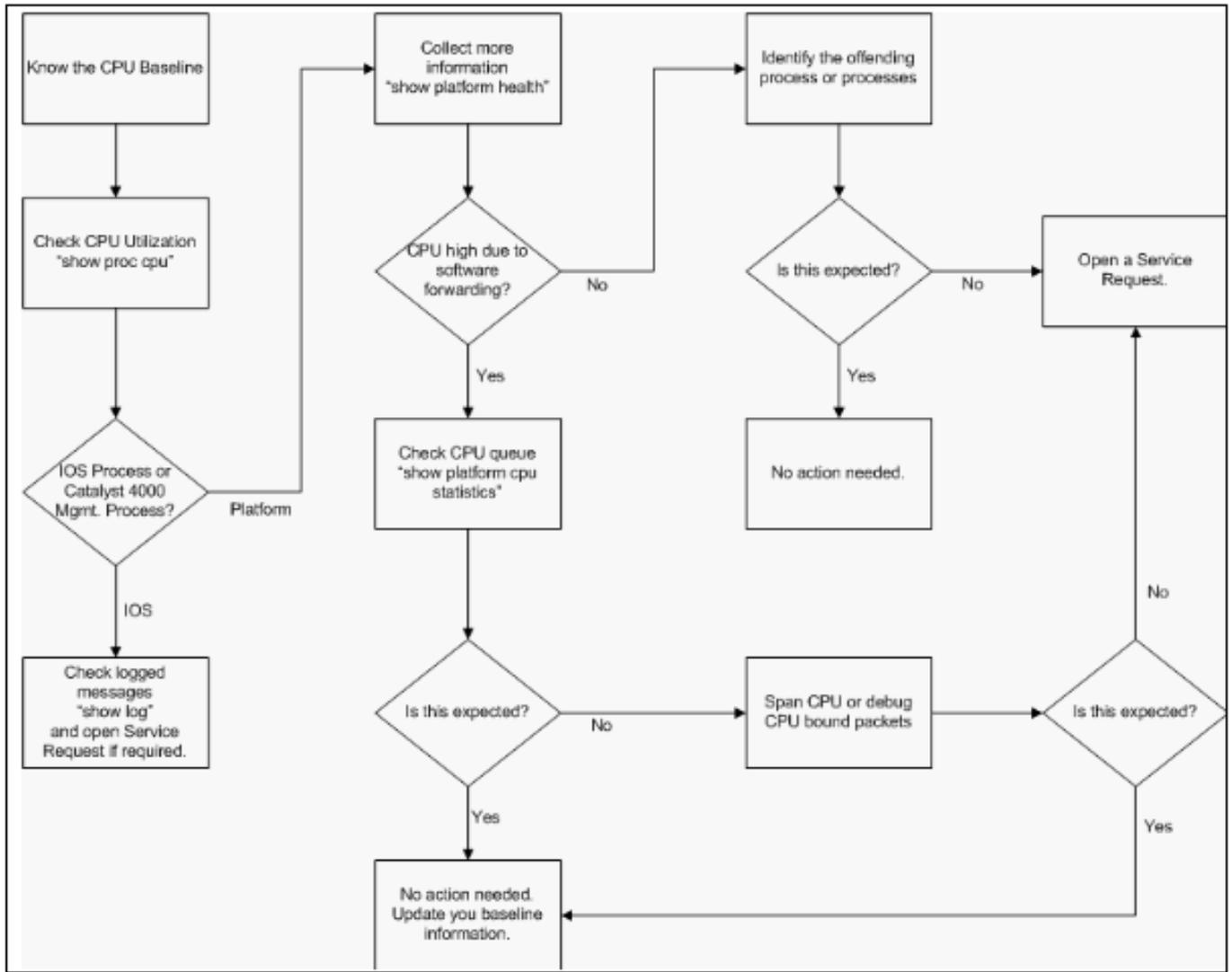
이제 Catalyst 4500 CPU 패킷 처리 아키텍처 및 설계를 이해했으므로 Catalyst 4500 CPU 사용률이 높은 이유를 계속 이해할 수 있습니다. Catalyst 4500에는 높은 CPU 사용률의 근본 원인을 식별하는 데 필요한 명령어와 톨이 있습니다. 이유를 확인한 후 관리자는 다음 작업 중 하나를 수행할 수 있습니다.

- 시정 조치 — 컨피그레이션 또는 네트워크 변경, 추가 분석을 위한 [Cisco Technical SupportService](#) 요청 생성이 포함될 수 있습니다.
- 작업 없음 - Catalyst 4500이 예상대로 수행됩니다. 필요한 모든 소프트웨어 패킷 전달 및 백그라운드 작업을 수행하기 위해 Supervisor Engine이 CPU 주기를 최대화하므로 CPU는 높은 CPU 사용률을 나타냅니다.

모든 경우에 시정 조치가 필요하지 않더라도 CPU 사용률이 높은 이유를 확인해야 합니다. 높은 CPU 사용률이 네트워크 문제의 한 증상일 수 있습니다. CPU 사용률을 낮추려면 해당 문제의 근본 원인을 해결해야 할 수 있습니다.

**그림 2**는 Catalyst 4500의 높은 CPU 사용률의 근본 원인을 파악하기 위해 사용할 수 있는 문제 해결 방법을 보여줍니다.

그림 2 - Catalyst 4500 스위치의 높은 CPU 사용률 문제 해결 방법론



Catalyst 4500 스위치의 높은 CPU 사용률 문제 해결 방법론

일반적인 문제 해결 단계는 다음과 같습니다.

1. CPU 사이클을 사용하는 Cisco IOS 프로세스를 식별하려면 show processes cpu 명령어를 실행합니다.
2. 플랫폼별 프로세스를 더 식별하기 위해 the show platform health command를 실행합니다.
3. 활성 프로세스가 k2CpuMan Review인 경우, CPU에 도달하는 트래픽의 유형을 식별하려면 show platform cpu packet statistics 명령을 실행합니다.

활동이 k2CpuMan Review process로 인한 것이 아닌 경우 4단계를 건너뛰고 5단계로 이동합니다.

4. 필요한 경우 Troubleshooting Tools를 사용하여 CPU에 도달하는 트래픽을 분석하여 CPU에 도달한 패킷을 식별합니다.

사용하는 트러블슈팅 툴의 예로는 CPU SPAN(Switched Port Analyzer)이 있습니다.

5. 이 문서와 [일반적인](#) 원인에 대한 [일반적인 높은 CPU 사용률 문제](#) 해결 섹션을 검토합니다.

그래도 근본 원인을 파악할 수 없는 경우 [Cisco 기술 지원](#)에 [문의하십시오](#).

## CPU 사용량 베이스라인

중요한 첫 번째 단계는 컨피그레이션 및 네트워크 설정을 위한 스위치의 CPU 사용률을 파악하는 것입니다. Catalyst 4500 스위치에서 CPU 사용률을 식별하려면 `show processes cpu` 명령을 사용합니다. 네트워크 설정에 구성을 추가하거나 네트워크 트래픽 패턴이 변경됨에 따라 기본 CPU 사용률을 지속적으로 업데이트해야 할 수 있습니다. [그림 2](#)는 이 요구 사항을 나타냅니다.

다음은 완전히 로딩된 Catalyst 4507R에서 출력되었습니다. 정상 상태 CPU는 약 32~38%이며, 이 스위치에 대한 관리 기능을 수행하는 데 필요합니다.

```
<#root>
```

```
Switch#
```

```
show processes cpu
```

```
CPU utilization for five seconds: 38%/1%; one minute: 32%; five minutes: 32%
```

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	0	63	0	0.00%	0.00%	0.00%	0	Chunk Manager
2	60	50074	1	0.00%	0.00%	0.00%	0	Load Meter
3	0	1	0	0.00%	0.00%	0.00%	0	Deferred Events
<i>!--- Output suppressed.</i>								
27	524	250268	2	0.00%	0.00%	0.00%	0	TTY Background
28	816	254843	3	0.00%	0.00%	0.00%	0	Per-Second Jobs
29	101100	5053	20007	0.00%	0.01%	0.00%	0	Per-minute Jobs
30	26057260	26720902	975	12.07%	11.41%	11.36%	0	Cat4k Mgmt HiPri
31	19482908	29413060	662	24.07%	19.32%	19.20%	0	Cat4k Mgmt LoPri
32	4468	162748	27	0.00%	0.00%	0.00%	0	Galios Reschedul
33	0	1	0	0.00%	0.00%	0.00%	0	Cisco IOS ACL Helper
34	0	2	0	0.00%	0.00%	0.00%	0	NAM Manager

5초 CPU 사용률은 다음과 같이 표시됩니다.

```
x%/y%
```

x%는 총 CPU 사용률을 나타내고 andy%는 인터럽트 레벨에서 사용된 CPU를 나타냅니다. Catalyst 4500 스위치의 문제를 해결할 때는 총 CPU 사용률에만 집중하십시오.

Catalyst 4500 스위치의 `show processes cpu` 명령어 이해

이 show processes cpu 출력을 보면 CPU를 사용하는 두 가지 프로세스, 즉 Cat4k Mgmt HiPriandCat4k Mgmt LoPri가 있습니다. 이 두 프로세스는 Catalyst 4500에서 필수 관리 기능을 수행하는 여러 플랫폼별 프로세스를 집계합니다. 이러한 프로세스는 컨트롤 플레인과 소프트웨어 전환 또는 처리가 필요한 데이터 패킷을 처리합니다.

Cat4k Mgmt HiPriandCat4k Mgmt LoPri 컨텍스트에서 어떤 플랫폼별 프로세스가 CPU를 사용하는지 확인하려면 show platform healthcommand를 실행합니다.

각 플랫폼별 프로세스에는 CPU의 대상/예상 사용률이 있습니다. 해당 프로세스가 대상 내에 있으면 CPU는 우선순위가 높은 상황에서 프로세스를 실행합니다. 이러한 프로세스에서 cpucmd 출력은 Cat4k 관리 HiPri에서 사용률을 계산합니다. 프로세스가 목표/예상 사용률을 초과하면 우선순위가 낮은 상황에서 프로세스가 실행됩니다. 이러한 프로세스에서 cpucmd 출력은 Cat4k Mgmt LoPri에 따른 추가 사용률을 계산합니다. ThisCat4k Mgmt LoPri는 백그라운드 프로세스 및 기타 낮은 우선 순위 프로세스(예: 일관성 검사 및 읽기 인터페이스 카운터)를 실행하는 데에도 사용됩니다. 이 메커니즘을 통해 CPU는 필요한 경우 높은 우선 순위 프로세스를 실행할 수 있으며, 남아 있는 유휴 CPU 주기는 낮은 우선 순위 프로세스에 사용됩니다. 목표 CPU 사용률을 소량으로 초과하거나 사용률의 일시적 급증은 조사가 필요한 문제를 나타내지 않습니다.

<#root>

Switch#

show platform health

	%CPU Target	%CPU Actual	RunTimeMax Target	RunTimeMax Actual	Priority Fg	Priority Bg	Average 5Sec	%CPU Min	%CPU Hour	Total CPU
Lj-poll	1.00									
0.02										
2	1	100	500	0						
0	0									
1:09										
GalChassisVp-review	3.00									
0.29										
10	3	100	500	0						
0	0									
11:15										
S2w-JobEventSchedule	10.00									
0.32										
10	7	100	500	0						
0	0									
10:14										
Stub-JobEventSchedule	10.00									
12.09										
10	6	100	500	14						

13 9

396:35  
StatValueMan Update 1.00

0.22

1 0 100 500 0

0 0

6:28  
Pim-review 0.10

0.00

1 0 100 500 0

0 0

0:22  
Ebm-host-review 1.00

0.00

8 0 100 500 0

0 0

0:05  
Ebm-port-review 0.10

0.00

1 0 100 500 0

0 0

0:01  
Protocol-aging-revie 0.20

0.00

2 0 100 500 0

0 0

0:00  
Ac1-Flattener e 1.00

0.00

10 0 100 500 0

0 0

0:00  
KxAc1PathMan create/ 1.00

0.00

10 5 100 500 0

0 0

0:39  
KxAc1PathMan update 2.00

0.00

10 0 100 500 0  
0 0  
0:00  
KxAc1PathMan reprogr 1.00  
0.00  
2 0 100 500 0  
0 0  
0:00  
TagMan-RecreateMtegR 1.00  
0.00  
10 0 100 500 0  
0 0  
0:00  
K2CpuMan Review 30.00  
10.19  
30 28 100 500 14  
13 9  
397:11  
K2Acce1PacketMan: Tx 10.00  
2.20  
20 0 100 500 2  
2 1  
82:06  
K2Acce1PacketMan: Au 0.10  
0.00  
0 0 100 500 0  
0 0  
0:00  
K2Ac1Man-taggedFlatA 1.00  
0.00  
10 0 100 500 0  
0 0  
0:00  
K2Ac1CamMan stale en 1.00  
0.00  
10 0 100 500 0  
0 0  
0:00  
K2Ac1CamMan hw stats 3.00

1.04

10 5 100 500 1

1 0

39:36

K2Ac1CamMan kx stats 1.00

0.00

10 5 100 500 0

0 0

13:40

K2Ac1CamMan Audit re 1.00

0.00

10 5 100 500 0

0 0

13:10

K2Ac1PolicerTableMan 1.00

0.00

10 1 100 500 0

0 0

0:38

K2L2 Address Table R 2.00

0.00

12 5 100 500 0

0 0

0:00

K2L2 New Static Addr 2.00

0.00

10 1 100 500 0

0 0

0:00

K2L2 New Multicast A 2.00

0.00

10 5 100 500 0

0 0

0:01

K2L2 Dynamic Address 2.00

0.00

10 0 100 500 0

0 0

0:00

K2L2 Vlan Table Revi 2.00

0.00

12 9 100 500 0  
0 0

0:01

K2 L2 Destination Ca 2.00

0.00

10 0 100 500 0  
0 0

0:00

K2PortMan Review 2.00

0.72

15 11 100 500 1  
1 0

37:22

Gigaport65535 Review 0.40

0.07

4 2 100 500 0  
0 0

3:38

Gigaport65535 Review 0.40

0.08

4 2 100 500 0  
0 0

3:39

K2Fib cam usage revi 2.00

0.00

15 0 100 500 0  
0 0

0:00

K2Fib IrmFib Review 2.00

0.00

15 0 100 500 0  
0 0

0:00

K2Fib Vrf Default Ro 2.00

0.00

15 0 100 500 0  
0 0

0:00  
K2Fib AdjRepop Revie 2.00  
0.00  
15 0 100 500 0  
0 0

0:00  
K2Fib Vrf Unpunt Rev 2.00  
0.01  
15 0 100 500 0  
0 0

0:23  
K2Fib Consistency Ch 1.00  
0.00  
5 2 100 500 0  
0 0

29:25  
K2FibAdjMan Stats Re 2.00  
0.30  
10 4 100 500 0  
0 0

6:21  
K2FibAdjMan Host Mov 2.00  
0.00  
10 4 100 500 0  
0 0

0:00  
K2FibAdjMan Adj Chan 2.00  
0.00  
10 0 100 500 0  
0 0

0:00  
K2FibMulticast Signa 2.00  
0.01  
10 2 100 500 0  
0 0

2:04  
K2FibMulticast Entry 2.00  
0.00  
10 7 100 500 0

0 0  
0:00  
K2FibMulticast Irm M 2.00  
0.00  
10 7 100 500 0  
0 0  
0:00  
K2FibFastDropMan Rev 2.00  
0.00  
7 0 100 500 0  
0 0  
0:00  
K2FibPbr route map r 2.00  
0.06  
20 5 100 500 0  
0 0  
16:42  
K2FibPbr flat acl pr 2.00  
0.07  
20 2 100 500 0  
0 0  
3:24  
K2FibPbr consolidati 2.00  
0.01  
10 0 100 500 0  
0 0  
0:24  
K2FibPerVlanPuntMan 2.00  
0.00  
15 4 100 500 0  
0 0  
0:00  
K2FibFlowCache flow 2.00  
0.01  
10 0 100 500 0  
0 0  
0:23  
K2FibFlowCache flow 2.00  
0.00

10	0	100	500	0
0	0			
0:00				
K2FibFlowCache adj r			2.00	
0.01				
10	0	100	500	0
0	0			
0:20				
K2FibFlowCache flow			2.00	
0.00				
10	0	100	500	0
0	0			
0:06				
K2MetStatsMan Review			2.00	
0.14				
5	2	100	500	0
0	0			
23:40				
K2FibMulticast MET S			2.00	
0.00				
10	0	100	500	0
0	0			
0:00				
K2QosDb1Man Rate DBL			2.00	
0.12				
7	0	100	500	0
0	0			
4:52				
IrmFibThrottler Thro			2.00	
0.01				
7	0	100	500	0
0	0			
0:21				
K2 VlanStatsMan Revi			2.00	
1.46				
15	7	100	500	2
2	1			
64:44				
K2 Packet Memory Dia			2.00	

0.00

15 8 100 500 0

1 1

45:46

K2 L2 Aging Table Re 2.00

0.12

20 3 100 500 0

0 0

7:22

RkiosPortMan Port Re 2.00

0.73

12 7 100 500 1

1 1

52:36

Rkios Module State R 4.00

0.02

40 1 100 500 0

0 0

1:28

Rkios Online Diag Re 4.00

0.02

40 0 100 500 0

0 0

1:15

RkiosIpPbr IrmPort R 2.00

0.02

10 3 100 500 0

0 0

2:44

RkiosAcIMan Review 3.00

0.06

30 0 100 500 0

0 0

2:35

MatMan Review 0.50

0.00

4 0 100 500 0

0 0

0:00

Slot 3 ILC Manager R 3.00  
0.00  
10 0 100 500 0  
0 0

0:00  
Slot 3 ILC S2wMan Re 3.00  
0.00  
10 0 100 500 0  
0 0

0:00  
Slot 4 ILC Manager R 3.00  
0.00  
10 0 100 500 0  
0 0

0:00  
Slot 4 ILC S2wMan Re 3.00  
0.00  
10 0 100 500 0  
0 0

0:00  
Slot 5 ILC Manager R 3.00  
0.00  
10 0 100 500 0  
0 0

0:00  
Slot 5 ILC S2wMan Re 3.00  
0.00  
10 0 100 500 0  
0 0

0:00  
Slot 6 ILC Manager R 3.00  
0.00  
10 0 100 500 0  
0 0

0:00  
Slot 6 ILC S2wMan Re 3.00  
0.00  
10 0 100 500 0  
0 0

```
0:00
Slot 7 ILC Manager R    3.00
```

```
0.00
    10      0  100  500   0
0  0
```

```
0:00
Slot 7 ILC S2wMan Re    3.00
```

```
0.00
    10      0  100  500   0
0  0
```

```
0:00
EthHoleLinecardMan(1    1.66
```

```
0.04
    10      0  100  500   0
0  0
```

```
1:18
EthHoleLinecardMan(2    1.66
```

```
0.02
    10      0  100  500   0
0  0
```

```
1:18
EthHoleLinecardMan(6    1.66
```

```
0.17
    10      6  100  500   0
0  0
```

```
6:38
-----
%CPU Totals          212.80
35.63
```

## Catalyst 4500 스위치의 show platform health 명령어 이해

theshow platform healthcommand는 개발 엔지니어에게만 관련된 많은 정보를 제공합니다. 높은 CPU 사용률을 트러블슈팅하려면 출력의 %CPU actual 열에서 더 높은 숫자를 찾습니다. 또한 1분 및 %CPUcolumn 1개 영역에서 해당 프로세스의 CPU 사용량을 확인하려면 해당 행의 오른쪽을 살펴봐야 합니다. 경우에 따라 프로세스가 일시적으로 최고조에 도달하지만 오랫동안 CPU를 유지시키지 않는 경우가 있습니다. 일시적으로 높은 CPU 사용률 중 일부는 하드웨어 프로그래밍 또는 프로그래밍 최적화 중에 발생합니다. 예를 들어, TCAM에서 대규모 ACL을 하드웨어 프로그래밍하는 동안의 CPU 사용률 급증은 정상입니다.

이 섹션의 how platform healthcommand 출력에서 Catalyst 4500 스위치의 show processes cpu Command를 이해하면 Stub-JobEventSchedule and theK2CpuMan Reviewprocesses는 더 많은 수의 CPU 사이클을 사용합니다. [표 2에서는](#) 이러한 how platformhealthcommand 출력에 나타나는 공통 플랫폼별 프로세스에 대한 몇 가지 기본 정보를 제공합니다.

표 2 – show platform health 명령어의 플랫폼별 프로세스에 대한 설명

플랫폼별 프로세스 이름	설명
Pim-review	새시/라인 카드 상태 관리
Ebm	이더넷 브리지 모듈(예: 에이징 및 모니터링)
Ac1-Flattener/K2Ac1Man	ACL 병합 프로세스
KxAclPathMan - PathTagMan-Review	ACL 상태 관리 및 유지 보수
K2CpuMan Review	소프트웨어 패킷 전달을 수행하는 프로세스 이 프로세스로 인해 CPU 사용률이 높은 경우 how platform cpu packet statistics 명령을 사용하여 CPU에 도달한 패킷을 조사합니다.
K2AccelPacketMan	CPU에서 보내는 패킷을 전송하기 위해 패킷 엔진과 상호 작용하는 드라이버
K2Ac1CamMan	QoS 및 보안 기능을 위해 입력 및 출력 TCAM 하드웨어 관리
K2Ac1PolicerTableMan	입력 및 출력 폴리서 관리
K2L2	Catalyst 4500 Cisco IOS 소프트웨어의 L2 전달 하위 시스템을 나타냅니다. 이러한 프로세스는 다양한 L2 테이블의 유지 보수를 담당합니다.
K2PortMan Review	다양한 포트 관련 프로그래밍 기능 관리
K2Fib	FIB <sup>1</sup> 관리
K2FibFlowCache	PBR <sup>2</sup> 캐시 관리
K2FibAdjMan	FIB 인접 테이블 관리
K2FibMulticast	멀티캐스트 FIB 항목 관리
K2MetStatsMan Review	MET <sup>3</sup> 통계 관리
K2QosDb1Man Review	QoS DBL <sup>4</sup> 관리
IrmFibThrotler Thro	IP 라우팅 모듈
K2 L2 Aging Table Re	L2 에이징 기능 관리
GalChassisVp-review	새시 상태 모니터링
S2w-JobEventSchedule	라인 카드 상태를 모니터링하기 위해 S2W <sup>5</sup> 프로토콜을 관리합니다.
Stub-JobEventSchedule	Stub ASIC 기반 라인 카드 모니터링 및 유지 보수
RkiosPortMan Port Re	포트 상태 모니터링 및 유지 보수
Rkios Module State R	라인 카드 모니터링 및 유지 보수
EthHoleLinecardMan	각 라인 카드에서 GBIC <sup>6</sup> 관리

<sup>1</sup>FIB = 전달 정보 데이터베이스

<sup>2</sup>PBR = 정책 기반 라우팅

<sup>3</sup>MET = 멀티캐스트 확장 테이블

<sup>4</sup>DBL = 동적 버퍼 제한.

<sup>5</sup>S2W = 직렬-유선

<sup>6</sup>GBIC = 기가비트 인터페이스 컨버터.

## 일반적인 높은 CPU 사용률 문제 해결

이 섹션에서는 Catalyst 4500 스위치의 몇 가지 일반적인 높은 CPU 사용률 문제를 다룹니다.

### 프로세스 전환 패킷으로 인한 높은 CPU 사용률

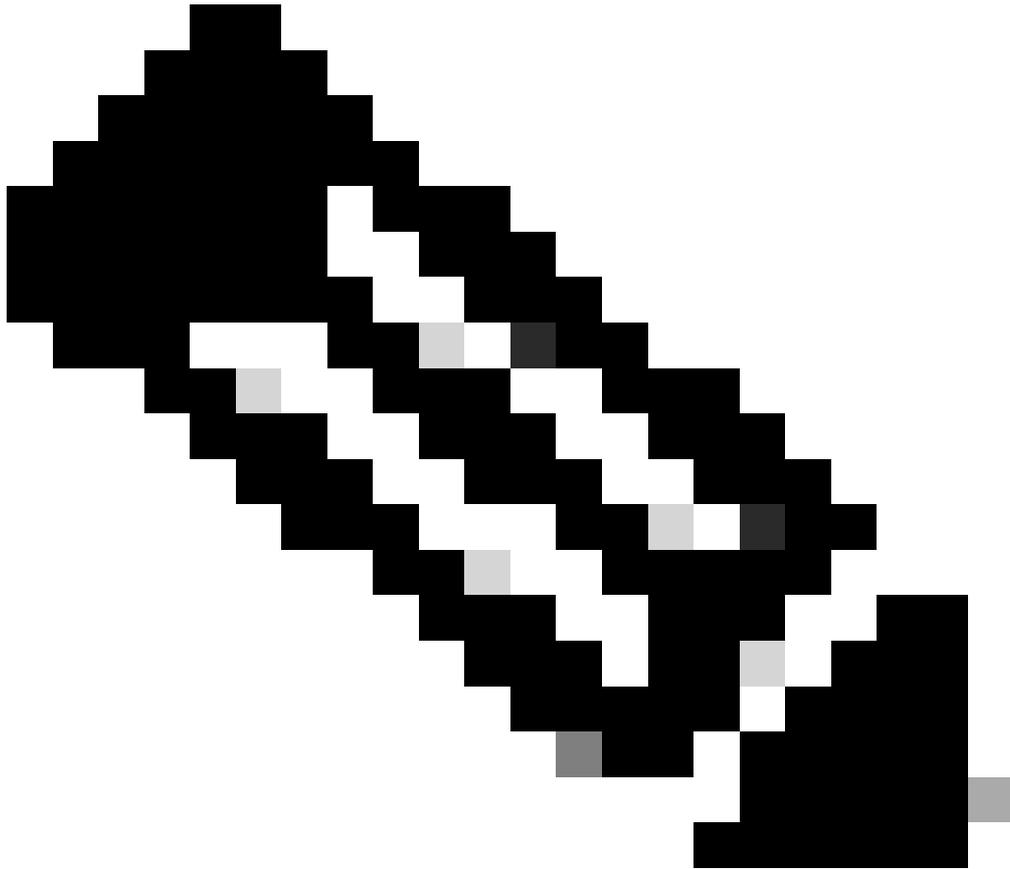
CPU 사용률이 높은 일반적인 이유 중 하나는 Catalyst 4500 CPU가 소프트웨어 전달 패킷 또는 제어 패킷의 패킷 프로세스로 사용 중이기 때문입니다. 소프트웨어 전달 패킷의 예로는 BPDU와 같은 IPX 또는 제어 패킷이 있습니다. 이러한 패킷 중 적은 수가 일반적으로 CPU로 전송됩니다. 그러나 지속적으로 많은 수의 패킷이 있을 경우 컨피그레이션 오류 또는 네트워크 이벤트를 나타낼 수 있습니다. 처리를 위해 패킷을 CPU로 전달하는 이벤트의 원인을 식별해야 합니다. 이러한 식별을 통해 높은 CPU 사용률 문제를 디버깅할 수 있습니다.

프로세스 전환 패킷으로 인해 CPU 사용률이 높아지는 일반적인 원인은 다음과 같습니다.

- [다수의 스페닝 트리 포트 인스턴스](#)
- [ICMP 리디렉션, 동일한 인터페이스에서 패킷 라우팅](#)
- [IPX 또는 AppleTalk 라우팅](#)
- [호스트 학습](#)
- [보안 ACL을 위한 하드웨어 리소스 부족\(TCAM\)](#)
- [ACL의 Theologkeyword](#)
- [레이어 2 전달 루프](#)

패킷이 CPU로 전환되는 다른 이유는 다음과 같습니다.

- MTU 프래그멘테이션 - 패킷 경로의 모든 인터페이스에 동일한 MTU가 있는지 확인합니다.
- 설정되지 않은 TCP 플래그가 있는 ACL
- IP 버전 6(IPv6) 라우팅 - 소프트웨어 스위칭 경로를 통해서만 지원됩니다.
- GRE — 소프트웨어 스위칭 경로를 통해서만 지원됩니다.
- 입력 또는 출력 라우터 ACL(RACL)의 트래픽 거부



참고: Cisco IOS Software 릴리스 12.1(13)EW1 이상에서는 이 기능이 속도 제한적입니다.

---

ACL의 인터페이스에서 `no ip unreachablesc` 명령을 실행합니다.

- 직접 연결된 다수의 호스트로 인해 과도한 ARP 및 DHCP 트래픽이 처리를 위해 CPU에 영향을 줌

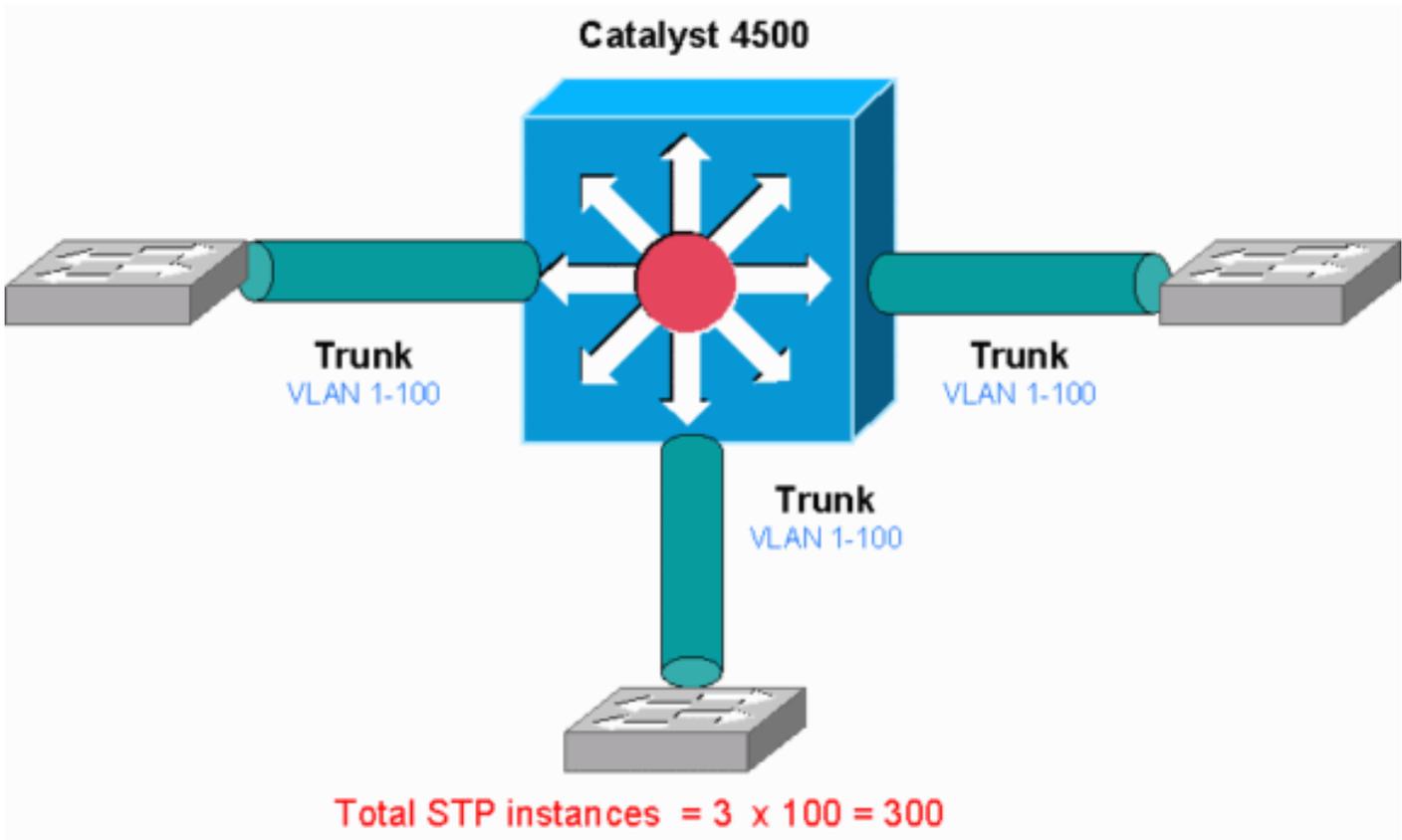
DHCP 공격이 의심되는 경우 DHCP 스누핑을 사용하여 특정 호스트 포트에서 DHCP 트래픽을 속도 제한합니다.

- 합법적이거나 오작동하는 엔드 스테이션에 의한 과도한 SNMP 폴링

다수의 스페닝 트리 포트 인스턴스

Catalyst 4500은 PVST+(Per VLAN Spanning Tree+) 모드에서 3,000개의 스페닝 트리 포트 인스턴스 또는 활성 포트를 지원합니다. Supervisor Engine II+, II+TS 및 Catalyst 4948을 제외한 모든 Supervisor Engine에서 지원됩니다. Supervisor Engine II+ 및 II+TS와 Catalyst 4948은 최대 1,500개의 포트 인스턴스를 지원합니다. 이러한 STP 인스턴스 권장 사항을 초과하면 스위치에서

높은 CPU 사용률을 보입니다.



Catalyst 4500

이 다이어그램에는 각각 1~100개의 VLAN을 전달하는 3개의 트렁크 포트가 있는 Catalyst 4500이 나와 있습니다. 이는 300개의 스페닝 트리 포트 인스턴스와 같습니다. 일반적으로 다음 공식을 사용하여 스페닝 트리 포트 인스턴스를 계산할 수 있습니다.

Total number of STP instances = Number of access ports + Sum of all VLANs that are carried in each of the trunks

다이어그램에는 액세스 포트가 없지만 3개의 트렁크에 1~100개의 VLAN이 있습니다.

Total number of STP instances = 0 + 100 + 100 + 100 = 300

1단계: 명령을 사용하여 Cisco IOS 프로세스 `show processes cpu` 를 확인합니다.

이 섹션에서는 높은 CPU 사용률 문제를 줄이기 위해 관리자가 사용하는 명령어를 검토합니다. `show processes cpu` 명령을 실행하면 두 개의 기본 프로세스인 `Cat4k Mgmt LoPriandSpanning` 트리에서 주로 CPU를 사용한다는 것을 확인할 수 있습니다. 이 정보만 사용하면 스페닝 트리 프로세스에서 CPU 사이클의 상당한 부분을 사용하게 됩니다.

<#root>

Switch#

show processes cpu

CPU utilization for five seconds: 74%/1%; one minute: 73%; five minutes: 50%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	4	198	20	0.00%	0.00%	0.00%	0	Chunk Manager
2	4	290	13	0.00%	0.00%	0.00%	0	Load Meter

!--- Output suppressed.

25	488	33	14787	0.00%	0.02%	0.00%	0	Per-minute Jobs
26	90656	223674	405	6.79%	6.90%	7.22%	0	Cat4k Mgmt HiPri

27	158796	59219	2681	32.55%	33.80%	21.43%	0	Cat4k Mgmt LoPri
----	--------	-------	------	--------	--------	--------	---	------------------

28	20	1693	11	0.00%	0.00%	0.00%	0	Galios Reschedul
29	0	1	0	0.00%	0.00%	0.00%	0	Cisco IOS ACL Helper
30	0	2	0	0.00%	0.00%	0.00%	0	NAM Manager

!--- Output suppressed.

41	0	1	0	0.00%	0.00%	0.00%	0	SFF8472
42	0	2	0	0.00%	0.00%	0.00%	0	AAA Dictionary R

43	78564	20723	3791	32.63%	30.03%	17.35%	0	Spanning Tree
----	-------	-------	------	--------	--------	--------	---	---------------

44	112	999	112	0.00%	0.00%	0.00%	0	DTP Protocol
45	0	147	0	0.00%	0.00%	0.00%	0	Ethchnl

2단계: show platform health 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

어떤 플랫폼별 프로세스가 CPU를 사용하는지 알아보려면 how **platform healthcommand**를 실행합니다. 이 출력에서 CPU 바인딩된 패킷을 처리하는 작업인 **K2CpuMan Reviewprocess**가 CPU를 사용하는 것을 볼 수 있습니다.

<#root>

Switch#

**show platform health**

%CPU	%CPU	RunTimeMax	Priority	Average	%CPU	Total						
		Target	Actual	Target	Actual	Fg	Bg	5Sec	Min	Hour	CPU	
<i>!--- Output suppressed.</i>												
TagMan-RecreateMtegR		1.00	0.00	10	0	100	500	0	0	0	0:00	
<b>K2CpuMan Review</b>		<b>30.00</b>	<b>37.62</b>	<b>30</b>	<b>53</b>	<b>100</b>	<b>500</b>	<b>41</b>	<b>33</b>	<b>1</b>	<b>2:12</b>	
K2Acce1PacketMan: Tx		10.00	4.95	20	0	100	500	5	4	0	0:36	
K2Acce1PacketMan: Au		0.10	0.00	0	0	100	500	0	0	0	0:00	
K2Ac1Man-taggedFlatA		1.00	0.00	10	0	100	500	0	0	0	0:00	

3단계: CPU 바인딩 트래픽의 유형을 식별하기 위해 트래픽을 수신하는 CPU 큐를 확인합니다.

어떤 CPU show platform cpu packet statistics 대기열이 CPU 바인딩 패킷을 수신하는지 확인하기 위해 명령을 실행합니다. 이 섹션의 출력은 제어 대기열이 많은 패킷을 수신하는 것을 보여줍니다. 표 1의 [정보](#)와 [1단계](#)에서 도출한 [결론](#)을 사용하십시오. CPU가 처리하는 패킷과 CPU 사용률이 높은 이유가 BPDU 처리 때문인지 확인할 수 있습니다.

<#root>

Switch#

show platform cpu packet statistics

!--- Output suppressed.

Total packet queues 16  
Packets Received by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Esmp	202760	196	173	128	28
Control	388623	2121	1740	598	16

Packets Dropped by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Control	17918	0	19	24	3

4단계: 근본 원인 파악

명령을 show spanning-tree summary 실행합니다. BPDU 수신이 스페닝 트리 포트 인스턴스의 수가 많기 때문인지 확인할 수 있습니다. 출력은 근본 원인을 명확하게 식별합니다.

<#root>

Switch#

show spanning-tree summary

```

Switch is in pvst mode
Root bridge for: none
Extended system ID          is enabled
Portfast Default            is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default          is disabled
EtherChannel misconfig guard is enabled
UplinkFast                  is disabled
BackboneFast                is disabled
Configured Pathcost method used is short

```

!--- Output suppressed.

Name	Blocking	Listening	Learning	Forwarding	STP	Active
-----	-----	-----	-----	-----	-----	-----
2994 vlans	0	0	0	5999		5999

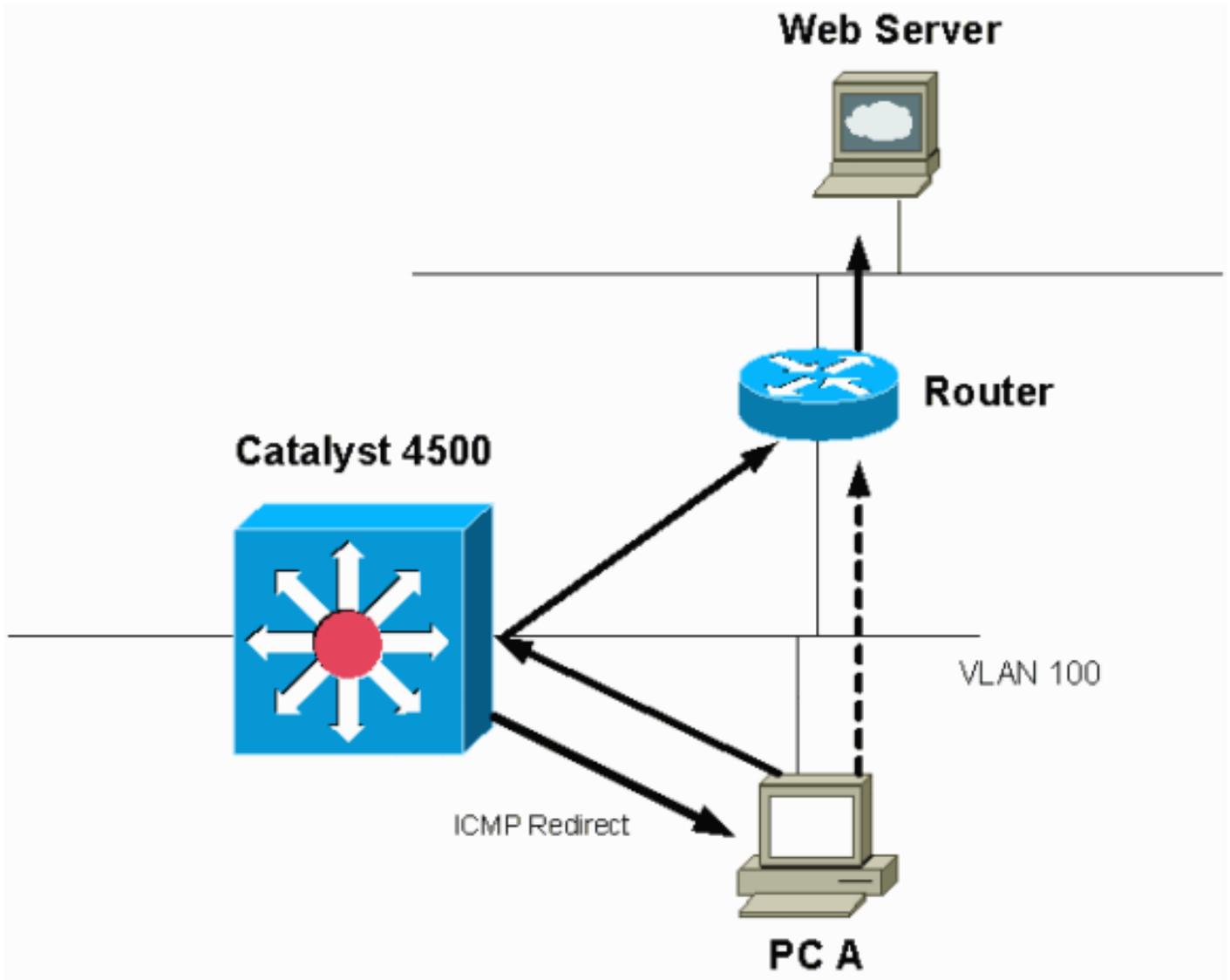
PVST+ 모드 컨피그레이션을 사용하는 VLAN이 많이 있습니다. 이 문제를 해결하려면 STP 모드를 MST(Multiple Spanning Tree)로 변경합니다. 모든 트렁크 포트에서 많은 수의 VLAN이 전달되어 STP 인스턴스의 수가 많은 경우도 있습니다. 이 경우, STP 활성 포트 수를 권장 값 이하로 낮추기 위해 트렁크에서 필요하지 않은 VLAN을 수동으로 정리합니다.

**팁:** IP 전화 포트를 트렁크 포트에 구성하지 마십시오. 이는 일반적인 컨피그레이션 오류입니다. 음성 VLAN 컨피그레이션으로 IP 전화기 포트를 구성합니다. 이 컨피그레이션은 의사(pseudo) 트렁크를 생성하지만 불필요한 VLAN을 수동으로 정리하지 않아도 됩니다. 음성 포트를 구성하는 방법에 대한 자세한 내용은 [음성 인터페이스 구성소프트웨어](#) 컨피그레이션 가이드를 참조하십시오. Cisco 이외의 IP 전화기는 이 음성 VLAN 또는 보조 VLAN 컨피그레이션을 지원하지 않습니다. Cisco 이외의 IP 전화기로 포트를 수동으로 정리해야 합니다.

#### ICMP 리디렉션, 동일한 인터페이스에서 패킷 라우팅

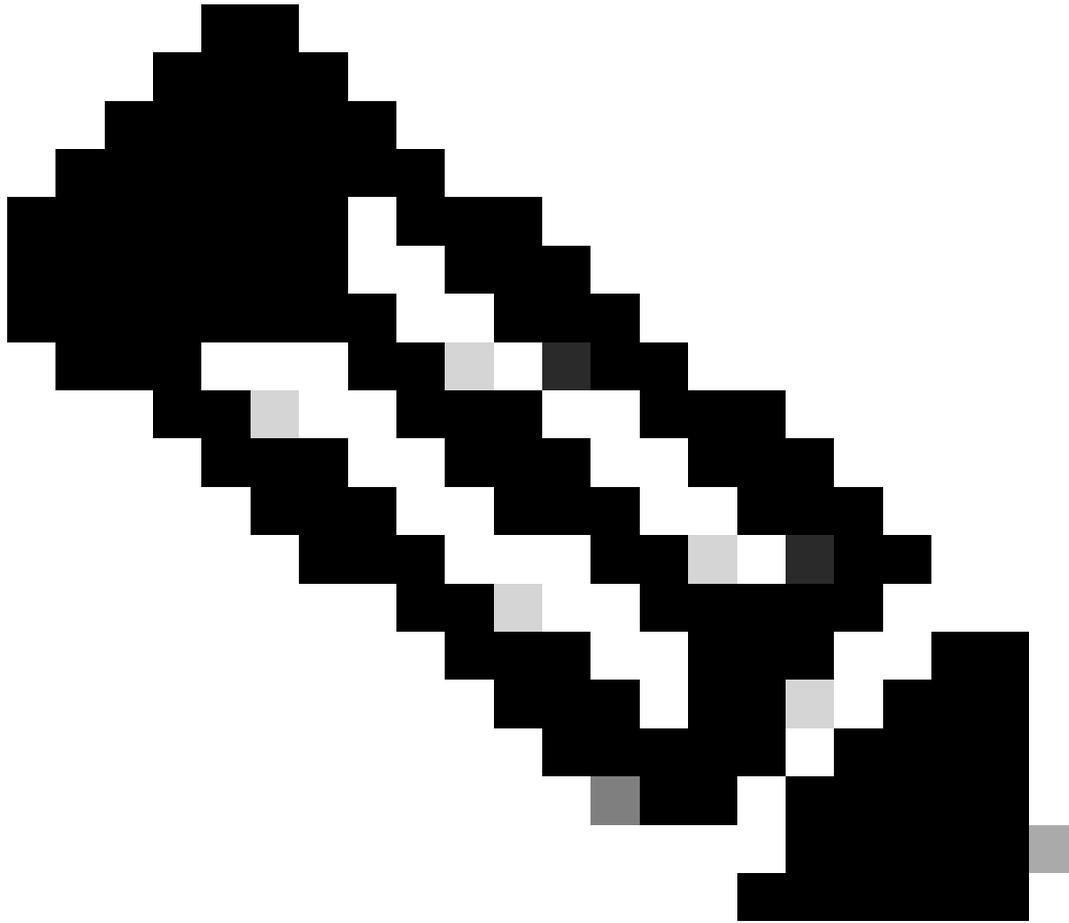
동일한 인터페이스에서의 패킷 라우팅 또는 동일한 L3 인터페이스에서의 트래픽 인그레스 및 이그레스는 스위치에 의한 ICMP 리디렉션을 초래할 수 있습니다. 최종 대상에 대한 다음 홉 디바이스가 전송 디바이스와 동일한 서브넷에 있음을 스위치가 인식하면 스위치는 소스에 대한 ICMP 리디렉션을 생성합니다. 리디렉션 메시지는 패킷을 다음 홉 디바이스로 직접 전송하도록 소스에 표시합니다. 이 메시지는 다음 홉 디바이스가 대상에 대한 더 나은 경로(해당 스위치보다 하나 적은 홉의 경로)를 가지고 있음을 나타냅니다.

이 섹션의 다이어그램에서 PC A는 웹 서버와 통신합니다. PC A의 기본 게이트웨이는 VLAN 100 인터페이스 IP 주소를 가리킵니다. 그러나 Catalyst 4500이 대상에 연결할 수 있도록 하는 다음 홉 라우터는 PC A와 동일한 서브넷에 있습니다. 이 경우 가장 좋은 경로는 "라우터"로 직접 전송하는 것입니다. Catalyst 4500은 PC A에 ICMP 리디렉션 메시지를 전송합니다. 메시지는 PC A에 Catalyst 4500 대신 라우터를 통해 웹 서버로 전송되는 패킷을 보낼 것을 지시합니다. 그러나 대부분의 경우 엔드 디바이스는 ICMP 리디렉션에 응답하지 않습니다. 응답 부족으로 인해 Catalyst 4500은 인그레스 패킷과 동일한 인터페이스를 통해 Catalyst가 전달하는 모든 패킷에 대해 이러한 ICMP 리디렉션을 생성하는 데 많은 CPU 사이클을 소비하게 됩니다.



기본적으로 *ICMP* 리디렉션은 활성화되어 있습니다.

기본적으로 *ICMP* 리디렉션은 활성화되어 있습니다. 비활성화하려면 명령을 `ip icmp redirects` 사용합니다. 관련 *SVI* 또는 *L3* 인터페이스에서 명령어를 실행합니다.



참고: 기본 `ip icmp redirects` 명령이므로 명령 출력에 `show running-configuration` 표시되지 않습니다.

---

1단계: 명령을 사용하여 Cisco IOS 프로세스를 `show processes cpu` 확인합니다.

명령을 `show processes cpu` 실행합니다. 두 가지 주요 프로세스인 `Cat4k Mgmt LoPriandIP Input`은 주로 CPU를 사용합니다. 이 정보만 사용하면 IP 패킷의 프로세스가 CPU의 상당한 부분을 차지한다는 것을 알 수 있습니다.

<#root>

Switch#

show processes cpu

CPU utilization for five seconds: 38%/1%; one minute: 32%; five minutes: 32%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	0	63	0	0.00%	0.00%	0.00%	0	Chunk Manager
2	60	50074	1	0.00%	0.00%	0.00%	0	Load Meter
3	0	1	0	0.00%	0.00%	0.00%	0	Deferred Events

!--- Output suppressed.

27	524	250268	2	0.00%	0.00%	0.00%	0	TTY Background
28	816	254843	3	0.00%	0.00%	0.00%	0	Per-Second Jobs
29	101100	5053	20007	0.00%	0.01%	0.00%	0	Per-minute Jobs
30	26057260	26720902	975	5.81%	6.78%	5.76%	0	Cat4k Mgmt HiPri

31	19482908	29413060	662	19.64%	18.20%	20.48%	0	Cat4k Mgmt LoPri
----	----------	----------	-----	--------	--------	--------	---	------------------

!--- Output suppressed. show platform health 35 60 902 0 0.00% 0.00% 0.00% 0 DHCP Snooping

36	504625304	645491491	781	72.40%	72.63%	73.82%	0	IP Input
----	-----------	-----------	-----	--------	--------	--------	---	----------

2단계: 명령을 사용하여 Catalyst 4500 관련 프로세스를 show platform health 확인합니다.

명령의 출력에서 **show platform health**는 CPU 바인딩된 패킷을 처리하기 위한 CPU 사용을 확인합니다.

<#root>

Switch#

**show platform health**

%CPU	%CPU	RunTimeMax	Priority	Average %CPU	Total						
		Target	Actual	Target	Actual	Fg	Bg	5Sec	Min	Hour	CPU

--- Output suppressed.

TagMan-RecreateMtegR	1.00	0.00	10	0	100	500	0	0	0	0:00
K2CpuMan Review	330.00	19.18	150	79	25	500	20	19	18	5794:08
K2Acce1PacketMan: Tx	10.00	4.95	20	0	100	500	5	4	0	0:36
K2Acce1PacketMan: Au	0.10	0.00	0	0	100	500	0	0	0	0:00
K2Ac1Man-taggedFlatA	1.00	0.00	10	0	100	500	0	0	0	0:00

3단계: CPU 바인딩 트래픽의 유형을 식별하기 위해 트래픽을 수신하는 CPU 큐를 확인합니다.

어떤 CPU **show platform cpu packet statistics** 대기열이 CPU 바인딩 패킷을 수신하는지 확인하기 위해 명령을 실행합니다. L3 Fwd Lowqueue는 상당히 많은 트래픽을 수신합니다.

<#root>

Switch#

**show platform cpu packet statistics**

!--- Output suppressed.

Packets Received by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Esmp	48613268	38	39	38	39
Control	142166648	74	74	73	73
Host Learning	1845568	2	2	2	2
L3 Fwd High	17	0	0	0	0
L3 Fwd Medium	2626	0	0	0	0
<b>L3 Fwd Low</b>	<b>4717094264</b>	<b>3841</b>	<b>3879</b>	<b>3873</b>	<b>3547</b>
L2 Fwd Medium	1	0	0	0	0
L3 Rx High	257147	0	0	0	0
L3 Rx Low	5325772	10	19	13	7
RPF Failure	155	0	0	0	0
ACL fwd(snooping)	65604591	53	54	54	53
ACL log, unreach	11013420	9	8	8	8

#### 4단계: 근본 원인 파악

이 경우, CPU에 영향을 주는 트래픽을 확인하기 위해 CPU SPAN을 사용합니다. CPU SPAN에 대한 자세한 내용은 이 문서의 [Tool 1: Monitor the CPU Traffic with SPAN—Cisco IOS Software Release 12.1\(19\)EW and Later](#) section을 참조하십시오. 이 명령을 사용하여 트래픽 분석 및 컨피그레이션을 show running-configuration 완료합니다. 이 경우 패킷은 동일한 인터페이스를 통해 라우팅되므로 각 패킷에 대해 ICMP 리디렉션 문제가 발생합니다. 이 근본 원인은 Catalyst 4500의 높은 CPU 사용률의 일반적인 이유 중 하나입니다.

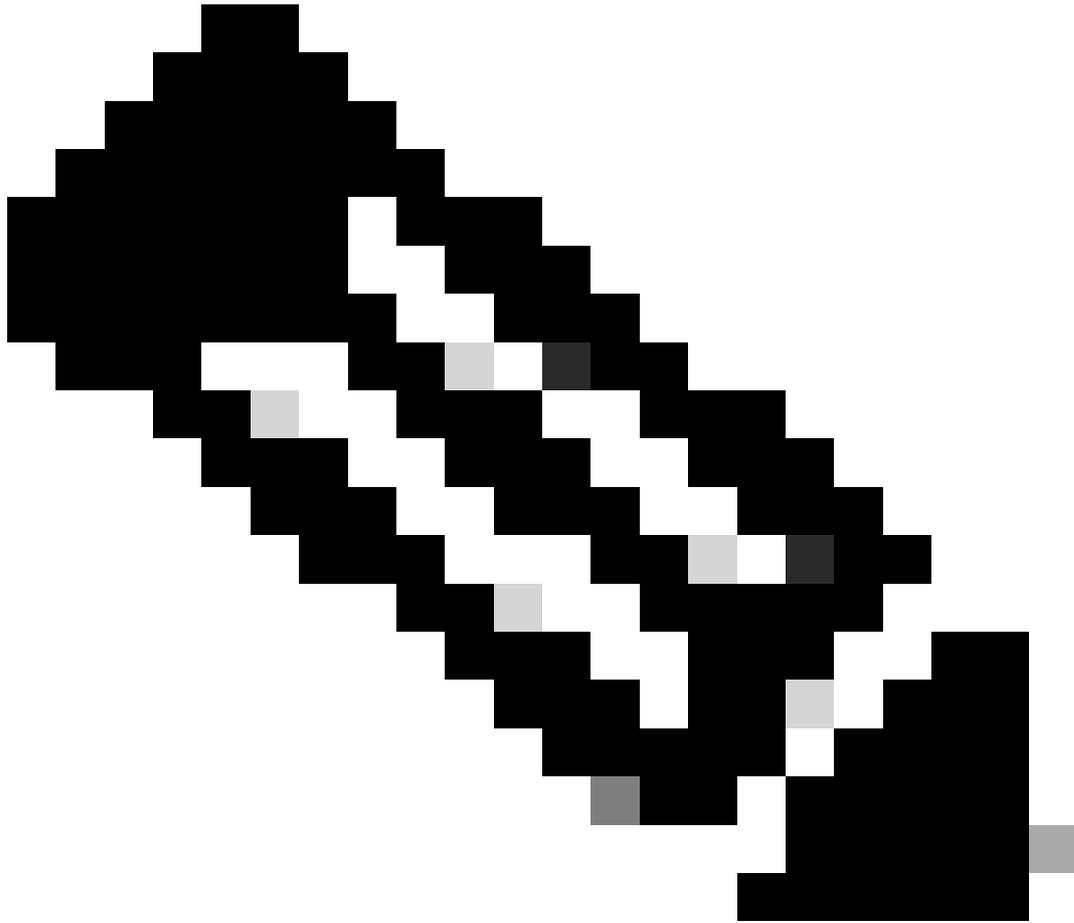
소싱 디바이스가 Catalyst 4500에서 전송하는 ICMP 리디렉션에 대해 작업을 수행하고 대상에 대한 다음 홉을 변경할 것으로 예상할 수 있습니다. 그러나 모든 디바이스가 ICMP 리디렉션에 응답하는 것은 아닙니다. 디바이스가 응답하지 않으면 Catalyst 4500은 스위치가 전송 디바이스에서 수신하는 모든 패킷에 대해 리디렉션을 전송해야 합니다. 이러한 리디렉션은 많은 CPU 리소스를 소비할 수 있습니다. 해결 방법은 ICMP 리디렉션을 비활성화하는 것입니다. 인터페이스 아래 no ip redirects에서 명령을 실행합니다.

이 시나리오는 보조 IP 주소도 구성된 경우 발생할 수 있습니다. 보조 IP 주소를 활성화하면 IP 리디렉션이 자동으로 비활성화됩니다. IP 리디렉션을 수동으로 활성화하지 마십시오.

이 [ICMP 리디렉션, 동일한 인터페이스의 라우팅 패킷](#)이 표시했듯이 대부분의 엔드 디바이스는 ICMP 리디렉션에 응답하지 않습니다. 따라서 일반적으로 해당 기능을 비활성화합니다.

#### IPX 또는 AppleTalk 라우팅

Catalyst 4500은 소프트웨어 전달 경로를 통해서만 IPX 및 AppleTalk 라우팅을 지원합니다. 이러한 프로토콜의 컨피그레이션에서는 CPU 사용률이 더 높은 것이 정상입니다.



**참고:** 동일한 VLAN에서 IPX 및 AppleTalk 트래픽을 스위칭할 때는 프로세스 스위칭이 필요하지 않습니다. 라우팅해야 하는 패킷에만 소프트웨어 경로 전달이 필요합니다.

---

1단계: 명령을 사용하여 Cisco IOS 프로세스를 `show processes cpu` 확인합니다.

어떤 Cisco `show processes cpu` IOS 프로세스에서 CPU를 사용하는지 확인하려면 명령을 실행합니다. 이 명령 출력에서 상위 프로세스는 `Cat4k Mgmt LoPri`입니다.

<#root>

witch#

show processes cpu

CPU utilization for five seconds: 87%/10%; one minute: 86%; five minutes: 87%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	4	53	75	0.00%	0.00%	0.00%	0	Chunk Manager

!--- Output suppressed.

25	8008	1329154	6	0.00%	0.00%	0.00%	0	Per-Second Jobs
26	413128	38493	10732	0.00%	0.02%	0.00%	0	Per-minute Jobs
27	148288424	354390017	418	2.60%	2.42%	2.77%	0	Cat4k Mgmt HiPri

28	285796820	720618753	396	50.15%	59.72%	61.31%	0	Cat4k Mgmt LoPri
----	-----------	-----------	-----	--------	--------	--------	---	------------------

2단계: 명령을 사용하여 Catalyst 4500 관련 프로세스를 show platform health 확인합니다.

이 명령의 show platform health출력에서는 CPU 바인딩 패킷을 처리하기 위한 CPU 사용을 확인합니다.

<#root>

Switch#

show platform health

%CPU	%CPU	RunTimeMax	Priority	Average %CPU	Total				
Target	Actual	Target	Actual	Fg	Bg	5Sec	Min	Hour	CPU

!--- Output suppressed.

```

TagMan-RecreateMtegR  1.00  0.00  10  4 100 500  0  0  0  0:00

K2CpuMan Review      30.00 27.39  30  53 100 500  42 47  42 4841:

K2Acce1PacketMan: Tx 10.00  8.03  20  0 100 500  21 29  26 270:4

```

3단계: CPU 바인딩 트래픽의 유형을 식별하려면 트래픽을 수신하는 CPU 큐를 확인합니다.

CPU에 도달하는 트래픽의 유형을 확인하려면 명령을 `show platform cpu packet statistics` 실행합니다.

```
<#root>
```

```
Switch#
```

```
show platform cpu packet statistics
```

*!--- Output suppressed.*

Packets Received by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
EsmP	48613268	38	39	38	39
Control	142166648	74	74	73	73
Host Learning	1845568	2	2	2	2
L3 Fwd High	17	0	0	0	0
L3 Fwd Medium	2626	0	0	0	0
L3 Fwd Low	1582414	1	1	1	1
L2 Fwd Medium	1	0	0	0	0
<b>L2 Fwd Low</b>	<b>576905398</b>	<b>1837</b>	<b>1697</b>	<b>1938</b>	<b>1515</b>

L3 Rx High	257147	0	0	0	0
L3 Rx Low	5325772	10	19	13	7
RPF Failure	155	0	0	0	0
ACL fwd(snooping)	65604591	53	54	54	53
ACL log, unreach	11013420	9	8	8	8

#### 4단계: 근본 원인 파악

관리자가 IPX 또는 AppleTalk 라우팅을 구성했으므로 근본 원인을 쉽게 식별할 수 있어야 합니다. 그러나 확인하려면 CPU 트래픽을 SPAN하고 표시되는 트래픽이 예상 트래픽인지 확인합니다. CPU SPAN에 대한 자세한 내용은 이 문서의 [Tool 1: Monitor the CPU Traffic with SPAN—Cisco IOS Software Release 12.1\(19\)EW and Later](#) section을 참조하십시오.

이 경우 관리자는 베이스라인 CPU를 현재 값으로 업데이트해야 합니다. Catalyst 4500 CPU는 CPU가 소프트웨어 전환 패킷을 처리할 때 예상대로 작동합니다.

#### 호스트 학습

MAC 주소가 MAC 주소 테이블에 없는 경우 Catalyst 4500은 다양한 호스트의 MAC 주소를 학습합니다. 스위칭 엔진은 새 MAC 주소가 포함된 패킷의 복사본을 CPU에 전달합니다.

모든 VLAN 인터페이스(레이어 3)는 새시 기본 하드웨어 주소를 MAC 주소로 사용합니다. 따라서 MAC 주소 테이블에 항목이 없으며 이러한 VLAN 인터페이스로 향하는 패킷은 처리를 위해 CPU로 전송되지 않습니다.

스위치에서 학습할 새 MAC 주소의 수가 너무 많으면 CPU 사용률이 높아질 수 있습니다.

1단계: 명령을 사용하여 Cisco IOS 프로세스를 `show processes cpu` 확인합니다.

어떤 `show processes cpu` Cisco IOS 프로세스가 CPU를 사용하는지 확인하기 위해 명령을 실행합니다. 이 명령 출력에서 상위 프로세스는 `Cat4k Mgmt LoPri`입니다.

<#root>

Switch#

`show processes cpu`

```
CPU utilization for five seconds: 89%/1%; one minute: 74%; five minutes: 71%
PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min TTY Process
  1             4         53       75  0.00%  0.00%  0.00%  0 Chunk Manager
```

!--- Output suppressed.

25	8008	1329154	6	0.00%	0.00%	0.00%	0	Per-Second Jobs
26	413128	38493	10732	0.00%	0.02%	0.00%	0	Per-minute Jobs
27	148288424	354390017	418	26.47%	10.28%	10.11%	0	Cat4k Mgmt HiPri

28	285796820	720618753	396	52.71%	56.79%	55.70%	0	Cat4k Mgmt LoPri
----	-----------	-----------	-----	--------	--------	--------	---	------------------

2단계: show platform health 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

thesaw **platform healthcommand**의 출력에서는 CPU 바인딩된 패킷을 처리하기 위해 CPU를 사용함을 확인합니다.

<#root>

Switch#

**show platform health**

	%CPU Target	%CPU Actual	RunTimeMax Target	RunTimeMax Actual	Priority Fg	Priority Bg	Average 5Sec	%CPU Min	%CPU Hour	Total CPU
--	----------------	----------------	----------------------	----------------------	----------------	----------------	-----------------	-------------	--------------	--------------

!--- Output suppressed.

TagMan-RecreateMtegR	1.00	0.00	10	4	100	500	0	0	0	0:00
----------------------	------	------	----	---	-----	-----	---	---	---	------

K2CpuMan Review	30.00	46.88	30	47	100	500	30	29	21	265:01
-----------------	-------	-------	----	----	-----	-----	----	----	----	--------

K2AccelPacketMan: Tx	10.00	8.03	20	0	100	500	21	29	26	270:4
----------------------	-------	------	----	---	-----	-----	----	----	----	-------

3단계: CPU 바인딩 트래픽의 유형을 식별하려면 트래픽을 수신하는 CPU 큐를 확인합니다.

CPU에 도달하는 트래픽의 유형을 확인하려면 `show platform cpu packet statistics` 명령을 실행합니다.

<#root>

Switch#

`show platform cpu packet statistics`

*!--- Output suppressed.*

Packets Received by Packet Queue

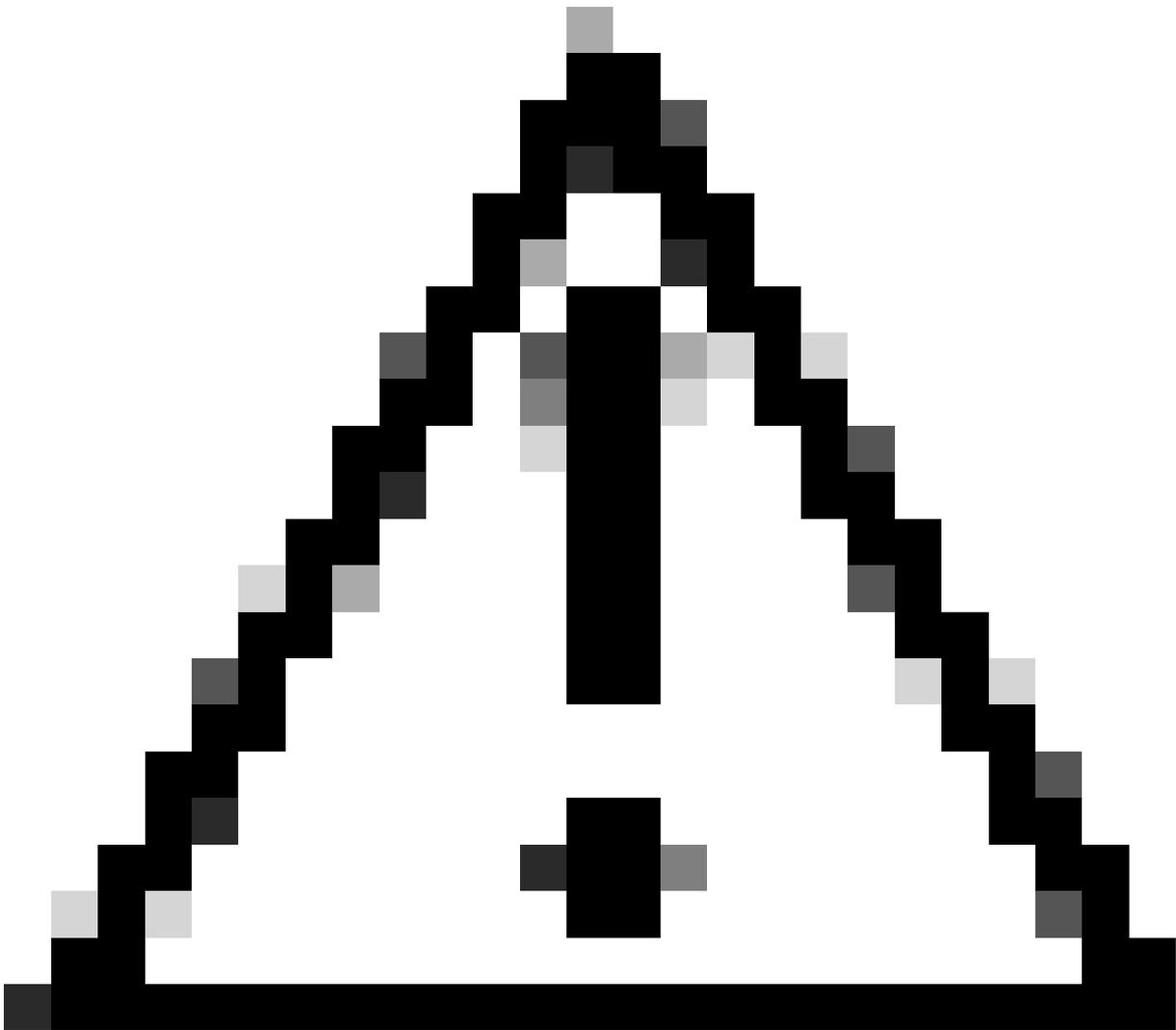
Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Ecmp	48613268	38	39	38	39
Control	142166648	74	74	73	73
<b>Host Learning</b>	<b>1845568</b>	<b>1328</b>	<b>1808</b>	<b>1393</b>	<b>1309</b>
L3 Fwd High	17	0	0	0	0
L3 Fwd Medium	2626	0	0	0	0
L3 Fwd Low	1582414	1	1	1	1
L2 Fwd Medium	1	0	0	0	0
L2 Fwd Low	576905398	37	7	8	5
L3 Rx High	257147	0	0	0	0
L3 Rx Low	5325772	10	19	13	7
RPF Failure	155	0	0	0	0
ACL fwd(snooping)	65604591	53	54	54	53
ACL log, unreachable	11013420	9	8	8	8

4단계: 근본 원인 파악

명령의 출력 `show platform health`은 CPU에 많은 새 MAC 주소가 표시됨을 보여줍니다. 이러한 상황은 종종 네트워크 토폴로지 불안정의 결과입니다. 예를 들어 스페닝 트리 토폴로지가 변경되면 스위치는 TCN(Topology Change Notifications)을 생성합니다. TCN 문제

는 PVST+ 모드에서 에이징 타임을 15초로 줄입니다. MAC 주소 항목은 해당 기간 내에 주소가 확인되지 않으면 플러시됩니다. RSTP(Rapid STP) (IEEE 802.1w) 또는 MST(IEEE 802.1s)의 경우 TCN이 다른 스위치에서 오면 항목이 즉시 만료됩니다. 이 기간이 만료되면 MAC 주소가 새로 학습됩니다. 토폴로지 변경이 드물다면 이는 큰 문제가 아닙니다. 그러나 플래핑 링크, 결함있는 스위치 또는 PortFast에 대해 활성화되지 않은 호스트 포트로 인해 토폴로지 변경이 지나치게 많이 발생할 수 있습니다. 많은 수의 MAC 테이블이 플러시되고 후속 재학습이 발생할 수 있습니다. 근본 원인 식별의 다음 단계는 네트워크 문제를 해결하는 것입니다. 스위치는 예상대로 작동하고 호스트 주소 학습을 위해 패킷을 CPU로 전송합니다. 과도한 TCN을 초래하는 결함이 있는 디바이스를 식별하고 수정합니다.

네트워크에는 버스트로 트래픽을 전송하는 많은 디바이스가 있을 수 있으며, 이로 인해 MAC 주소가 만료되어 나중에 스위치에서 다시 학습됩니다. 이 경우 약간의 여유를 제공하기 위해 MAC 주소 테이블 에이징 시간을 늘립니다. 에이징 시간이 더 길면 스위치는 시간이 경과하기 전에 디바이스 MAC 주소를 더 오랫동안 테이블에 유지합니다.



주의: 신중하게 고려한 후에만 이 나이 차이를 변경합니다. 네트워크에 이동하는 디바이스가 있는 경우 이러한 변경으로 인해 트래픽 블랙홀이 발생할 수 있습니다.

보안 ACL을 위한 하드웨어 리소스 부족(TCAM)

Catalyst 4500은 Cisco TCAM을 사용하여 구성된 ACL을 프로그래밍합니다. TCAM을 사용하면 하드웨어 전달 경로에 ACL을 적용할 수 있습니다. 전달 경로에 ACL이 있든 없든 스위치의 성능에는 영향을 주지 않습니다. ACL 조회의 성능은 회선 속도이므로 ACL의 크기에도 불구하고 성능은 일정합니다. 그러나 TCAM은 한정된 리소스입니다. 따라서 과도한 수의 ACL 항목을 구성할 경우 TCAM 용량을 초과하게 됩니다. 표 3은 각 Catalyst 4500 Supervisor Engine 및 스위치에서 사용 가능한 TCAM 리소스의 수를 보여줍니다.

표 3 - Catalyst 4500 Supervisor Engines/스위치의 TCAM 용량

제품	기능 TCAM(방향별)	QoS TCAM(방향별)
Supervisor Engine II+/II+TS	1,024 마스크가 포함된 8,192개 항목	1,024 마스크가 포함된 8,192개 항목
Supervisor Engine III/IV/V 및 Catalyst 4948	2,048 마스크가 포함된 16,384개 항목	2,048 마스크가 포함된 16,384개 항목
Supervisor Engine V-10GE 및 Catalyst 4948-10GE	16,384 마스크가 포함된 16,384개 항목	16,384 마스크가 포함된 16,384개 항목

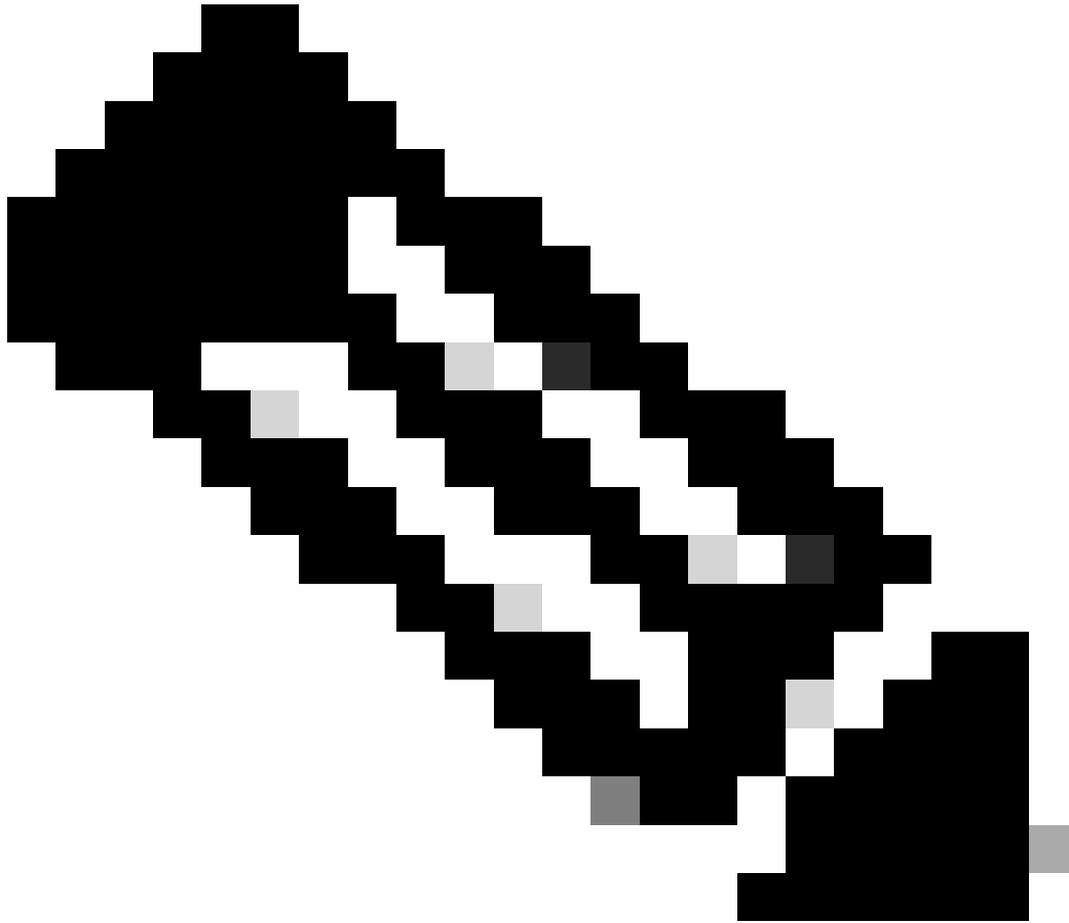
스위치는 RAACL 및 VAACL(VLAN ACL)과 같은 보안 ACL을 프로그래밍하기 위해 TCAM 기능을 사용합니다. 또한 스위치는 동적 ACL을 위한 IPSG(IP Source Guard)와 같은 보안 기능을 위해 TCAM 기능을 사용합니다. 분류 및 폴리서 ACL을 프로그래밍하기 위해 스위치는 QoS TCAM을 사용합니다.

보안 ACL을 프로그래밍하는 동안 Catalyst 4500에 TCAM 리소스가 부족해지면 소프트웨어 경로를 통해 ACL의 일부 애플리케이션이 적용됩니다. 이러한 ACE에 영향을 주는 패킷은 소프트웨어에서 처리되므로 CPU 사용률이 높아집니다. ACL은 위에서 아래로 프로그래밍됩니다. 즉, ACL이 TCAM에 맞지 않을 경우 ACL의 맨 아래 부분에 있는 ACE가 TCAM에서 프로그래밍되지 않을 가능성이 높습니다.

이 경고 메시지는 TCAM 오버플로 발생 시 나타납니다.

```
%C4K_HWACLMAN-4-ACLHWPROGERRREASON: (Suppressed 1times) Input(null, 12/Normal)
Security: 140 - insufficient hardware TCAM masks.
%C4K_HWACLMAN-4-ACLHWPROGERR: (Suppressed 4 times) Input Security: 140 - hardware TCAM
Limit, some packet processing can be software switched.
```

이 오류 메시지는 how loggingcommand 출력에서 확인할 수 있습니다. 이 메시지는 일부 소프트웨어 처리가 발생할 수 있으며 결과적으로 CPU 사용률이 높을 수 있음을 최종적으로 나타냅니다.



참고: 큰 ACL을 변경하는 경우 변경된 ACL이 TCAM에서 다시 프로그래밍되기 전에 이 메시지가 잠깐 표시됩니다.

---

1단계: `show processes cpu` 명령을 사용하여 Cisco IOS 프로세스를 확인합니다.

`show processes cpucommand`를 실행합니다. Cat4k **Mgmt** LoPriprocess가 CPU 주기의 대부분을 차지하기 때문에 CPU 사용률이 높습니다.

<#root>

Switch#

show processes cpu

CPU utilization for five seconds: 99%/0%; one minute: 99%; five minutes: 99%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	0	11	0	0.00%	0.00%	0.00%	0	Chunk Manager
2	9716	632814	15	0.00%	0.00%	0.00%	0	Load Meter
3	780	302	2582	0.00%	0.00%	0.00%	0	SpanTree Helper

!--- Output suppressed.

23	18208	3154201	5	0.00%	0.00%	0.00%	0	TTY Background
24	37208	3942818	9	0.00%	0.00%	0.00%	0	Per-Second Jobs
25	1046448	110711	9452	0.00%	0.03%	0.00%	0	Per-minute Jobs
26	175803612	339500656	517	4.12%	4.31%	4.48%	0	Cat4k Mgmt HiPri

27	835809548	339138782	2464	86.81%	89.20%	89.76%	0	Cat4k Mgmt LoPri
----	-----------	-----------	------	--------	--------	--------	---	------------------

28	28668	2058810	13	0.00%	0.00%	0.00%	0	Galios Reschedul
----	-------	---------	----	-------	-------	-------	---	------------------

2단계: 명령을 사용하여 Catalyst 4500 관련 프로세스를 show platform health 확인합니다.

명령을 show platform health 실행합니다. CPU 바인딩된 패킷을 처리하는 작업인 K2CpuMan Review에서 CPU를 사용하는 것을 볼 수 있습니다.

<#root>

Switch#

show platform health

%CPU	%CPU	RunTimeMax	Priority		Average %CPU		Total		5Sec	Min	Hour	CPU
			Target	Actual	Target	Actual	Fg	Bg				
Lj-poll		1.00	0.01	2	0	100	500	0	0	0	13:45	
GalChassisVp-review		3.00	0.20	10	16	100	500	0	0	0	88:44	
S2w-JobEventSchedule		10.00	0.57	10	7	100	500	1	0	0	404:22	
Stub-JobEventSchedule		10.00	0.00	10	0	100	500	0	0	0	0:00	
StatValueMan Update		1.00	0.09	1	0	100	500	0	0	0	91:33	
Pim-review		0.10	0.00	1	0	100	500	0	0	0	4:46	
Ebm-host-review		1.00	0.00	8	4	100	500	0	0	0	14:01	
Ebm-port-review		0.10	0.00	1	0	100	500	0	0	0	0:20	
Protocol-aging-revie		0.20	0.00	2	0	100	500	0	0	0	0:01	
Ac1-Flattener		1.00	0.00	10	5	100	500	0	0	0	0:04	
KxAc1PathMan create/		1.00	0.00	10	5	100	500	0	0	0	0:21	
KxAc1PathMan update		2.00	0.00	10	6	100	500	0	0	0	0:05	
KxAc1PathMan reprogr		1.00	0.00	2	1	100	500	0	0	0	0:00	
TagMan-InformMtegRev		1.00	0.00	5	0	100	500	0	0	0	0:00	
TagMan-RecreateMtegR		1.00	0.00	10	14	100	500	0	0	0	0:18	
<b>K2CpuMan Review</b>		<b>30.00</b>	<b>91.31</b>	<b>30</b>	<b>92</b>	<b>100</b>	<b>500</b>	<b>128</b>	<b>119</b>	<b>84</b>	<b>13039:02</b>	
K2Acce1PacketMan: Tx	10.00	2.30	20	0	100	500	2	2	2	1345:30		
K2Acce1PacketMan: Au	0.10	0.00	0	0	100	500	0	0	0	0:00		

3단계: CPU 바인딩 트래픽의 유형을 식별하려면 트래픽을 수신하는 CPU 큐를 확인합니다.

또한 어떤 CPU 대기열에 어떤 트래픽이 있는지, 그리고 어떤 유형의 트래픽이 CPU 대기열에 영향을 주는지 더욱 잘 이해해야 합니다. `show platform cpu packet statistics` 명령을 실행합니다. ACL sw 처리 대기열이 많은 패킷을 받는다는 것을 볼 수 있습니다. 따라서 TCAM 오버플로는 이러한 높은 CPU 사용률 문제의 원인입니다.

<#root>

Switch#

`show platform cpu packet statistics`

*!--- Output suppressed.*

## Packets Received by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Control	57902635	22	16	12	3
Host Learning	464678	0	0	0	0
L3 Fwd Low	623229	0	0	0	0
L2 Fwd Low	11267182	7	4	6	1
L3 Rx High	508	0	0	0	0
L3 Rx Low	1275695	10	1	0	0
ACL fwd(snooping)	2645752	0	0	0	0
ACL log, unreach	51443268	9	4	5	5
<b>ACL sw processing</b>	<b>842889240</b>	<b>1453</b>	<b>1532</b>	<b>1267</b>	<b>1179</b>

## Packets Dropped by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
L2 Fwd Low	3270	0	0	0	0
ACL sw processing	12636	0	0	0	0

### 4단계: 문제 해결

[3단계](#)에서 이 시나리오의 근본 원인을 확인했습니다. 오버플로를 일으킨 ACL을 제거하거나 오버플로를 방지하기 위해 ACL을 최소화합니다. 또한 하드웨어에서 [ACL 컨피그레이션](#) 및 프로그래밍을 최적화하려면 ACL을 사용하여 네트워크 보안 구성 지침을 검토하십시오.

### ACL의 로그 키워드

Catalyst 4500은 특정 ACL 항목에 영향을 주는 패킷 세부 사항 로깅을 지원하지만 과도한 로깅은 CPU 사용률을 높일 수 있습니다. 트래픽 검색 단계를 제외하고 logkeywords를 사용하지 마십시오. 트래픽 검색 단계에서는 명시적으로 ACE를 구성하지 않은 네트워크를 통과하는 트래픽을 식별합니다. 통계를 수집하기 위해 logkeyword를 사용하지 마십시오. Cisco IOS Software 릴리스 12.1(13)EW 이상에서는 로그 메시지가 속도가 제한됩니다. ACL과 일치하는 패킷 수를 카운트하기 위해 메시지를 사용할 경우 카운트가 정확하지 않습니다. 대신 정확한 통계를 `show access-list` 위해 명령을 사용합니다. 컨피그레이션 또는 로그 메시지를 검토하면 ACL 로깅 기능의 사용을 나타낼 수 있으므로 이러한 근본 원인을 더 쉽게 식별할 수 있습니다.

1단계: `show processes cpu` 명령을 사용하여 Cisco IOS 프로세스를 확인합니다.

어떤 Cisco `show processes cpu` 프로세스가 CPU를 사용하는지 확인하려면 이를 실행합니다. 이 명령 출력에서는 상위 프로세스가 `Cat4k Mgmt LoPri`임을 알 수 있습니다.

<#root>

Switch#

show processes cpu

CPU utilization for five seconds: 99%/0%; one minute: 99%; five minutes: 99%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	0	11	0	0.00%	0.00%	0.00%	0	Chunk Manager
2	9716	632814	15	0.00%	0.00%	0.00%	0	Load Meter

!--- Output suppressed.

26	175803612	339500656	517	4.12%	4.31%	4.48%	0	Cat4k Mgmt HiPri
----	-----------	-----------	-----	-------	-------	-------	---	------------------

27	835809548	339138782	2464	86.81%	89.20%	89.76%	0	Cat4k Mgmt LoPri
----	-----------	-----------	------	--------	--------	--------	---	------------------

28	28668	2058810	13	0.00%	0.00%	0.00%	0	Galios Reschedul
----	-------	---------	----	-------	-------	-------	---	------------------

2단계: show platform health 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

CPU를 사용하는 플랫폼별 프로세스를 확인합니다. 명령을 show platform health 실행합니다. 출력에서 **K2CpuMan Reviewprocess**는 CPU 주기의 대부분을 사용합니다. 이 활동은 CPU가 대상 패킷을 처리할 때 사용 중임을 나타냅니다.

<#root>

Switch#

show platform health

%CPU Target	%CPU Actual	RunTimeMax Target	RunTimeMax Actual	Priority Fg	Priority Bg	Average 5Sec	%CPU Min	%CPU Hour	Total CPU
-------------	-------------	-------------------	-------------------	-------------	-------------	--------------	----------	-----------	-----------

Lj-poll	1.00	0.01	2	0	100	500	0	0	0	13:45
GalChassisVp-review	3.00	0.20	10	16	100	500	0	0	0	88:44
S2w-JobEventSchedule	10.00	0.57	10	7	100	500	1	0	0	404:22
Stub-JobEventSchedule	10.00	0.00	10	0	100	500	0	0	0	0:00
StatValueMan Update	1.00	0.09	1	0	100	500	0	0	0	91:33
Pim-review	0.10	0.00	1	0	100	500	0	0	0	4:46
Ebm-host-review	1.00	0.00	8	4	100	500	0	0	0	14:01
Ebm-port-review	0.10	0.00	1	0	100	500	0	0	0	0:20
Protocol-aging-revie	0.20	0.00	2	0	100	500	0	0	0	0:01
Ac1-Flattener	1.00	0.00	10	5	100	500	0	0	0	0:04
KxAc1PathMan create/	1.00	0.00	10	5	100	500	0	0	0	0:21
KxAc1PathMan update	2.00	0.00	10	6	100	500	0	0	0	0:05
KxAc1PathMan reprog	1.00	0.00	2	1	100	500	0	0	0	0:00
TagMan-InformMtegRev	1.00	0.00	5	0	100	500	0	0	0	0:00
TagMan-RecreateMtegR	1.00	0.00	10	14	100	500	0	0	0	0:18
<b>K2CpuMan Review</b>	<b>30.00</b>	<b>91.31</b>	<b>30</b>	<b>92</b>	<b>100</b>	<b>500</b>	<b>128</b>	<b>119</b>	<b>84</b>	<b>13039:02</b>
K2Acce1PacketMan: Tx	10.00	2.30	20	0	100	500	2	2	2	1345:30
K2Acce1PacketMan: Au	0.10	0.00	0	0	100	500	0	0	0	0:00

3단계: CPU 바인딩 트래픽의 유형을 식별하려면 트래픽을 수신하는 CPU 큐를 확인합니다.

CPU에 도달하는 트래픽의 유형을 확인하려면 명령을 **show platform cpu packet statistics** 실행합니다. 이 명령 출력에서 패킷 수신이 ACLlogkeyword로 인한 것임을 알 수 있습니다.

<#root>

Switch#

**show platform cpu packet statistics**

*!--- Output suppressed.*

Total packet queues 16  
Packets Received by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
-------	-------	-----------	-----------	-----------	------------

Control	1198701435	35	35	34	35
Host Learning	874391	0	0	0	0
L3 Fwd High	428	0	0	0	0
L3 Fwd Medium	12745	0	0	0	0
L3 Fwd Low	2420401	0	0	0	0
L2 Fwd High	26855	0	0	0	0
L2 Fwd Medium	116587	0	0	0	0
L2 Fwd Low	317829151	53	41	31	31
L3 Rx High	2371	0	0	0	0
L3 Rx Low	32333361	7	1	2	0
RPF Failure	4127	0	0	0	0
ACL fwd (snooping)	107743299	4	4	4	4

ACL log, unreachable 1209056404 1987 2125 2139 2089

#### Packets Dropped by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
ACL log, unreachable	193094788	509	362	437	394

#### 4단계: 문제 해결

3단계에서 이 시나리오의 근본 원인을 확인했습니다. 이 문제를 방지하려면 ACL에서 logkeyword를 제거하십시오. Cisco IOS 소프트웨어 릴리스 12.1(13)EW1 이상에서는 CPU 사용률이 너무 높아지지 않도록 패킷의 속도가 제한됩니다. ACL 적중 수를 추적하는 방법으로 액세스 목록 카운터를 사용합니다. 명령 출력에서 액세스 목록 카운터를 볼 수 `show access-list acl_id` 있습니다.

#### 레이어 2 전달 루프

레이어 2 전달 루프는 STP(Spanning Tree Protocol)의 잘못된 구현 및 STP에 영향을 미칠 수 있는 다양한 문제로 인해 발생할 수 있습니다.

1단계: 명령을 사용하여 Cisco IOS 프로세스를 `show processes cpu` 확인합니다

이 섹션에서는 높은 CPU 사용률 문제를 줄이기 위해 관리자가 사용하는 명령어를 검토합니다. 이 명령을 `show processes cpu` 실행하면 두 가지 기본 프로세스인 `Cat4k Mgmt LoPriandSpanning 트리`에서 주로 CPU를 사용하는 것을 볼 수 있습니다. 이 정보만 사용하면 스페닝 트리 프로세스에서 CPU 사이클의 상당한 부분을 사용하게 됩니다.

Switch#

show processes cpu

CPU utilization for five seconds: 74%/1%; one minute: 73%; five minutes: 50%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	4	198	20	0.00%	0.00%	0.00%	0	Chunk Manager
2	4	290	13	0.00%	0.00%	0.00%	0	Load Meter

!--- Output suppressed.

25	488	33	14787	0.00%	0.02%	0.00%	0	Per-minute Jobs
26	90656	223674	405	6.79%	6.90%	7.22%	0	Cat4k Mgmt HiPri

27	158796	59219	2681	32.55%	33.80%	21.43%	0	Cat4k Mgmt LoPri
----	--------	-------	------	--------	--------	--------	---	------------------

28	20	1693	11	0.00%	0.00%	0.00%	0	Galios Reschedul
29	0	1	0	0.00%	0.00%	0.00%	0	IOS ACL Helper
30	0	2	0	0.00%	0.00%	0.00%	0	NAM Manager

!--- Output suppressed.

41	0	1	0	0.00%	0.00%	0.00%	0	SFF8472
42	0	2	0	0.00%	0.00%	0.00%	0	AAA Dictionary R

43	78564	20723	3791	32.63%	30.03%	17.35%	0	Spanning Tree
----	-------	-------	------	--------	--------	--------	---	---------------

44	112	999	112	0.00%	0.00%	0.00%	0	DTP Protocol
45	0	147	0	0.00%	0.00%	0.00%	0	Ethchnl

2단계: show platform health 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

어떤 플랫폼별 프로세스가 CPU를 사용하는지 파악하려면 명령을 show platform health 실행합니다. 이 출력에서 CPU 바인딩된 패킷을

처리하는 작업인 K2CpuMan Reviewprocess가 CPU를 사용하는 것을 볼 수 있습니다.

```
<#root>
```

```
Switch#
```

```
show platform health
```

```
%CPU   %CPU   RunTimeMax  Priority  Average %CPU  Total
          Target Actual Target Actual   Fg   Bg 5Sec Min Hour  CPU
!--- Output suppressed.

TagMan-RecreateMtegR   1.00   0.00    10     0  100  500    0   0   0  0:00

K2CpuMan Review       30.00  37.62    30    53  100  500   41  33   1  2:12

K2Acce1PacketMan: Tx  10.00   4.95    20     0  100  500    5   4   0  0:36
K2Acce1PacketMan: Au   0.10   0.00     0     0  100  500    0   0   0  0:00
K2Ac1Man-taggedFlatA  1.00   0.00    10     0  100  500    0   0   0  0:00
```

3단계: CPU 바인딩 트래픽의 유형을 식별하기 위해 트래픽을 수신하는 CPU 큐를 확인합니다.

어떤 CPU show platform cpu packet statistics 대기열이 CPU 바인딩 패킷을 수신하는지 확인하기 위해 명령을 실행합니다. 이 섹션의 출력은 제어 대기열이 많은 패킷을 수신하는 것을 보여줍니다. 표 1의 [정보](#)와 [1단계](#)에서 도출한 결론을 사용하십시오. CPU가 처리하는 패킷과 CPU 사용률이 높은 이유가 BPDU 처리 때문인지 확인할 수 있습니다.

```
<#root>
```

```
Switch#
```

show platform cpu packet statistics

!--- Output suppressed.

Total packet queues 16  
Packets Received by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Esmp	202760	196	173	128	28
Control	388623	2121	1740	598	16

Packets Dropped by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Control	17918	0	19	24	3

4단계: 근본 원인 파악 및 문제 해결

일반적으로 문제 해결을 위해 다음 단계를 완료할 수 있습니다(상황에 따라 일부 단계는 필요하지 않음).

- 루프를 식별합니다.

- 루프의 범위를 확인합니다.

- 루프를 해제합니다.

-

루프의 원인을 수정합니다.

•

이중화를 복구합니다.

각 단계는 Troubleshooting Forwarding Loops - Troubleshooting [STP on Catalyst Switches Running Cisco IOS System Software](#)에서 자세히 설명합니다.

5단계: 고급 STP 기능 구현

•

**BDPU 가드 - Portfast 지원 포트에 연결된 무단 네트워크 디바이스로부터 STP를 보호합니다.** 자세한 내용은 [스패닝 트리 포트Fast BPDU 가드](#) 개선 사항을 참조하십시오.

•

**루프 가드 - 레이어 2 네트워크의 안정성을 높입니다.** 자세한 내용은 [Loop Guard 및 BPDU Skew Detection 기능을 사용한 스패닝 트리 프로토콜](#) 개선 사항을 참조하십시오.

•

**루트 가드 - 네트워크에 루트 브리지를 배치합니다.** 자세한 내용은 [스패닝 트리 프로토콜 루트 가드](#) 개선 사항을 참조하십시오.

•

**UDLD - 단방향 링크를 탐지하고 전달 루프를 방지합니다.** 자세한 내용은 [Unidirectional Link Detection Protocol 기능 이해 및 구성](#)을 참조하십시오.

CPU 사용률이 높은 기타 원인

다음은 높은 CPU 사용률의 기타 알려진 원인입니다.

•

[과도한 링크 플랩](#)

•

## FIB 일관성 검사로 인한 CPU 사용률 급증

- 

### K2FibAdjMan 호스트 이동 프로세스의 높은 CPU 사용률

- 

### RkiosPortMan 포트 검토 프로세스의 높은 CPU 사용률

- 

### 트렁크 포트를 사용하여 IP 전화기에 연결된 경우 높은 CPU 사용률

- 

### RSPAN 및 레이어 3 제어 패킷을 통한 높은 CPU 사용률

- 

### 대규모 ACL 프로그래밍 중 급증

CPU 사용률 급증은 인터페이스에서 대규모 ACL을 적용하거나 제거하는 동안 발생합니다.

### 과도한 링크 플랩

하나 이상의 연결된 링크가 과도하게 플랩되기 시작하면 Catalyst 4500은 높은 CPU 사용률을 나타냅니다. 이러한 상황은 Cisco IOS 소프트웨어 릴리스 12.2(20)EWA 이전의 Cisco IOS 소프트웨어 릴리스에서 발생합니다.

1단계: show processes cpu 명령을 사용하여 Cisco IOS 프로세스를 확인합니다.

show **processes** cpucommand를 실행하여 어떤 Cisco IOS 프로세스가 CPU를 사용하는지 확인합니다. 이 명령 출력에서 상위 프로세스는 **Cat4k Mgmt LoPri**입니다.

```
<#root>
```

```
Switch#
```

show processes cpu

CPU utilization for five seconds: 96%/0%; one minute: 76%; five minutes: 68%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	0	4	0	0.00%	0.00%	0.00%	0	Chunk Manager
2	9840	463370	21	0.00%	0.00%	0.00%	0	Load Meter
3	0	2	0	0.00%	0.00%	0.00%	0	SNMP Timers

!--- Output suppressed.

27	232385144	530644966	437	13.98%	12.65%	12.16%	0	Cat4k Mgmt HiPri
----	-----------	-----------	-----	--------	--------	--------	---	------------------

28	564756724	156627753	3605	64.74%	60.71%	54.75%	0	Cat4k Mgmt LoPri
----	-----------	-----------	------	--------	--------	--------	---	------------------

29	9716	1806301	5	0.00%	0.00%	0.00%	0	Galios Reschedul
----	------	---------	---	-------	-------	-------	---	------------------

2단계: show platform health 명령을 사용하여 Catalyst 4500 관련 프로세스를 확인합니다.

thesaw platform healthcommand의 출력은 KxAclPathMan createprocess가 CPU를 사용함을 나타냅니다. 이 프로세스는 내부 경로 생성을 위한 것입니다.

<#root>

Switch#

show platform health

	%CPU		RunTimeMax		Priority		Average %CPU			Total CPU
	Target	Actual	Target	Actual	Fg	Bg	5Sec	Min	Hour	
Lj-poll	1.00	0.03	2	0	100	500	0	0	0	9:49
GalChassisVp-review	3.00	1.11	10	62	100	500	0	0	0	37:39
S2w-JobEventSchedule	10.00	2.85	10	8	100	500	2	2	2	90:00
Stub-JobEventSchedule	10.00	5.27	10	9	100	500	4	4	4	186:2
Pim-review	0.10	0.00	1	0	100	500	0	0	0	2:51
Ebm-host-review	1.00	0.00	8	4	100	500	0	0	0	8:06
Ebm-port-review	0.10	0.00	1	0	100	500	0	0	0	0:14

Protocol-aging-revie	0.20	0.00	2	0	100	500	0	0	0	0:00
Ac1-Flattener	1.00	0.00	10	5	100	500	0	0	0	0:00
<b>KxAclPathMan create/</b>	<b>1.00</b>	<b>69.11</b>	<b>10</b>	<b>5</b>	<b>100</b>	<b>500</b>	<b>42</b>	<b>53</b>	<b>22</b>	<b>715:0</b>
KxAclPathMan update	2.00	0.76	10	6	100	500	0	0	0	86:00
KxAclPathMan reprogr	1.00	0.00	2	1	100	500	0	0	0	0:00
TagMan-InformMtegRev	1.00	0.00	5	0	100	500	0	0	0	0:00
TagMan-RecreateMtegR	1.00	0.00	10	227	100	500	0	0	0	0:00
K2CpuMan Review	30.00	8.05	30	57	100	500	6	5	5	215:0
K2Acce1PacketMan: Tx	10.00	6.86	20	0	100	500	5	5	4	78:42

3단계: 근본 원인 파악

링크 가동/중단 메시지에 대한 로깅을 활성화합니다. 이 로깅은 기본적으로 활성화되어 있지 않습니다. 활성화하면 문제가 되는 링크를 매우 빠르게 좁힐 수 있습니다. 모든 인터페이스에서 **logging event link-statuscommand** 명령을 실행합니다. 다음 예와 같이 인터페이스 범위에서 편리하게 활성화하려면 **interface rangecommand**를 사용할 수 있습니다.

```
<#root>
```

```
Switch#
```

```
configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#
```

```
interface range gigabitethernet 5/1 - 48
```

```
Switch(config-if-range)#
```

```
logging event link-status
```

```
Switch(config--if-range)#
```

```
end
```

```
<#root>
```

```
Switch#
```

```
show logging
```

```
!--- Output suppressed.
```

```
3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to down  
3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to up  
3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to down  
3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to up  
3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to down  
3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to up
```

결함 또는 플래핑 인터페이스를 식별한 후에는 높은 CPU 사용률 문제를 해결하기 위해 인터페이스를 종료합니다. Cisco IOS 소프트웨어 릴리스 12.2(20)EWA 이상에서는 이 flapping-links 조건에 대한 Catalyst 4500 동작이 개선되었습니다. 따라서 CPU에 미치는 영향은 개선 이전만큼 크지 않습니다. 이 프로세스는 백그라운드 프로세스입니다. 이 문제로 인해 CPU 사용률이 높더라도 Catalyst 4500 스위치에 부정적인 영향을 미치지 않습니다.

FIB 일관성 검사로 인한 CPU 사용률 급증

Catalyst 4500에서는 FIB 테이블 일관성 검사 중 CPU 사용률이 일시적으로 급증할 수 있습니다. FIB 테이블은 CEF 프로세스에서 생성하는 L3 전달 테이블입니다. 일관성 검사는 Cisco IOS 소프트웨어 FIB 테이블과 하드웨어 항목 간의 일관성을 유지합니다. 이러한 일

관성으로 인해 패킷이 잘못 라우팅되지 않습니다. 확인은 2초마다 수행되며 우선순위가 낮은 백그라운드 프로세스로 실행됩니다. 이 프로세스는 정상적인 동작이며 우선순위가 높은 다른 프로세스 또는 패킷을 방해하지 않습니다.

thesaw platform healthcommand의 출력은 K2Fib Consistency가 CPU의 대부분을 사용한다는 것을 보여줍니다.

참고: 이 프로세스에 대한 평균 CPU 사용률이 1분 또는 1시간 동안 미미하며, 이는 이 검사가 간단한 정기 검토임을 확인합니다. 이 백그라운드 프로세스는 유휴 CPU 사이클만 사용합니다.

<#root>

Switch#

show platform health

	%CPU Target	%CPU Actual	RunTimeMax Target	RunTimeMax Actual	Priority Fg	Priority Bg	Average 5Sec	%CPU Min	%CPU Hour	Total CPU
Lj-poll	1.00	0.02	2	1	100	500	0	0	0	1:09
GalChassisVp-review	3.00	0.29	10	3	100	500	0	0	0	11:15

!--- Output suppressed.

K2Fib cam usage revi	2.00	0.00	15	0	100	500	0	0	0	0:00
K2Fib IrmFib Review	2.00	0.00	15	0	100	500	0	0	0	0:00
K2Fib Vrf Default Ro	2.00	0.00	15	0	100	500	0	0	0	0:00
K2Fib AdjRepop Revie	2.00	0.00	15	0	100	500	0	0	0	0:00
K2Fib Vrf Unpunt Rev	2.00	0.01	15	0	100	500	0	0	0	0:23
K2Fib Consistency Ch	1.00	60.40	5	2	100	500				

0 0 0

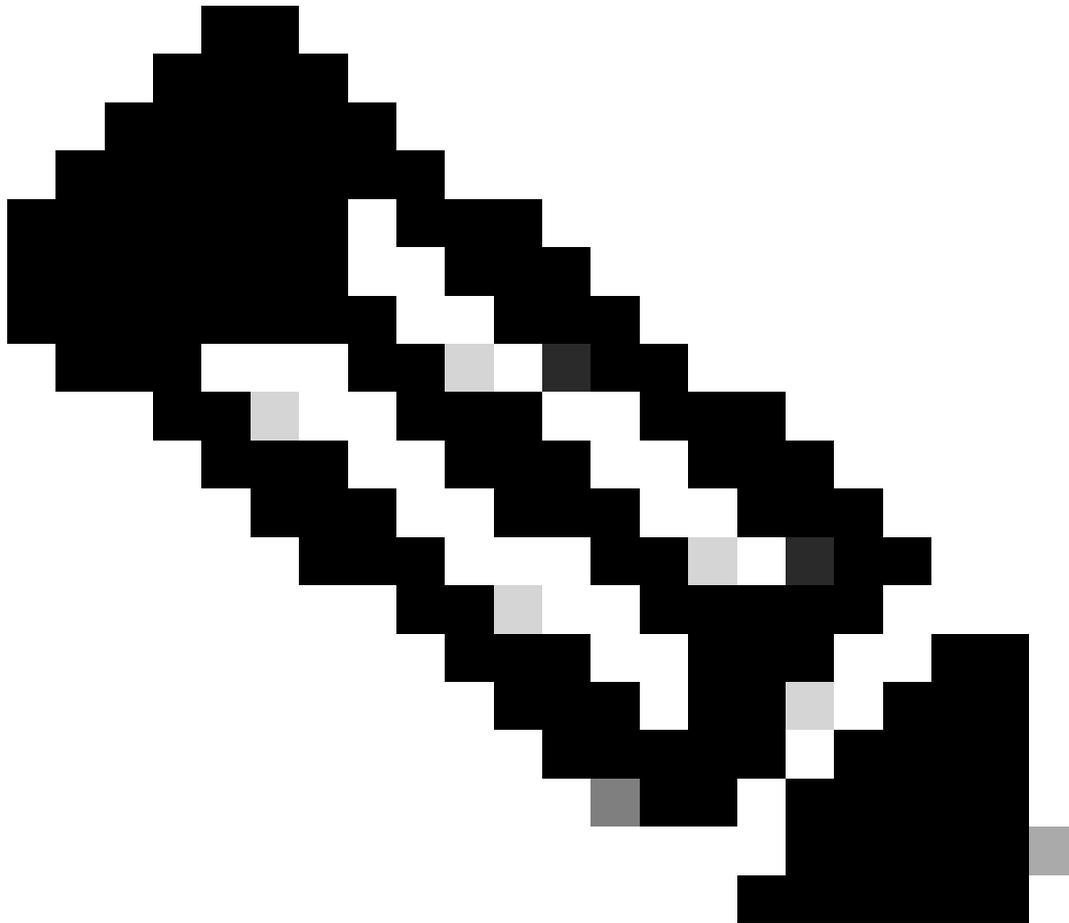
100:23

K2FibAdjMan Stats Re	2.00	0.30	10	4	100	500	0	0	0	6:21
K2FibAdjMan Host Mov	2.00	0.00	10	4	100	500	0	0	0	0:00
K2FibAdjMan Adj Chan	2.00	0.00	10	0	100	500	0	0	0	0:00
K2FibMulticast Signa	2.00	0.01	10	2	100	500	0	0	0	2:04

K2FibAdjMan Host Move 프로세스에서 높은 CPU 사용률

Catalyst 4500은 K2FibAdjMan 호스트 이동 프로세스에서 높은 CPU 사용률을 표시할 수 있습니다. 이러한 높은 사용률은 `show platform healthcommand`의 출력에 나타납니다. 많은 MAC 주소가 자주 만료되거나 새 포트에서 학습되므로 CPU 사용률이 높아집니다. `mac-address-table aging-time`의 기본값은 5분 또는 300초입니다. 이 문제를 해결하려면 MAC 주소 에이징 시간을 늘리거나, MAC 주소 이동이 많이 발생하지 않도록 네트워크를 엔지니어링할 수 있습니다. Cisco IOS 소프트웨어 릴리스 12.2(18)EW 이상에서는 더 적은 CPU를 소비하기 위해 이 프로세스 동작을 개선했습니다. Cisco 버그 IDCSCed15021을 [참조하십시오.](#)

---



참고: 등록된 Cisco 사용자만 내부 Cisco 툴 및 정보에 액세스할 수 있습니다.

---

Switch#

show platform health

	%CPU Target	%CPU Actual	RunTimeMax Target	RunTimeMax Actual	Priority Fg	Priority Bg	Average 5Sec	%CPU Min	%CPU Hour	Total CPU
Lj-poll	1.00	0.02	2	1	100	500	0	0	0	1:09
GalChassisVp-review	3.00	0.29	10	3	100	500	0	0	0	11:15
S2w-JobEventSchedule	10.00	0.32	10	7	100	500	0	0	0	10:14

!--- Output suppressed.

K2FibAdjMan Stats Re	2.00	0.30	10	4	100	500	0	0	0	6:21
----------------------	------	------	----	---	-----	-----	---	---	---	------

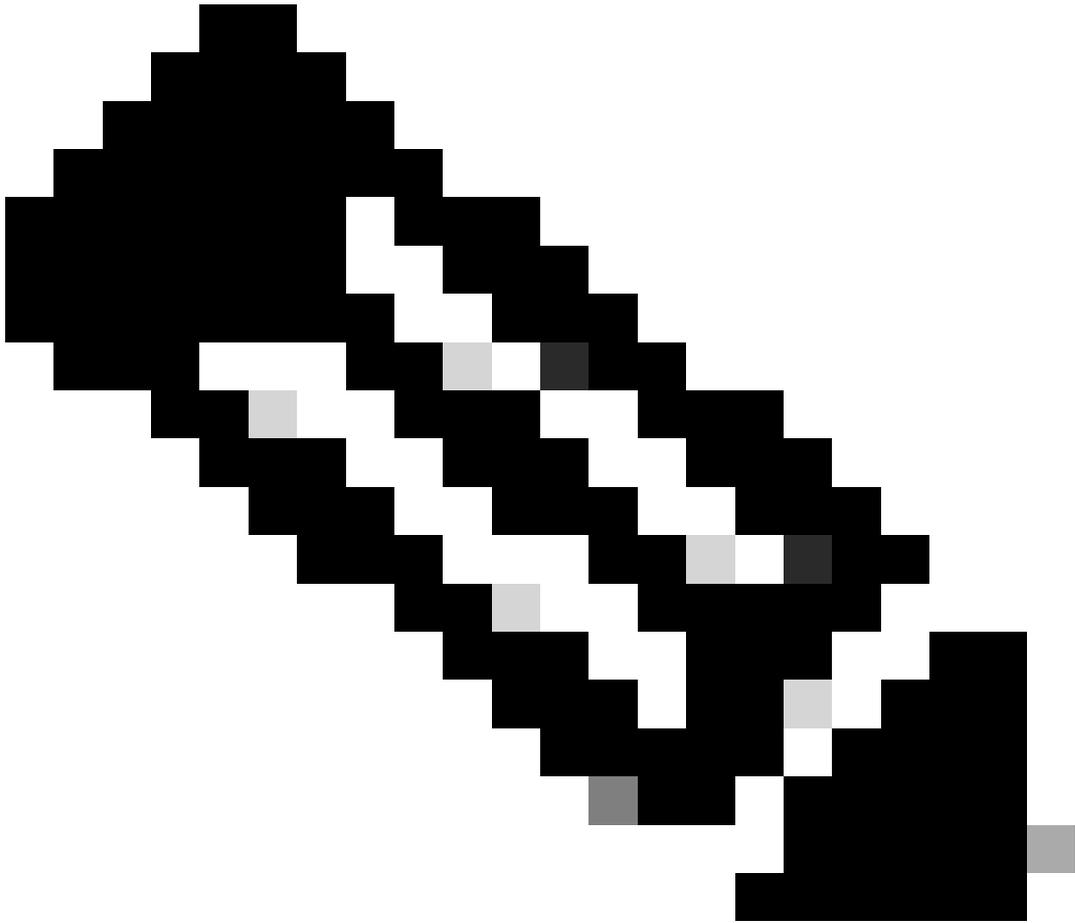
K2FibAdjMan Host Mov	2.00	18.68	10	4	100	500	25	29	28	2134:39
----------------------	------	-------	----	---	-----	-----	----	----	----	---------

K2FibAdjMan Adj Chan	2.00	0.00	10	0	100	500	0	0	0	0:00
K2FibMulticast Signa	2.00	0.01	10	2	100	500	0	0	0	2:04
K2FibMulticast Entry	2.00	0.00	10	7	100	500	0	0	0	0:00

전역 컨피그레이션 모드에서 MAC 주소의 최대 에이징 시간을 수정할 수 있습니다. 명령 구문은 라우터의 경우 **mac-address-table 에이징-시간 초**와 Catalyst 스위치의 경우 **mac-address-table 에이징-시간 초[vlan-id]**입니다. 자세한 내용은 [Cisco IOS Switching Services 명령 참조 설명서를 참조하십시오.](#)

RkiosPortMan Port Review 프로세스에서 높은 CPU 사용률

Catalyst 4500은 Cisco IOS Software 릴리스 12.2(25)EWA 및 12.2(25)EWA1의 **how platform healthcommand** 출력에서 RkiosPortMan Port Reviewprocess의 높은 CPU 사용률을 표시할 수 있습니다. Cisco 버그 [IDCSCeh08768](#)은 Cisco IOS Software 릴리스 12.2(25)EWA2에서 확인되는 높은 사용률을 유발합니다. 이 프로세스는 백그라운드 프로세스이며 Catalyst 4500 스위치의 안정성에 영향을 주지 않습니다.



참고: 등록된 Cisco 사용자만 내부 Cisco 툴 및 정보에 액세스할 수 있습니다.

---

<#root>

Switch#

show platform health

	%CPU	%CPU	RunTimeMax		Priority		Average %CPU			Total
	Target	Actual	Target	Actual	Fg	Bg	5Sec	Min	Hour	CPU
Lj-poll	1.00	0.02	2	1	100	500	0	0	0	1:09
GalChassisVp-review	3.00	0.29	10	3	100	500	0	0	0	11:15
S2w-JobEventSchedule	10.00	0.32	10	7	100	500	0	0	0	10:14

!--- Output suppressed.

K2 Packet Memory Dia	2.00	0.00	15	8	100	500	0	1	1	45:46
K2 L2 Aging Table Re	2.00	0.12	20	3	100	500	0	0	0	7:22

RkiosPortMan Port Re 2.00 87.92 12 7 100 500 99 99 89 1052:36

Rkios Module State R 4.00 0.02 40 1 100 500 0 0 0 1:28  
Rkios Online Diag Re 4.00 0.02 40 0 100 500 0 0 0 1:15

트렁크 포트를 사용하여 IP 전화기에 연결된 경우 높은 CPU 사용률

포트가 음성 VLAN 옵션 및 액세스 VLAN 옵션 모두에 대해 구성된 경우 포트는 다중 VLAN 액세스 포트로서 작동합니다. 장점은 음성 및 액세스 VLAN 옵션에 대해 구성된 이러한 VLAN만 트렁킹된다는 것입니다.

전화기로 트렁크되는 VLAN은 STP 인스턴스 수를 늘립니다. 스위치는 STP 인스턴스를 관리합니다. STP 인스턴스의 증가를 관리하면 STP CPU 사용률도 증가합니다.

또한 모든 VLAN을 트렁킹하면 불필요한 브로드캐스트, 멀티캐스트 및 알 수 없는 유니캐스트 트래픽을 유발하여 전화기 링크에 영향을 줍니다.

<#root>

Switch#

show processes cpu

CPU utilization for five seconds: 69%/0%; one minute: 72%; five minutes: 73%

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	4	165	24	0.00%	0.00%	0.00%	0	Chunk Manager
2	29012	739091	39	0.00%	0.00%	0.00%	0	Load Meter
3	67080	13762	4874	0.00%	0.00%	0.00%	0	SpanTree Helper
4	0	1	0	0.00%	0.00%	0.00%	0	Deferred Events
5	0	2	0	0.00%	0.00%	0.00%	0	IpSecMibTopN
6	4980144	570766	8725	0.00%	0.09%	0.11%	0	Check heaps
26	539173952	530982442	1015	13.09%	13.05%	13.20%	0	Cat4k Mgmt HiPri
27	716335120	180543127	3967	17.61%	18.19%	18.41%	0	Cat4k Mgmt LoPri
33	1073728	61623	17424	0.00%	0.03%	0.00%	0	Per-minute Jobs
34	1366717824	231584970	5901	38.99%	38.90%	38.92%	0	Spanning Tree
35	2218424	18349158	120	0.00%	0.03%	0.02%	0	DTP Protocol
36	5160	369525	13	0.00%	0.00%	0.00%	0	Ethchnl
37	271016	2308022	117	0.00%	0.00%	0.00%	0	VLAN Manager
38	958084	3965585	241	0.00%	0.01%	0.01%	0	UDLD
39	1436	51011	28	0.00%	0.00%	0.00%	0	DHCP Snooping
40	780	61658	12	0.00%	0.00%	0.00%	0	Port-Security
41	1355308	12210934	110	0.00%	0.01%	0.00%	0	IP Input

RSPAN 및 레이어 3 제어 패킷을 통한 높은 CPU 사용률

RSPAN으로 캡처된 레이어 3 제어 패킷은 RSPAN 대상 인터페이스가 아닌 CPU로 전송되므로 높은 CPU를 유발합니다. L3 제어 패킷은 CPU로 전달 작업이 있는 정적 CAM 항목에 의해 캡처됩니다. 정적 CAM 항목은 모든 VLAN에 대해 글로벌합니다. 불필요한 CPU 플래딩을 방지하려면 Cisco IOS 소프트웨어 릴리스 12.2(37)SG 이상에서 제공되는 VLAN별 제어 트래픽 인터셉트 기능을 사용하십시오.

<#root>

Switch(config)#

```
access-list hardware capture mode vlan
```

정적 ACL은 입력 기능 TCAM의 상단에 설치되어 224.0.0.\* 범위에서 잘 알려진 IP 멀티캐스트 주소로 향하는 제어 패킷을 캡처합니다.

다. 정적 ACL은 부팅 시 설치되며 사용자가 구성한 ACL 앞에 표시됩니다. 정적 ACL은 항상 먼저 실행되고 모든 VLAN에서 CPU에 대한 제어 트래픽을 차단할 수 있습니다.

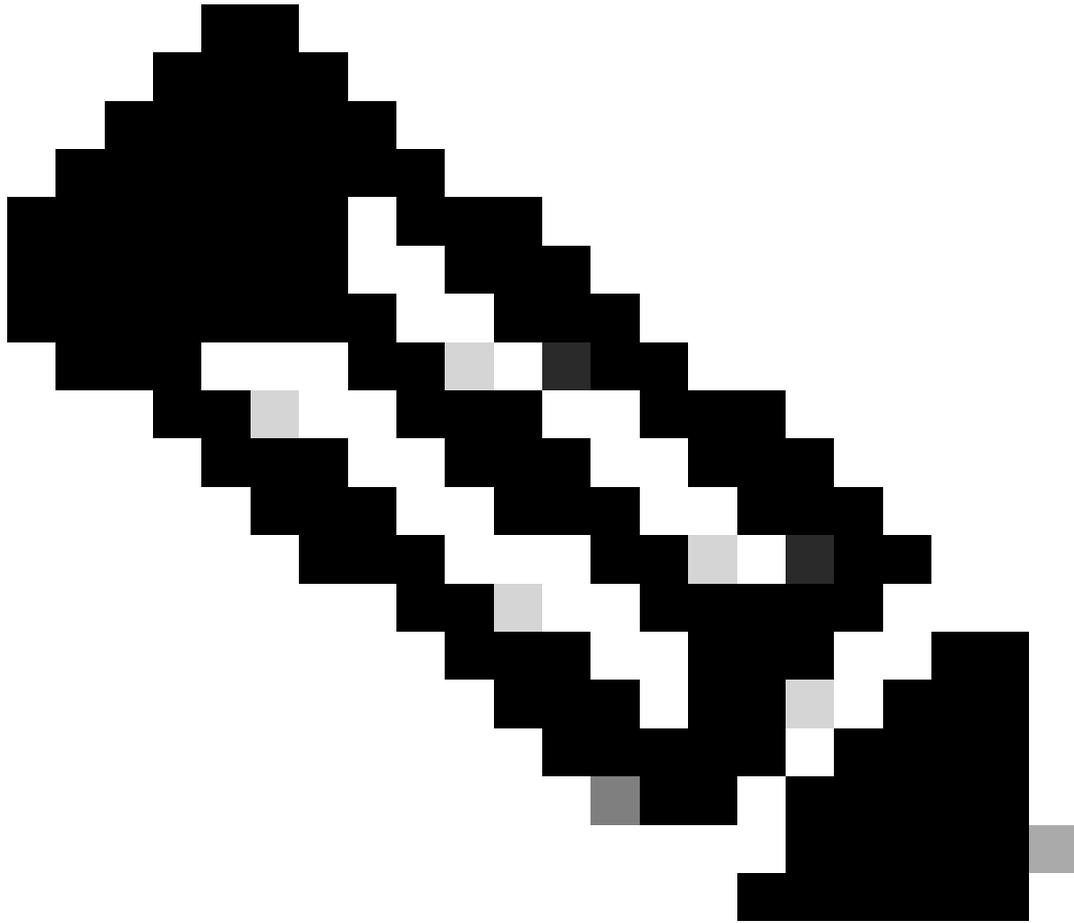
VLAN별 제어 트래픽 인터셉트 기능은 제어 트래픽을 캡처하는 선택적 VLAN별 경로 관리 모드를 제공합니다. 입력 기능 TCAM의 해당 정적 CAM 항목은 새 모드에서 무효화됩니다. 제어 패킷은 스누핑 또는 라우팅 기능이 활성화된 VLAN에 연결된 기능별 ACL에 의해 캡처됩니다. RSPAN VLAN에 연결된 기능별 ACL이 없습니다. 따라서 RSPAN VLAN에서 수신된 모든 레이어 3 제어 패킷은 CPU로 전달되지 않습니다.

#### CPU로 향하는 트래픽을 분석하기 위한 문제 해결 도구

이 문서에서 알 수 있듯이, CPU로 향하는 트래픽이 Catalyst 4500의 CPU 사용률이 높은 주요 원인 중 하나입니다. CPU 대상 트래픽은 컨피그레이션으로 인해 의도적인 것일 수도 있고, 컨피그레이션 오류 또는 서비스 거부 공격으로 인해 의도하지 않은 것일 수도 있습니다. CPU에는 이러한 트래픽으로 인한 부정적인 네트워크 영향을 방지하기 위한 QoS 메커니즘이 내장되어 있습니다. 그러나 CPU 바인딩 트래픽의 근본 원인을 파악하고 원하지 않는 경우 트래픽을 제거합니다.

#### 툴 1: SPAN으로 CPU 트래픽 모니터링—Cisco IOS Software 릴리스 12.1(19)EW 이상

Catalyst 4500에서는 표준 SPAN 기능을 사용하여 인바운드 또는 이그레스 CPU 바인딩 트래픽을 모니터링할 수 있습니다. 대상 인터페이스는 패킷 스니퍼 소프트웨어를 실행하는 패킷 모니터 또는 관리자 노트북 컴퓨터에 연결됩니다. 이 툴은 CPU가 처리하는 트래픽을 빠르고 정확하게 분석하는 데 도움이 됩니다. 이 툴은 CPU 패킷 엔진에 바인딩된 개별 대기열을 모니터링하는 기능을 제공합니다.



참고: 스위칭 엔진은 CPU 트래픽에 대해 32개의 대기열을 가지며, CPU 패킷 엔진은 16개의 대기열을 가집니다.

---

<#root>

Switch(config)#

monitor session 1 source cpu ?

```
both Monitor received and transmitted traffic
queue SPAN source CPU queue
rx Monitor received traffic only
tx Monitor transmitted traffic only
<cr>
Switch(config)#
```

```
monitor session 1 source cpu queue ?
```

```
<1-32> SPAN source CPU queue numbers
acl Input and output ACL [13-20]
adj-same-if Packets routed to the incoming interface [7]
all All queues [1-32]
bridged L2/bridged packets [29-32]
control-packet Layer 2 Control Packets [5]
mtu-exceeded Output interface MTU exceeded [9]
nfl Packets sent to CPU by netflow (unused) [8]
routed L3/routed packets [21-28]
rpf-failure Multicast RPF Failures [6]
span SPAN to CPU (unused) [11]
unknown-sa Packets with missing source address [10]
Switch(config)#
```

```
monitor session 1 source cpu queue all rx
```

```
Switch(config)#
```

```
monitor session 1 destination interface gigabitethernet 1/3
```

```
Switch(config)#
```

```
end
```

```
4w6d: %SYS-5-CONFIG_I: Configured from console by console
```

Switch#

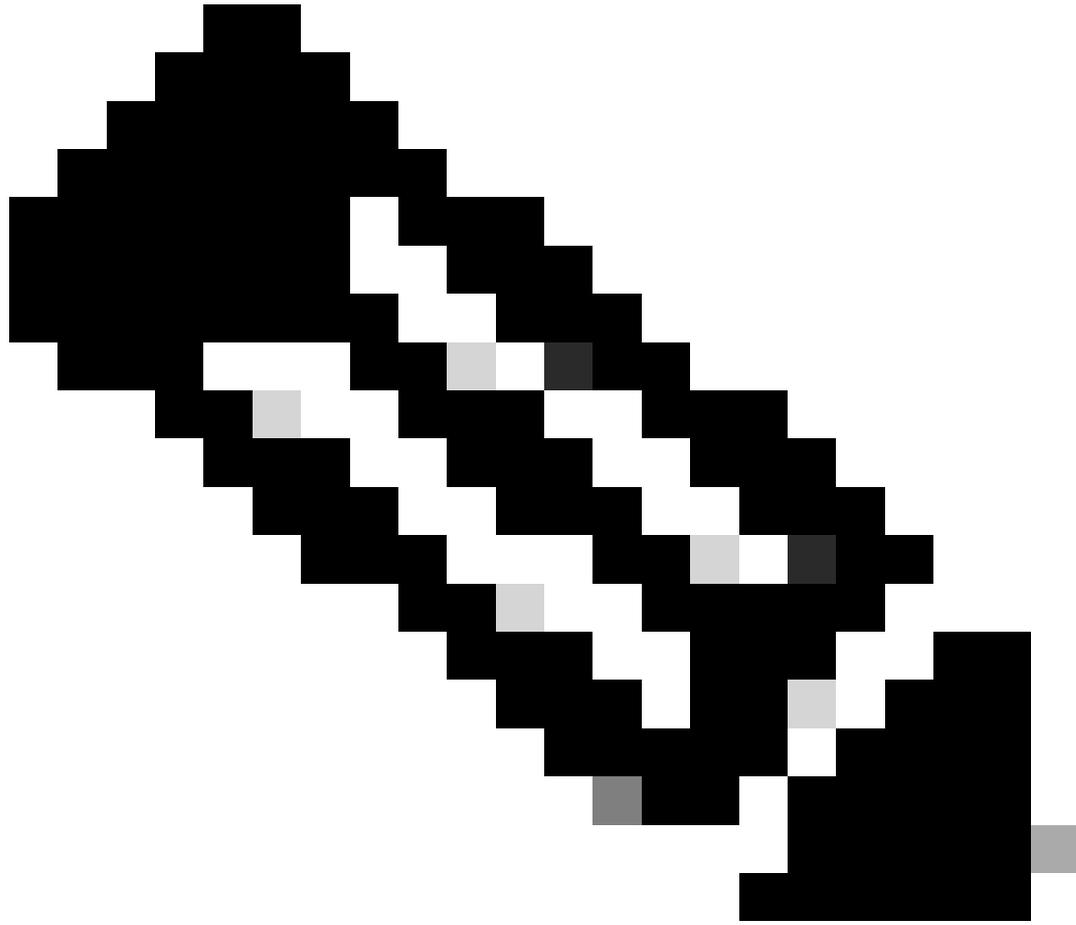
```
show monitor session 1
```

Session 1

-----

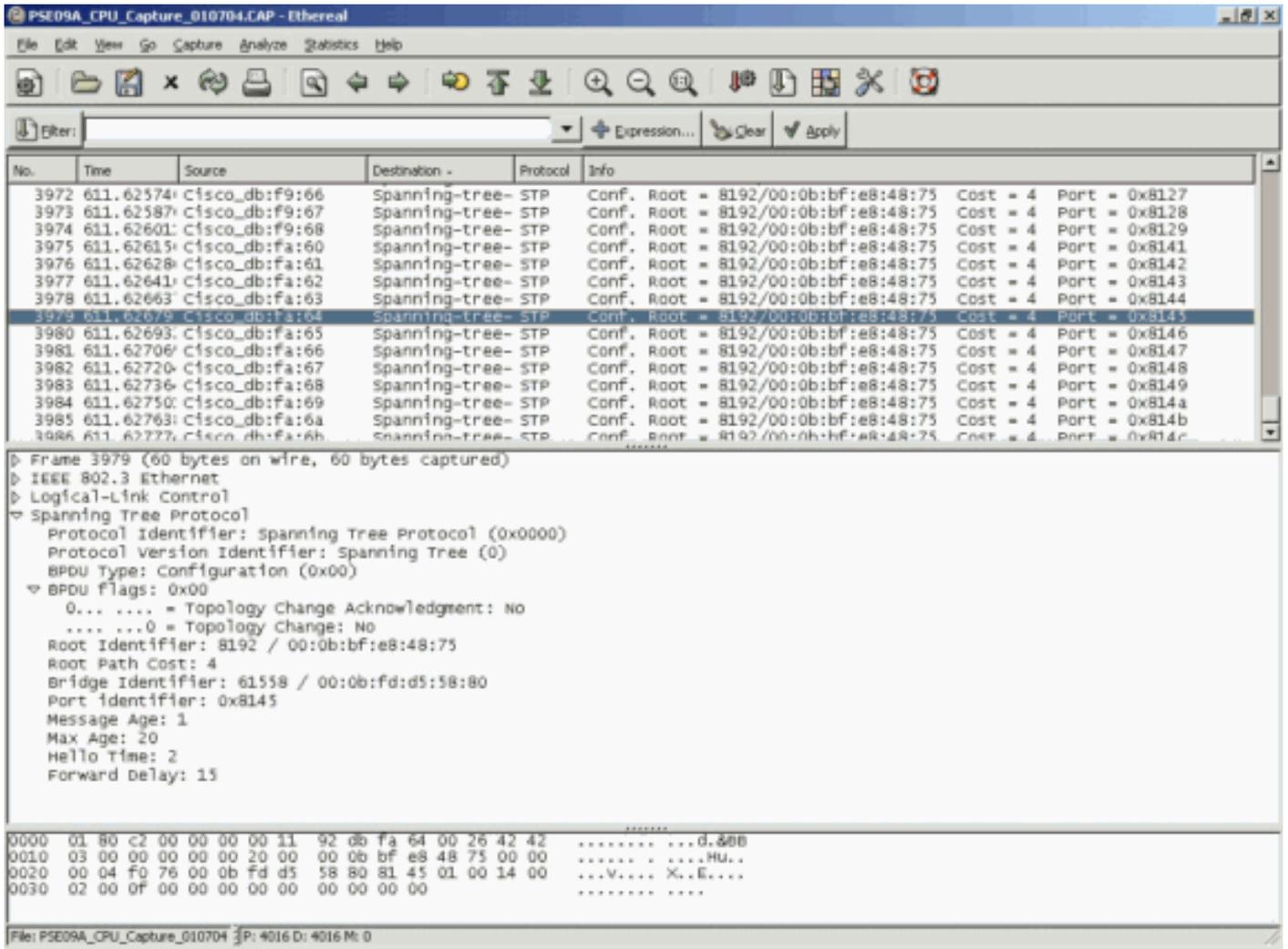
```
Type                : Local Session
Source Ports        :
  RX Only           : CPU
Destination Ports   : Gi1/3
  Encapsulation     : Native
  Ingress           : Disabled
  Learning          : Disabled
```

스니퍼 프로그램을 실행하는 PC를 연결하는 경우 트래픽을 빠르게 분석할 수 있습니다. 이 섹션의 창에 표시되는 출력에서 높은 CPU 사용률의 원인이 과도한 STP BPDU의 수량임을 확인할 수 있습니다.



**참고:** CPU 스니퍼의 STP BPDU는 정상입니다. 그러나 예상보다 더 많은 정보가 표시되는 경우 Supervisor Engine에 대한 권장 제한을 초과했습니다. 자세한 내용은 이 문서의 다수의 스페닝 트리 포트 인스턴스 섹션을 참조하십시오.

---



높은 CPU 사용률은 과도한 STP BPDU 수입니다

## 틀 2: 내장 CPU 스니퍼—Cisco IOS Software 릴리스 12.2(20)EW 이상

Catalyst 4500은 CPU에 영향을 주는 트래픽을 빠르게 식별할 수 있도록 내장된 CPU 스니퍼 및 디코더를 제공합니다. 이 섹션의 예에서 보여주는 debug 것처럼 명령을 사용하여 이 기능을 활성화할 수 있습니다. 이 기능은 한 번에 1,024개 패킷을 유지할 수 있는 순환 버퍼를 구현합니다. 새 패킷이 도착하면 이전 패킷을 덮어씁니다. 이 기능은 높은 CPU 사용률 문제를 해결할 때 안전하게 사용할 수 있습니다.

```
<#root>
```

```
Switch#
```

```
debug platform packet all receive buffer
```

```
platform packet debugging is on
```

Switch#

show platform cpu packet buffered

Total Received Packets Buffered: 36

-----  
Index 0:

7 days 23:6:32:37214 - RxVlan: 99,

RxPort: Gi4/48

Priority: Crucial, Tag: Dot1Q Tag, Event: Control Packet, Flags: 0x40, Size: 68  
Eth:

Src 00-0F-F7-AC-EE-4F

Dst 01-00-0C-CC-CC-CD Type/Len 0x0032

Remaining data:

0: 0xAA 0xAA 0x3 0x0 0x0 0xC 0x1 0xB 0x0 0x0  
10: 0x0 0x0 0x0 0x80 0x0 0x0 0x2 0x16 0x63 0x28  
20: 0x62 0x0 0x0 0x0 0x0 0x80 0x0 0x0 0x2 0x16  
30: 0x63 0x28 0x62 0x80 0xF0 0x0 0x0 0x14 0x0 0x2  
40: 0x0 0xF 0x0 0x0 0x0 0x0 0x0 0x2 0x0 0x63

Index 1:

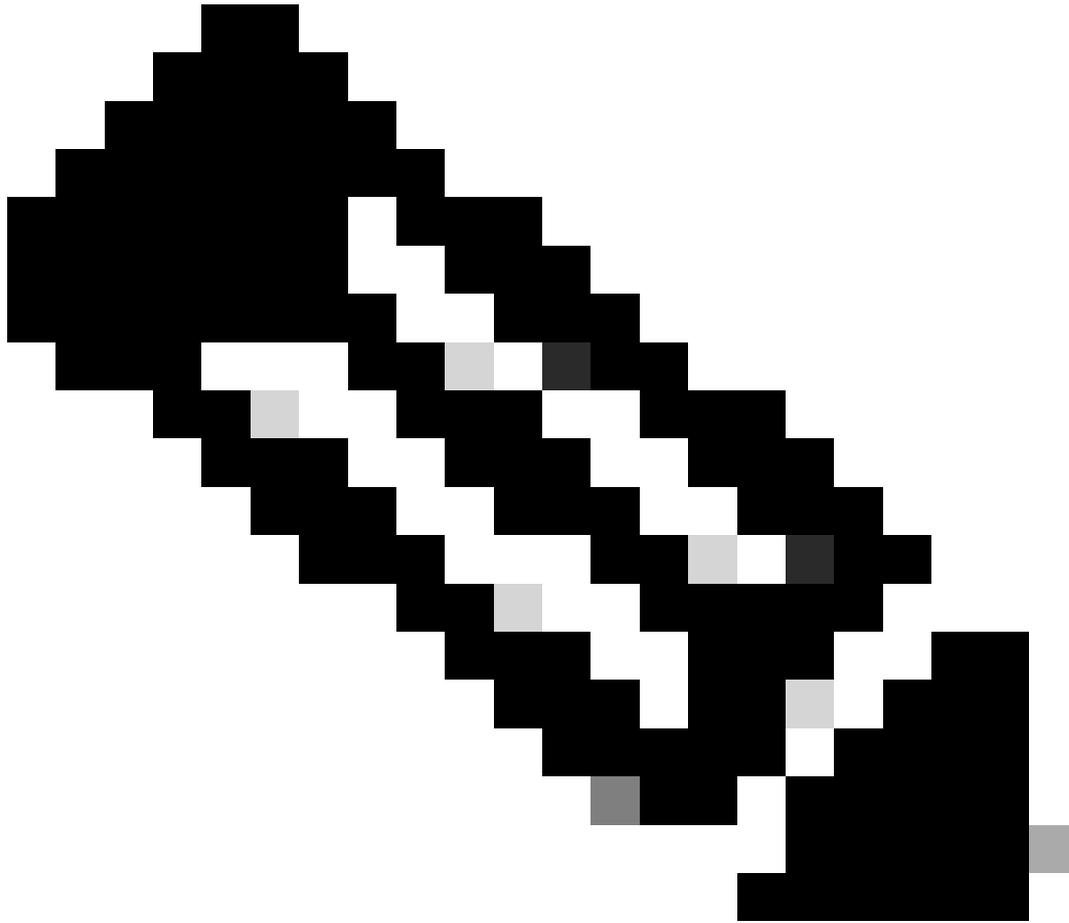
7 days 23:6:33:180863 - RxVlan: 1, RxPort: Gi4/48

Priority: Crucial, Tag: Dot1Q Tag, Event: Control Packet, Flags: 0x40, Size: 68

Eth: Src 00-0F-F7-AC-EE-4F Dst 01-00-0C-CC-CC-CD Type/Len 0x0032

Remaining data:

0: 0xAA 0xAA 0x3 0x0 0x0 0xC 0x1 0xB 0x0 0x0  
10: 0x0 0x0 0x0 0x80 0x0 0x0 0x2 0x16 0x63 0x28  
20: 0x62 0x0 0x0 0x0 0x0 0x80 0x0 0x0 0x2 0x16  
30: 0x63 0x28 0x62 0x80 0xF0 0x0 0x0 0x14 0x0 0x2  
40: 0x0 0xF 0x0 0x0 0x0 0x0 0x0 0x2 0x0 0x63



**참고:** 명령을 실행할 때의 CPU 사용률debug 은 항상 거의 100%입니다. 명령을 실행할 때 CPU 사용률이 높은 것이 debug 일반적입니다.

---

### 툴 3: CPU에 트래픽을 전송하는 인터페이스 식별—Cisco IOS Software 릴리스 12.2(20)EW 이상

Catalyst 4500은 CPU 처리를 위해 트래픽/패킷을 전송하는 상위 인터페이스를 식별하는 또 다른 유용한 툴을 제공합니다. 이 툴을 사용하면 많은 수의 브로드캐스트 또는 기타 서비스 거부 공격을 CPU에 전송하는 심부름 디바이스를 빠르게 식별할 수 있습니다. 이 기능은 높은 CPU 사용률 문제를 해결할 때도 안전하게 사용할 수 있습니다.

Switch#

debug platform packet all count

platform packet debugging is on  
Switch#

show platform cpu packet statistics

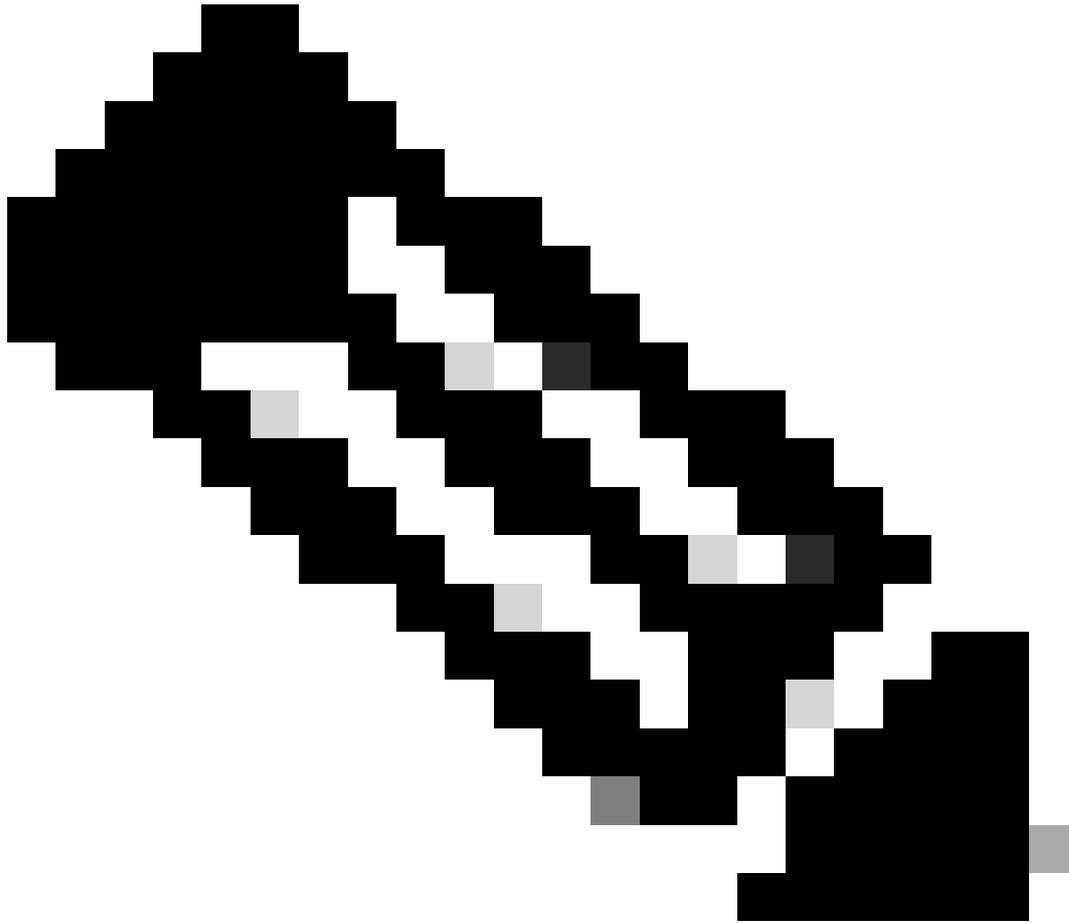
*!--- Output suppressed.*

Packets Transmitted from CPU per Output Interface

Interface	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Gi4/47	1150	1	5	10	0
Gi4/48	50	1	0	0	0

Packets Received at CPU per Input Interface

Interface	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Gi4/47	23130	5	10	50	20
Gi4/48	50	1	0	0	0



**참고:** debug 명령을 실행할 때의 CPU 사용률은 항상 거의 100%입니다. debug 명령어를 실행할 때 CPU 사용률이 높은 것이 일반적입니다.

## 요약

Catalyst 4500 스위치는 하드웨어에서 높은 속도의 IPv4(IP version 4) 패킷 전달을 처리합니다. 일부 기능 또는 예외로 인해 CPU 프로세스 경로를 통해 일부 패킷이 전달될 수 있습니다. Catalyst 4500은 정교한 QoS 메커니즘을 사용하여 CPU 바인딩 패킷을 처리합니다. 이 메커니즘은 스위치의 신뢰성과 안정성을 보장하는 동시에 패킷의 소프트웨어 전달을 위해 CPU를 최대화합니다. Cisco IOS 소프트웨어 릴리스 12.2(25)EWA2 이상에서는 어카운팅은 물론 패킷/프로세스 처리를 위한 추가 개선 사항을 제공합니다. 또한 Catalyst 4500에는 CPU 사용률이 높은 시나리오의 근본 원인을 파악하는 데 도움이 되는 충분한 명령과 강력한 툴이 있습니다. 그러나 대부분의 경우 Catalyst 4500의 CPU 사용률이 높더라도 네트워크가 불안정해지거나 문제가 되지 않습니다.

## 관련 정보

- [CatOS 소프트웨어를 실행하는 Catalyst 4500/4000, 2948G, 2980G 및 7500G 스위치의 CPU 사용률](#)
- [LAN 제품 지원 페이지](#)
- [LAN 스위칭 지원 페이지](#)
- [기술 지원 및 문서 - Cisco Systems](#)

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.