

Catalyst 스위치의 STP 문제 해결

목차

[소개](#)

[배경 정보](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기 규칙](#)

[STP 오류의 원인](#)

[전달 루프 문제 해결](#)

[1. 루프 식별](#)

[2. 루프의 토폴로지\(범위\) 검색](#)

[3. 루프 끊기](#)

[4. 루프 원인 찾기 및 수정](#)

[5. 이중화 복원](#)

[토폴로지 변경 내용 조사](#)

[범람의 원인 찾기](#)

[TC의 출처 찾기](#)

[과도한 TC를 방지하기 위한 단계 수행](#)

[컨버전스 시간 관련 문제 해결](#)

[STP 디버그 명령 사용](#)

[전달 루프로부터 네트워크 보호](#)

[1. 모든 스위치 간 링크에서 UDLD\(Unidirectional Link Detection\)를 활성화합니다](#)

[2. 모든 스위치에서 Loop Guard 활성화](#)

[3. 모든 엔드 스테이션 포트에서 Portfast 활성화](#)

[4. EtherChannel을 양쪽\(지원되는 경우\) 및 Non-SilentOption에서 DesirableMode로 설정합니다.](#)

[5. 스위치 간 링크에서 자동 협상\(지원되는 경우\)을 비활성화하지 마십시오](#)

[6. STP 타이머를 조정할 때는 주의하십시오](#)

[7. DoS\(Denial of Service\) 공격이 가능한 경우, Root Guard를 사용하여 네트워크 STP 경계를 보호합니다](#)

[8. Portfast 지원 포트에서 BPDU Guard를 활성화하여 STP가 포트에 연결된 인증되지 않은 네트워크 장치\(예: 허브, 스위치 및 브리징 라우터\)의 영향을 받지 않도록 합니다](#)

[9. 관리 VLAN에서 사용자 트래픽 방지](#)

[10. 예측 가능한\(하드코딩된\) STP 루트 및 백업 STP 루트 배치](#)

[관련 정보](#)

소개

이 문서에서는 Cisco IOS® 소프트웨어를 사용하여 STP(Spanning Tree Protocol) 문제를 해결하는 방법에 대해 설명합니다.

배경 정보

Catalyst 6500/6000에만 적용되는 특정 명령이 있습니다. 그러나 Cisco IOS 소프트웨어를 실행하는 모든 Cisco Catalyst 스위치에는 대부분의 원칙을 적용할 수 있습니다.

대부분의 STP 문제에는 다음 세 가지 문제가 있습니다.

- 루프를 전달하는 중입니다.
- 높은 STP 토폴로지 변경 비율(TC)로 인한 과도한 플러딩
- 컨버전스 시간과 관련된 문제.

브리지에는 특정 패킷이 여러 번 전달되는지 여부를 추적하는 메커니즘이 없기 때문에(예: IP TTL[Time to Live]) 네트워크에서 너무 오래 순환하는 트래픽을 삭제하는 데 사용됩니다. 동일한 L2(Layer 2) 도메인의 두 디바이스 간에는 하나의 경로만 존재할 수 있습니다.

STP의 목적은 STP 알고리즘을 기반으로 중복 포트를 차단하고, 중복 물리적 토폴로지를 트리와 같은 토폴로지로 해결하는 것입니다. 이중화 토폴로지의 어떤 포트도 차단되지 않고 트래픽이 무한정 원형으로 전달되는 경우 포워딩 루프(예: STP 루프)가 발생합니다.

포워딩 루프가 시작되면 해당 경로를 따라 가장 낮은 대역폭의 링크가 혼잡해집니다. 모든 링크의 대역폭이 동일한 경우 모든 링크가 혼잡해집니다. 이러한 혼잡으로 인해 패킷이 손실되고 영향을 받는 L2 도메인의 네트워크 중단 상황이 발생합니다.

홍수가 지나치면 증상이 뚜렷하지 않다. 느린 링크는 플러딩 트래픽으로 인해 혼잡해질 수 있으며, 이러한 혼잡한 링크 뒤에 있는 디바이스 또는 사용자는 느린 연결 또는 총 연결 손실을 경험할 수 있습니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 다양한 스페닝 트리 유형 및 구성 방법 자세한 [내용은 STP 및 IEEE 802.1s](#) MST 구성을 참조하십시오.
- 다양한 스페닝 트리 기능 및 구성 방법 자세한 [내용은 STP](#) 기능 구성을 참조하십시오.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Catalyst 6500(Supervisor 2 엔진 포함)
- Cisco IOS Software 릴리스 12.1(13)E

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 팁 표기 규칙](#)을 참조하십시오.

STP 오류의 원인

STP는 운영 환경에 대해 특정 가정을 합니다. 다음은 이 문서와 가장 관련이 있는 가정입니다.

- 두 브리지 사이의 각 링크는 양방향입니다. 이는 A가 B에게 직접 연결하면 A는 B가 보낸 것을 받고 B는 A가 보낸 것을 받는다는 것을 그 사이에 링크가 있는 한 받는 것이다.
- STP를 실행하는 각 브리지는 STP 패킷이라고도 하는 STP BPDU(Bridge Protocol Data Unit)를 정기적으로 수신, 처리 및 전송할 수 있습니다.

이러한 가정은 논리적이고 명백하게 보이지만, 충족되지 않는 상황이 있다. 이러한 상황의 대부분은 일종의 하드웨어 문제와 관련이 있습니다. 그러나 소프트웨어 결함은 STP 장애로 이어질 수도 있습니다. 다양한 하드웨어 장애, 컨피그레이션 오류, 연결 문제로 인해 대부분의 STP 장애가 발생하는 반면, 소프트웨어 장애는 소수를 차지합니다. STP 실패는 스위치 간에 존재하는 불필요한 추가 연결로 인해 발생할 수도 있습니다. 이러한 추가 연결 때문에 VLAN이 다운 상태가 됩니다. 이 문제를 해결하려면 스위치 간의 원치 않는 연결을 모두 제거하십시오.

이러한 가정들 중 하나가 충족되지 않을 때, 하나 이상의 브리지는 BPDU들을 수신 또는 프로세싱할 수 없다. 이는 브리지(또는 브리지)가 네트워크 토폴로지를 검색하지 않음을 의미합니다. 올바른 토폴로지에 대한 지식이 없으면 스위치에서 루프를 차단할 수 없습니다. 따라서 플러딩된 트래픽은 루핑된 토폴로지를 순환하고 모든 대역폭을 소비하며 네트워크를 중단합니다.

스위치에서 BPDU를 수신할 수 없는 이유로는 불량 트랜시버 또는 GBIC(Gigabit Interface Converter), 케이블 문제 또는 포트, 라인 카드 또는 수퍼바이저 엔진의 하드웨어 장애가 있습니다. STP 실패의 한 가지 일반적인 이유는 브리지 간의 단방향 링크입니다. 그러한 경우 하나의 브리지가 BPDU를 전송하지만 다운스트림 브리지는 이를 수신하지 않습니다. 스위치에서 수신한 BPDU를 처리할 수 없기 때문에 오버로드된 CPU(99% 이상)에 의해 STP 처리가 중단될 수도 있습니다. BPDU는 하나의 브리지에서 다른 브리지로의 경로를 따라 손상될 수 있으며, 이는 또한 적절한 STP 동작을 방지합니다.

포워딩 루프 이외에도, 차단된 포트가 없을 경우, 트래픽을 차단하는 포트를 통해 특정 패킷만 잘못 전달되는 상황이 있습니다. 대부분의 경우 이는 소프트웨어 문제로 인해 발생합니다. 이러한 행동은 "슬로우 루프(slow-loop)"를 유발할 수 있습니다. 즉, 일부 패킷은 루프되지만 링크의 혼잡이 없으므로 트래픽의 대부분은 여전히 네트워크를 통해 흐릅니다.

전달 루프 문제 해결

포워딩 루프는 원래(원인) 및 효과 모두에서 매우 다양합니다. STP에 영향을 미칠 수 있는 다양한 문제로 인해 이 문서에서는 전달 루프 트러블슈팅 방법에 대한 일반적인 가이드라인만 제공할 수 있습니다.

문제 해결을 시작하기 전에 다음 정보가 필요합니다.

- 모든 스위치 및 브리지를 자세히 설명하는 실제 토폴로지 다이어그램
- 해당 포트 번호(상호 연결됨).
- STP 컨피그레이션 세부사항(예: 어떤 스위치가 루트이고 백업 루트인지, 링크의 기본 비용 또는 우선순위가 아니며, 트래픽을 차단하는 포트의 위치)

1. 루프 식별

네트워크에서 전달 루프가 발생한 경우 일반적인 증상은 다음과 같습니다.

- 영향을 받는 네트워크 영역에 대한 연결, 네트워크 영역 간 연결 및 네트워크 영역을 통한 연결 끊김
- 영향을 받는 세그먼트 또는 VLAN에 연결된 라우터의 높은 CPU 사용률로 라우팅 프로토콜 네이버 플래핑 또는 HSRP(Hot Standby Router Protocol) 활성 라우터 플래핑과 같은 다양한 증상이 발생할 수 있습니다.
- 높은 링크 사용률(대개 100%).
- 높은 스위치 백플레인 사용률(베이스라인 사용률과 비교).
- 네트워크의 패킷 루프를 나타내는 Syslog 메시지(예: HSRP 중복 IP 주소 메시지).
- Syslog 메시지로, 지속적인 주소 재교육 또는 MAC 주소 플래핑 메시지를 나타냅니다.
- 많은 인터페이스에서 출력이 삭제되는 횟수가 증가합니다.

이러한 이유 중 하나만으로도 서로 다른 문제를 나타낼 수 있습니다(또는 문제가 전혀 없음). 그러나 이러한 요소가 동시에 많이 관찰되는 경우, 네트워크에 포워딩 루프가 발생했을 가능성이 높습니다. 이를 확인하는 가장 빠른 방법은 스위치 백플레인 트래픽 사용률을 확인하는 것입니다.

```
<#root>
```

```
cat#
```

```
show catalyst6000 traffic-meter
```

```
traffic meter = 13%
```

```
Never cleared
```

```
peak = 14%
```

```
reached at 12:08:57 CET Fri Oct 4 2002
```

 참고: Cisco IOS 소프트웨어가 설치된 Catalyst 4000에서는 현재 이 명령을 지원하지 않습니다



현재 트래픽 수준이 과도하거나 기준 수준을 알 수 없는 경우에는 최근 피크 수준이 달성되었는지, 현재 트래픽 수준에 근접했는지 등을 확인합니다. 예를 들어, 피크 트래픽 레벨이 15%이고 2분 전에 도달했으며 현재 트래픽 레벨이 14%이면 스위치의 로드가 비정상적으로 높습니다. 트래픽 로드가 정상 수준이면 루프가 없거나 이 디바이스가 루프에 포함되어 있지 않음을 의미할 수 있습니다. 하지만, 그것은 여전히 슬로우 루프에 연루될 수 있다.

2. 루프의 토폴로지(범위) 검색

네트워크 중단이 포워딩 루프인 것으로 확인되면 루프를 중지하고 네트워크 작업을 복원하는 것이 가장 우선입니다.

루프를 중지하려면 루프에 참여하는 포트를 알아야 합니다. 링크 사용률(초당 패킷 수)이 가장 높은 포트를 확인하십시오. `show interface` Cisco IOS software 명령은 각 인터페이스의 사용률을 표시합니다.

(빠른 분석을 위해) 사용률 정보와 인터페이스 이름만 표시하려면 Cisco IOS 소프트웨어로 정규식 출력을 필터링합니다. 방법 인터페이스 발행 | `include line|\sec` command - 초당 패킷 통계 및 인터페이스 이름만 표시합니다.

<#root>

cat#

```
show interface | include line|\sec
```

```
GigabitEthernet2/1 is up, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/2 is up, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/3 is up, line protocol is up
  5 minute input rate 99765230 bits/sec, 24912 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/4 is up, line protocol is up
  5 minute input rate 1000 bits/sec, 27 packets/sec
  5 minute output rate 101002134 bits/sec, 25043 packets/sec
GigabitEthernet2/5 is administratively down, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/6 is administratively down, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/7 is up, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

GigabitEthernet2/8 is up, line protocol is up


5 minute input rate 2000 bits/sec, 41 packets/sec


5 minute output rate 99552940 bits/sec, 24892 packets/sec


링크 사용률이 가장 높은 인터페이스에 유의하십시오. 이 예에서는 인터페이스 g2/3, g2/4 및 g2/8이며, 루프에 참여하는 포트입니다.

3. 루프 끊기

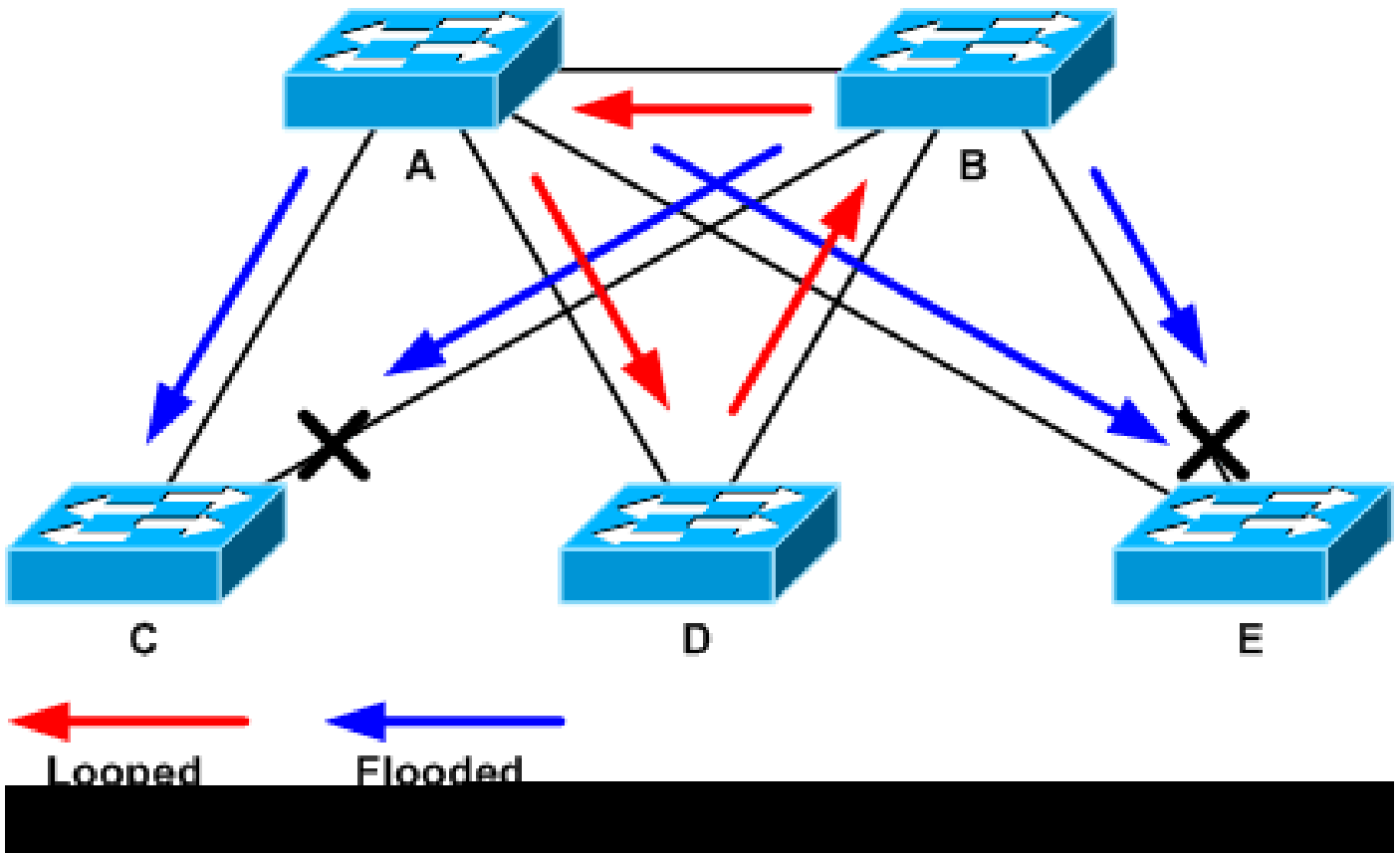
루프를 끊으려면 관련 포트를 종료하거나 연결을 끊어야 합니다. 루프를 중단하는 것뿐만 아니라 루프의 근본 원인을 찾아 고치는 것도 특히 중요합니다. 고리를 끊는 것이 비교적 쉽다

 참고: 모든 포트를 동시에 종료하거나 연결을 끊을 필요는 없습니다. 한 번에 하나씩 종료할 수 있습니다. 분배나 코어 스위치와 같이 루프의 영향을 받는 어그리게이션 지점에서 포트를 종료하는 것이 좋습니다. 모든 포트를 한 번에 종료하고 하나씩 활성화하거나 다시 연결하면 작동하지 않습니다. 루프가 중지되고 결함이 있는 포트가 다시 연결된 후 즉시 시작할 수 없습니다. 따라서 특정 포트에 대한 장애의 상관성을 파악하기 어렵습니다.

 참고: 루프를 중단하려면 스위치를 재부팅하기 전에 정보를 수집하는 것이 좋습니다. 그렇지 않으면 후속 근본 원인 분석이 어렵습니다. 각 포트를 비활성화하거나 연결을 끊은 후 스위치 백플레인 사용률이 정상 수준으로 돌아왔는지 확인해야 합니다.

 참고: 포트가 루프를 유지하지는 않지만 루프를 통해 도착하는 트래픽을 플러딩한다는 점에 유의하십시오. 이러한 플러딩 포트를 종료하면 백플레인 사용률이 약간 감소할 뿐 루프는 멈추지 않습니다.

다음 예의 토폴로지에서는 스위치 A, B, D 간에 루프가 있습니다. 따라서 링크 AB, AD, BD는 지속된다. 이러한 링크를 종료하면 루프가 중지됩니다. 링크 AC, AE, BC 및 BE는 루프와 함께 도착하는 트래픽을 플러딩합니다.



루프 및 플러딩 트래픽

지원 포트가 종료되면 백플레인 사용률이 정상 값으로 저하됩니다. 어떤 포트의 종료로 인해 백플레인 사용률(및 다른 포트의 사용률)이 정상 수준이 되었는지 알아야 합니다.

이 시점에서는 루프가 중지되고 네트워크 운영이 개선되지만, 루프의 원래 원인이 고정되지 않았기 때문에 여전히 다른 문제가 있습니다.

4. 루프 원인 찾기 및 수정

루프가 중지되면 루프가 시작된 이유를 확인해야 합니다. 이유는 다양할 수 있기 때문에 이 과정이 어려운 부분이다. 모든 경우에 적용되는 정확한 절차를 공식화하는 것도 어렵다.

지침:

- 토폴로지 다이어그램을 조사하여 중복 경로를 찾습니다. 여기에는 이전 단계에서 찾은 지원 포트가 포함되며, 이는 동일한 스위치(루프 중에 이야기되는 경로 패킷)로 돌아옵니다. 이전 예제 토폴로지에서 이 경로는 AD-DB-BA입니다.
- 이중화 경로의 모든 스위치에 대해 스위치가 올바른 STP 루트를 알고 있는지 확인합니다.

L2 네트워크의 모든 스위치는 공통 STP 루트에 동의해야 합니다. 브리지가 특정 VLAN 또는 STP 인스턴스에서 STP 루트의 다른 ID를 일관되게 표시할 경우 문제가 발생할 수 있다는 명백한 증상입니다. show spanning-tree vlan vlan-idcommand를 실행하여 지정된 VLAN의 루트 브리지 ID를 표시합니다.

<#root>

```
cat#
```

```
show spanning-tree vlan 333
```

```
MST03
```

```
Spanning tree enabled protocol mstp
```

```
Root ID      Priority      32771
             Address      0050.14bb.6000
             Cost          20000
             Port          136 (GigabitEthernet3/8)
             Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID    Priority      32771 (priority 32768 sys-id-ext 3)
             Address      00d0.003f.8800
             Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec
```

| Interface | Role | Sts | Cost | Prio.Nbr | Status |
|-----------|------|-----|-------|----------|--------|
| Gi3/8 | Root | FWD | 20000 | 128.136 | P2p |
| Po1 | Desg | FWD | 20000 | 128.833 | P2p |

VLAN 번호는 루프와 관련된 포트가 이전 단계에서 설정되었기 때문에 포트에서 찾을 수 있습니다. 문제의 포트가 트렁크인 경우 트렁크의 모든 VLAN이 관여하는 경우가 많습니다. 그렇지 않은 경우 (예: 루프가 단일 VLAN에서 발생한 경우) 이러한 방법 인터페이스를 실행할 수 있습니다 | include L2|line|broadcastcommand(수퍼바이저 1은 VLAN별 스위칭 통계를 제공하지 않으므로 Catalyst 6500/6000 Series 스위치의 수퍼바이저 2 이상 엔진에만 해당) VLAN 인터페이스만 살펴봅니다. 스위치드 패킷 수가 가장 많은 VLAN은 루프가 발생한 VLAN인 경우가 많습니다.

```
<#root>
```

```
cat#
```

```
show interface | include L2|line|broadcast
```

```
Vlan1 is up, line protocol is up
  L2 Switched: ucast: 653704527 pkt, 124614363025 bytes - mcast:
    23036247 pkt, 1748707536 bytes
  Received 23201637 broadcasts, 0 runts, 0 giants, 0 throttles

Vlan10 is up, line protocol is up
  L2 Switched: ucast: 2510912 pkt, 137067402 bytes - mcast:
    41608705 pkt, 1931758378 bytes
  Received 1321246 broadcasts, 0 runts, 0 giants, 0 throttles

Vlan11 is up, line protocol is up
  L2 Switched: ucast: 73125 pkt, 2242976 bytes - mcast:
    3191097 pkt, 173652249 bytes
  Received 1440503 broadcasts, 0 runts, 0 giants, 0 throttles

Vlan100 is up, line protocol is up
  L2 Switched: ucast: 458110 pkt, 21858256 bytes - mcast:
    64534391 pkt, 2977052824 bytes
  Received 1176671 broadcasts, 0 runts, 0 giants, 0 throttles

Vlan101 is up, line protocol is up
  L2 Switched: ucast: 70649 pkt, 2124024 bytes - mcast:
```



```
2175964 pkt, 108413700 bytes
Received 1104890 broadcasts, 0 runts, 0 giants, 0 throttles
```

이 예에서 VLAN 1은 가장 많은 브로드캐스트 수와 L2 스위치 트래픽을 차지합니다. 루트 포트가 올바르게 식별되었는지 확인합니다.

루트 포트는 루트 브리지에 대한 비용이 가장 낮아야 합니다. 저속 포트의 비용이 높기 때문에 홉(hop) 측면에서 경로는 짧지만 비용 측면에서는 더 긴 경우도 있습니다. 어떤 포트가 지정된 VLAN의 루트로 간주되는지 확인하려면 `show spanning-tree vlan` 명령을 실행합니다.

```
<#root>
```

```
cat#
```

```
show spanning-tree vlan 333
```

```
MST03
```

```
Spanning tree enabled protocol mstp
Root ID    Priority    32771
           Address    0050.14bb.6000
           Cost      20000
```

```
Port      136 (GigabitEthernet3/8)
```

```
        Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
Bridge ID Priority    32771 (priority 32768 sys-id-ext 3)
Address    00d0.003f.8800
Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

| Interface | Role | Sts | Cost | Prio.Nbr | Status |
|-----------|------|-----|-------|----------|--------|
| Gi3/8 | Root | FWD | 20000 | 128.136 | P2p |
| Po1 | Desg | FWD | 20000 | 128.833 | P2p |

BPDU가 루트 포트 및 차단할 포트에서 정기적으로 수신되는지 확인합니다.

BPDU는 모든 HELLOINTERVAL에서 루트 브리지에 의해 전송됩니다(기본 2초). 비루트 브리지는 루트에서 수신된 BPDU를 수신, 처리, 수정 및 전파합니다. `show spanning-tree interface detail` 명령을 실행하여 BPDU가 수신되는지 확인합니다.

```
<#root>
```

```
cat#
```

```
show spanning-tree interface g3/2 detail
```

```
Port 130 (GigabitEthernet3/2) of MST00 is backup blocking
Port path cost 20000, Port priority 128, Port Identifier 128.130.
```

```
Designated root has priority 0, address 0007.4f1c.e847
Designated bridge has priority 32768, address 00d0.003f.8800
Designated port id is 128.129, designated path cost 2000019
Timers: message age 4, forward delay 0, hold 0
```

```
Number of transitions to forwarding state: 0
```

```
Link type is point-to-point by default, Internal
Loop guard is enabled by default on the port
BPDU: sent 3,
```


```
received 53
```

```
cat#
```

```
show spanning-tree interface g3/2 detail
```

```
Port 130 (GigabitEthernet3/2) of MST00 is backup blocking
Port path cost 20000, Port priority 128, Port Identifier 128.130.
Designated root has priority 0, address 0007.4f1c.e847
Designated bridge has priority 32768, address 00d0.003f.8800
Designated port id is 128.129, designated path cost 2000019
Timers: message age 5, forward delay 0, hold 0
Number of transitions to forwarding state: 0
Link type is point-to-point by default, Internal
Loop guard is enabled by default on the port
BPDU: sent 3,
```

```
received 54
```

 참고: 명령의 두 출력 사이에 하나의 BPDU가 수신되었습니다(카운터는 53에서 54로 변경됨).

표시된 카운터는 실제로 STP 프로세스 자체에서 유지 관리되는 카운터입니다. 즉, 수신 카운터가 증가하면 물리적 포트에서 BPDU를 수신했을 뿐만 아니라 STP 프로세스에서도 수신했습니다. 이 received BPDU 카운터는 루트 대체 포트 또는 백업 포트에 간주되는 포트에서 증가하지 않습니다. 그런 다음 포트에서 멀티캐스트를 받는지(BPDU는 멀티캐스트로 전송됨) 확인합니다. show interface interface counterscommand를 실행합니다.

```
<#root>
```

```
cat#
```

```
show interface g3/2 counters
```

```
Port          InOctets  InUcastPkts
InMcastPkts
InBcastPkts
Gi3/2         14873036          2
89387
```

```

Port          OutOctets  OutUcastPkts  OutMcastPkts  OutBcastPkts
Gi3/2         114365997      83776        732086         19

```

```
cat#
```

```
show interface g3/2 counters
```

```

Port          InOctets  InUcastPkts
InMcastPkts
InBcastPkts
Gi3/2         14873677      2
89391
0

```

```

Port          OutOctets  OutUcastPkts  OutMcastPkts  OutBcastPkts
Gi3/2         114366106      83776        732087         19

```

STP 포트 역할에 대한 간략한 설명은 루프 가드 및 BPDU 스큐 [탐지 기능을 사용하는 스페닝 트리 프로토콜 개선 사항](#)의 [Enhance STP with Loop Guard and BPDU Skew Detection](#) 섹션에서 확인할 수 있습니다. 수신된 BPDU가 없는 경우, 포트에서 오류가 계산되는지 확인합니다. show interface interface counters errorscommand를 실행합니다.

```
<#root>
```

```
cat#
```

```
show interface g4/3 counters errors
```

```

Port      Align-Err  FCS-Err  Xmit-Err  Rcv-Err  UnderSize  OutDiscards
Gi4/3     0          0        0         0         0          0

```

```

Port      Single-Col  Multi-Col  Late-Col  Excess-Col  Carri-Sen  Runts  Giants
Gi4/3     0           0         0         0           0         0      0

```

BPDU가 물리적 포트에서 수신되지만 여전히 STP 프로세스에 도달하지 않을 수 있습니다. 앞의 두 예에서 사용된 명령에서 일부 멀티캐스트가 수신되었고 오류가 계산되지 않은 것으로 표시되면 STP 프로세스 레벨에서 BPDU가 삭제되었는지 확인합니다. Catalyst 6500에서 remote command switch test spanning-tree process-stats 명령을 실행합니다.

```
<#root>
```

```
cat#
```

```
remote command switch test spanning-tree process-stats
```

```

-----TX STATS-----
transmission rate/sec      = 2
paks transmitted           = 5011226

```

```

paks transmitted (opt)      = 0
opt chunk alloc failures   = 0
max opt chunk allocated    = 0
-----RX STATS-----

receive rate/sec           = 1

paks received at stp isr   = 3947627
paks queued at stp isr    = 3947627

paks dropped at stp isr    = 0
drop rate/sec             = 0

paks dequeued at stp proc  = 3947627
paks waiting in queue     = 0
queue depth               = 7(max) 12288(total)
-----PROCESSING STATS-----
queue wait time (in ms)   = 0(avg) 540(max)
processing time (in ms)  = 0(avg) 4(max)
proc switch count        = 100
add vlan ports           = 20
time since last clearing  = 2087269 sec

```

이 예에서 사용되는 명령은 STP 프로세스 통계를 표시합니다. 삭제 카운터가 증가하지 않고 수신된 패킷이 증가하는지 확인하는 것이 중요합니다. 수신된 패킷이 증가하지는 않지만 물리적 포트에서 멀티캐스트를 수신할 경우, 스위치 대역 내 인터페이스(CPU의 인터페이스)에서 패킷이 수신되는지 확인합니다. 원격 명령 스위치 show ibc를 실행합니다. | Catalyst 6500/6000의 i rx_inputcommand:

```
<#root>
```

```
cat#
```

```
remote command switch show ibc | i rx_input
```

```
rx_inputs=
```

```
5626468
```

```
, rx_cumbytes=859971138
```

```
cat#
```


```
remote command switch show ibc | i rx_input
```

```
rx_inputs=
```

```
5626471
```

```
, rx_cumbytes=859971539
```

이 예에서는 출력 간에 대역 내 포트에서 23개의 패킷을 수신했음을 보여 줍니다.

 참고: 이 23개 패킷은 BPDU 패킷일 뿐만 아니라 대역 내 포트에서 수신한 모든 패킷에 대한 전역 카운터입니다.

BPDU가 로컬 스위치 또는 포트에서 삭제되었다는 표시가 없는 경우 링크의 다른 쪽에 있는 스위치로 이동하여 해당 스위치가 BPDU를 전송하는지 확인해야 합니다. BPDU가 루트가 아닌 지정된 포트에서 정기적으로 전송되는지 확인합니다. 포트 역할이 일치하는 경우, 포트가 BPDU를 전송하지만 네이버가 이를 수신하지 않습니다. BPDU가 전송되는지 확인합니다. `show spanning-tree interface detail` 명령을 실행합니다.

<#root>

cat#

```
show spanning-tree interface g3/1 detail
```

Port 129 (GigabitEthernet3/1) of MST00 is

designated

forwarding

```
Port path cost 20000, Port priority 128, Port Identifier 128.129.
Designated root has priority 0, address 0007.4f1c.e847
Designated bridge has priority 32768, address 00d0.003f.8800
Designated port id is 128.129, designated path cost 2000019
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 0
Link type is point-to-point by default, Internal
Loop guard is enabled by default on the port
```

BPDUs: sent 1774

, received 1

cat#

```
show spanning-tree interface g3/1 detail
```

Port 129 (GigabitEthernet3/1) of MST00 is

designated


forwarding

```
Port path cost 20000, Port priority 128, Port Identifier 128.129.
Designated root has priority 0, address 0007.4f1c.e847
Designated bridge has priority 32768, address 00d0.003f.8800
Designated port id is 128.129, designated path cost 2000019
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 0
Link type is point-to-point by default, Internal
Loop guard is enabled by default on the port
```

BPDUs: sent 1776

, received 1

이 예에서는 두 개의 BPDU가 출력 간에 전송됩니다.

 참고: STP 프로세스는 BPDU: sentcounter를 유지 관리합니다. 즉, 카운터는 BPDU가 물리적 포트를 향해 전송되었으며 전송되었음을 나타냅니다. 전송된 멀티캐스트 패킷에 대해 포트 카운터가 증가하는지 확인합니다. show interface interface counterscommand를 실행합니다. 이를 통해 BPDU 트래픽 흐름을 확인할 수 있습니다.

<#root>

cat#

```
show interface g3/1 counters
```

| Port | InOctets | InUcastPkts | InMcastPkts | InBcastPkts |
|-------|-----------|-------------|-------------|-------------|
| Gi3/1 | 127985312 | 83776 | 812319 | 19 |

| Port | OutOctets | OutUcastPkts |
|------|-----------|--------------|
|------|-----------|--------------|

OutMcastPkts

| | OutBcastPkts | |
|-------|--------------|------|
| Gi3/1 | 131825915 | 3442 |

872342

386

cat#

```
show interface g3/1 counters
```

| Port | InOctets | InUcastPkts | InMcastPkts | InBcastPkts |
|-------|-----------|-------------|-------------|-------------|
| Gi3/1 | 127985312 | 83776 | 812319 | 19 |

| Port | OutOctets | OutUcastPkts |
|------|-----------|--------------|
|------|-----------|--------------|

OutMcastPkts

| | OutBcastPkts | |
|-------|--------------|------|
| Gi3/1 | 131826447 | 3442 |

872346

386

이 모든 단계를 통해 BPDU가 수신, 전송 또는 처리되지 않는 스위치 또는 링크를 찾는 것이 좋습니다. STP가 포트에 대한 올바른 상태를 계산했을 수 있지만 제어 평면 문제로 인해 포워딩 하드웨어에서 이 상태를 설정할 수 없습니다. 포트가 하드웨어 레벨에서 차단되지 않은 경우 루프를 생성할 수 있습니다. 네트워크의 문제라고 생각되면 [Cisco 기술 지원팀에 추가](#) 지원을 요청하십시오.

5. 이중화 복원

루프를 일으키는 장치나 링크가 발견되면 이 장치를 네트워크에서 격리하거나 문제를 해결해야 합


니다(예: 파이버 또는 GBIC 교체). 3단계에서 연결이 끊긴 이중 링크를 복원해야 합니다.

루프를 일으키는 장치나 링크를 조작하지 않는 것이 중요합니다. 루프를 초래하는 많은 조건이 일시적이고 간헐적이며 불안정하기 때문입니다. 조사 과정에서 또는 조사 후 상태를 걸어내면 한동안 상태가 발생하지 않거나 아예 발생하지 않는다는 의미다. [Cisco Technical Support](#)에서 더 자세히 조사할 수 있도록 [조건을](#) 기록해야 합니다. 스위치를 재설정하기 전에 상태에 대한 정보를 수집하는 것이 중요합니다. 조건이 없어지면 루프의 근본 원인을 파악할 수 없습니다. 정보를 수집하는 경우 이 문제로 인해 루프가 다시 발생하지 않도록 해야 합니다. 자세한 내용은 전달 루프에 [대한 네트워크 보안을 참조하십시오.](#)

토폴로지 변경 내용 조사

토폴로지 변경(TC) 메커니즘의 역할은 토폴로지가 변경된 후 L2 포워딩 테이블을 수정하는 것입니다. 이는 이전에 특정 포트를 통해 액세스할 수 있었던 MAC 주소가 변경되어 다른 포트를 통해 액세스할 수 있기 때문에 연결 중단을 방지하기 위해 필요합니다. TC는 TC가 발생하는 VLAN의 모든 스위치에서 포워딩 테이블 기간을 단축합니다. 따라서 주소가 재작성되지 않으면 에이징되고 패킷이 대상 MAC 주소에 도달하도록 플러딩이 발생합니다.

TC는 포트의 STP 상태가 STPforwardingstate로 또는 STPforwardingstate에서 STP 상태로 변경됨에 따라 트리거됩니다. TC 이후에는 특정 목적지 MAC 주소가 노후화되더라도 플러딩이 오래 지속되지 않습니다. 주소는 MAC 주소가 에이징 아웃된 호스트에서 오는 첫 번째 패킷에 의해 다시 생성됩니다. TC가 짧은 간격으로 반복적으로 발생하는 경우 문제가 발생할 수 있다. 이 스위치는 지속적으로 포워딩 테이블을 빠르게 노후화하므로 플러딩이 거의 일정하게 발생할 수 있습니다.

 참고: Rapid STP 또는 Multiple STP(IEEE 802.1w 및 IEEE 802.1s)를 사용하는 경우, TC는 포트의 상태를 포워딩로 변경하고, 지정된에서의 역할 변경을 통해 트리거됩니다. Rapid STP를 사용하면 에이징 시간을 단축하는 802.1d와 달리 L2 포워딩 테이블이 즉시 플러시됩니다. 전달 테이블을 즉시 플러시하면 연결이 더 빨리 복원되지만 플러딩이 더 많이 발생할 수 있습니다

TC는 잘 구성된 네트워크에서 드문 이벤트입니다. 스위치 포트의 링크가 작동 또는 중단되면 결국 TC가 발생합니다. 포트의 STP 상태가 전달로 변경되거나 전달에서 변경됩니다. 포트가 펄럭이면 반복적인 TC와 플러딩이 발생합니다.

STP 포트패스트 기능이 활성화된 포트는 TC가 전달 상태로 이동하거나 전달 상태에서 이동할 때 TC를 발생시킬 수 없습니다. 프린터, PC 및 서버와 같은 모든 엔드 디바이스 포트에서 portfast를 구성하면 TC를 소량으로 제한할 수 있으므로 적극 권장합니다.

네트워크에 반복 TC가 있는 경우 이러한 TC의 소스를 식별하고 이를 줄이기 위한 조치를 취해야 플러딩을 최소화할 수 있습니다.

802.1d를 사용하면 TC 이벤트에 대한 STP 정보가 특수한 유형의 BPDU인 TCN(TC Notification)을 통해 브리지 간에 전파됩니다. TCN BPDU를 수신하는 포트를 따라가면 TC를 시작한 디바이스를 찾을 수 있습니다.

범람의 원인 찾기

성능 저하로 인한 플러딩, 혼잡하지 않아야 하는 링크의 패킷 삭제, 그리고 패킷 분석기가 로컬 세그먼트에 없는 동일한 대상에 대한 여러 유니캐스트 패킷을 표시하도록 결정할 수 있습니다. 유니캐스트 플러딩에 대한 자세한 내용은 [스위치드 캠퍼스 네트워크의 유니캐스트 플러딩을 참조하십시오](#).

Cisco IOS 소프트웨어를 실행하는 Catalyst 6500/6000에서는 포워딩 엔진 카운터(Supervisor 2 엔진에서만)를 확인하여 플러딩 양을 추정할 수 있습니다. 원격 명령 스위치 show earl statistics 실행 | MISS_DA|ST_FR명령:

```
<#root>
cat#
remote command switch show earl statistics | i MISS_DA|ST_FR

          ST_MISS_DA    =          18          530308834
          ST_FRMS       =          97          969084354

cat#
remote command switch show earl statistics | i MISS_DA|ST_FR

          ST_MISS_DA    =           4          530308838
          ST_FRMS       =          23          969084377
```

이 예에서 첫 번째 열에는 이 명령이 마지막으로 실행된 이후의 변경 사항이 표시되고, 두 번째 열에는 마지막 재부팅 이후의 누적 값이 표시됩니다. 첫 번째 줄에는 플러딩된 프레임의 양이 표시되고 두 번째 줄에는 처리된 프레임의 양이 표시됩니다. 두 값이 근접하거나 첫 번째 값이 빠른 속도로 증가하면 스위치가 트래픽을 플러딩할 수 있습니다. 그러나 카운터가 세분화되지 않으므로 플러딩을 확인하는 다른 방법과만 함께 사용할 수 있습니다. 포트나 VLAN이 아닌 스위치당 하나의 카운터가 있습니다. 목적지 MAC 주소가 포워딩 테이블에 없는 경우 스위치가 항상 플러딩할 수 있으므로 일부 플러딩 패킷을 확인하는 것이 일반적입니다. 스위치에서 아직 학습되지 않은 목적지 주소의 패킷을 수신하는 경우가 이에 해당합니다.

TC의 출처 찾기

과도한 플러딩이 발생하는 VLAN에 대해 VLAN 번호가 알려진 경우 STP 카운터를 확인하여 TC 수가 높은지 또는 정기적으로 증가하는지 확인합니다. show spanning-tree vlan vlan-id detail 명령을 실행합니다(이 예에서는 VLAN 1이 사용됨).

```
<#root>
cat#
show spanning-tree vlan 1 detail

VLAN0001 is executing the ieee compatible Spanning Tree protocol
Bridge Identifier has priority 32768, sysid 1, address 0007.0e8f.04c0
Configured hello time 2, max age 20, forward delay 15
```



```
Current root has priority 0, address 0007.4f1c.e847
Root port is 65 (GigabitEthernet2/1), cost of root path is 119
Topology change flag not set, detected flag not set
```


```
Number of topology changes 1 last change occurred 00:00:35 ago
from GigabitEthernet1/1
```

```
Times: hold 1, topology change 35, notification 2
hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0, aging 300
```

VLAN 번호를 모르는 경우 패킷 분석기를 사용하거나 모든 VLAN에 대한 TC 카운터를 확인할 수 있습니다.

과도한 TC를 방지하기 위한 단계 수행

토폴로지 변경 카운터의 수를 모니터링하여 정기적으로 증가하는지 확인할 수 있습니다. 그런 다음 표시된 포트에 연결된 브리지로 이동하여 마지막 TC(이전 예에서는 포트 GigabitEthernet1/1)를 수신하고 해당 브리지에 대한 TC가 어디에서 왔는지 확인합니다. 이 프로세스는 STP portfast가 활성화되지 않은 엔드 스테이션 포트를 찾거나 수정해야 하는 플랩 링크를 찾을 때까지 반복해야 합니다. TC들이 다른 소스들로부터 나오는 경우, 전체 절차가 반복될 필요가 있다. 링크가 엔드 호스트에 속하는 경우 TC가 생성되지 않도록 portfast 기능을 구성할 수 있습니다.

 참고: Cisco IOS 소프트웨어 STP 구현에서 TCN BPDU가 VLAN의 포트에서 수신된 경우에만 TC에 대한 카운터가 증가할 수 있습니다. 설정된 TC 플래그를 갖는 정상 컨피그레이션 BPDU가 수신되는 경우, TC 카운터는 증가되지 않는다. 즉, TC가 플러딩 원인이라고 의심되면 해당 VLAN의 STP 루트 브리지에서 TC의 소스를 추적하기 시작합니다. TC들의 수 및 소스에 대한 가장 정확한 정보를 가질 수 있다.

컨버전스 시간 관련 문제 해결

STP의 실제 작동과 예상 동작이 일치하지 않는 경우가 있습니다. 다음은 가장 자주 발생하는 두 가지 문제입니다.

- STP 통합 또는 재통합은 예상보다 시간이 오래 걸립니다.
- 토폴로지 결과가 예상과 다릅니다.


이러한 행동을 하게 된 가장 큰 원인은 다음과 같습니다.

- 실제 토폴로지와 문서화된 토폴로지가 일치하지 않습니다.
- 컨피그레이션 오류(예: STP 타이머의 일관성 없는 컨피그레이션, STP 직경이 증가하거나 잘못된 컨피그레이션 발생)
- 컨버전스 또는 리컨버전스 중에 스위치 CPU가 오버로드되었습니다.
- 소프트웨어 결함.

앞에서 언급한 것처럼 이 문서에서는 STP에 영향을 줄 수 있는 다양한 문제로 인해 트러블슈팅에 대한 일반적인 지침만 제공할 수 있습니다. 컨버전스가 예상보다 오래 걸리는 이유를 알아보려면 STP 이벤트의 순서를 확인하여 어떤 상황이 발생하는지, 어떤 순서로 발생하는지 알아보십시오. Cisco IOS Software의 STP 구현에서는 결과를 로깅하지 않으므로(포트 불일치와 같은 특정 이벤트 제외) Cisco IOS Software를 사용하여 STP를 디버그하면 더 명확한 보기가 가능합니다. Cisco IOS 소프트웨어를 실행하는 Catalyst 6500/6000을 사용하는 STP의 경우 SP(Switch Processor)(또는 Supervisor)에서 처리가 수행되므로 SP에서 디버그를 활성화해야 합니다. Cisco IOS 소프트웨어 브리지 그룹의 경우 RP(Route Processor)에서 처리가 수행되므로 RP(MSFC)에서 디버그를 활성화해야 합니다.

STP 디버그 명령 사용

많은 STPdebugcommands는 개발 엔지니어링 용도로 사용됩니다. Cisco IOS 소프트웨어의 STP 구현에 대한 자세한 지식이 없는 사용자에게는 의미 있는 출력을 제공하지 않습니다. 일부 디버그는 포트 상태 변경, 역할 변경, TC와 같은 이벤트, 수신 및 전송된 BPDU의 덤프와 같이 즉시 읽을 수 있는 출력을 제공할 수 있습니다. 이 섹션에서는 모든 디버그에 대한 완전한 설명을 제공하는 것이 아니라 가장 자주 사용되는 디버그에 대해 간략하게 소개합니다.

 참고: debugcommands를 사용할 때 필요한 최소 디버그를 활성화합니다. 실시간 디버깅이 필요하지 않은 경우 출력을 콘솔에 인쇄하지 않고 로그에 기록합니다. 과도한 디버그는 CPU를 오버로드하고 스위치 작업을 중단시킬 수 있습니다.

디버그 출력을 콘솔 또는 텔넷 세션이 아닌 로그로 전달하려면 글로벌 컨피그레이션 모드에서 logging console informational and no logging monitor commands를 실행합니다. 일반 이벤트 로그를 보려면 PVST(Per VLAN Spanning-Tree) 및 Rapid-PVST에 대한 debug spanning-tree eventcommand를 실행합니다. 이는 STP의 상황에 대한 정보를 제공하는 첫 번째 디버그입니다. MST(다중 스페닝 트리) 모드에서는 debug spanning-tree eventcommand를 실행하지 않습니다. 따라서 debug spanning-tree mstp rolescommand를 실행하여 포트 역할 변경 사항을 확인합니다. 포트 STP 상태 변경 사항을 보려면 debug pm vpccommand와 함께 debug spanning-tree switch statecommand를 실행합니다.

```
<#root>
```

```
cat-sp#
```

```
debug spanning-tree switch state
```

```
Spanning Tree Port state changes debugging is on
```

```
cat-sp#
```

```
debug pm vp
```

```
Virtual port events debugging is on
```

```
Nov 19 14:03:37: SP: pm_vp 3/1(333): during state forwarding, got event 4(remove)
```

```
Nov 19 14:03:37: SP:
```

```
@@@
```

pm_vp 3/1(333):
forwarding -> notforwarding

port 3/1 (was forwarding) goes down in vlan 333

Nov 19 14:03:37: SP: *** vp_fwdchange: single: notfwd: 3/1(333)
Nov 19 14:03:37: SP: @@@ pm_vp 3/1(333): notforwarding -> present
Nov 19 14:03:37: SP: *** vp_linkchange: single: down: 3/1(333)
Nov 19 14:03:37: SP: @@@ pm_vp 3/1(333): present -> not_present
Nov 19 14:03:37: SP: *** vp_statechange: single: remove: 3/1(333)

Nov 19 14:03:37: SP: pm_vp 3/2(333): during state notforwarding,
got event 4(remove)

Nov 19 14:03:37: SP:

@@@

pm_vp 3/2(333): notforwarding -> present

Nov 19 14:03:37: SP: *** vp_linkchange: single: down: 3/2(333)

Port 3/2 (was not forwarding) in vlan 333 goes down

Nov 19 14:03:37: SP: @@@ pm_vp 3/2(333): present -> not_present
Nov 19 14:03:37: SP: *** vp_statechange: single: remove: 3/2(333)

Nov 19 14:03:53: SP: pm_vp 3/1(333): during state not_present,
got event 0(add)

Nov 19 14:03:53: SP: @@@ pm_vp 3/1(333): not_present -> present

Nov 19 14:03:53: SP: *** vp_statechange: single: added: 3/1(333)

Nov 19 14:03:53: SP: pm_vp 3/1(333): during state present,
got event 8(linkup)

Nov 19 14:03:53: SP:

@@@

pm_vp 3/1(333): present ->
notforwarding

Nov 19 14:03:53: SP: STP SW: Gi3/1 new blocking req for 0 vlans

Nov 19 14:03:53: SP: *** vp_linkchange: single: up: 3/1(333)

Port 3/1 link goes up and blocking in vlan 333

Nov 19 14:03:53: SP: pm_vp 3/2(333): during state not_present,
got event 0(add)

Nov 19 14:03:53: SP: @@@ pm_vp 3/2(333): not_present -> present

Nov 19 14:03:53: SP: *** vp_statechange: single: added: 3/2(333)

Nov 19 14:03:53: SP: pm_vp 3/2(333): during state present,
got event 8(linkup)

Nov 19 14:03:53: SP:

@@@

pm_vp 3/2(333): present ->
notforwarding

Nov 19 14:03:53: SP: STP SW: Gi3/2 new blocking req for 0 vlans

Nov 19 14:03:53: SP: *** vp_linkchange: single: up: 3/2(333)

Port 3/2 goes up and blocking in vlan 333

```
Nov 19 14:04:08: SP: STP SW: Gi3/1 new learning req for 1 vlans
Nov 19 14:04:23: SP: STP SW: Gi3/1 new forwarding req for 0 vlans
Nov 19 14:04:23: SP: STP SW: Gi3/1 new forwarding req for 1 vlans
Nov 19 14:04:23: SP:      pm_vp 3/1(333): during state notforwarding,
      got event 14(forward_notnotify)
Nov 19 14:04:23: SP:
```

```
@@@ pm_vp 3/1(333): notforwarding ->
      forwarding
```

```
Nov 19 14:04:23: SP: *** vp_list_fwdchange: forward: 3/1(333)
```

```
Port 3/1 goes via learning to forwarding in vlan 333
```

STP가 특정 방식으로 작동하는 이유를 이해하려면 스위치에서 보내고 받은 BPDU를 확인하는 것이 유용합니다.

```
<#root>
```

```
cat-sp#
```

```
debug spanning-tree bpdu receive
```

```
Spanning Tree BPDU Received debugging is on
```

```
Nov 6 11:44:27: SP: STP: VLAN1 rx BPDU: config protocol = ieee,
      packet from GigabitEthernet2/1 , linktype IEEE_SPANNING ,
      enctype 2, encsize 17
```

```
Nov 6 11:44:27: SP: STP: enc 01 80 C2 00 00 00 00 06 52 5F 0E 50 00 26 42 42 03
```

```
Nov 6 11:44:27: SP: STP: Data 000000000000000074F1CE8470000001380480006525F0E4
      080100100140002000F00
```

```
Nov 6 11:44:27: SP: STP: VLAN1 Gi2/1:0000 00 00 00 000000074F1CE847 00000013
      80480006525F0E40 8010 0100 1400 0200 0F00
```

이 디버그는 PVST, Rapid-PVST 및 MST 모드에서 작동하지만 BPDU의 내용을 디코딩하지는 않습니다. 그러나 BPDU를 수신하는 데 사용할 수 있습니다. BPDU의 내용을 보려면 PVST 및 Rapid-PVST에 대한 debug spanning-tree switch rx processcommand와 함께 debug spanning-tree switch rx decodecommand를 실행합니다. debug spanning-tree mstp bpdu-rxcommand를 실행하여 MST용 BPDU의 내용을 확인합니다.

```
<#root>
```

```
cat-sp#
```

```
debug spanning-tree switch rx decode
```

```
Spanning Tree Switch Shim decode received packets debugging is on
```

```
cat-sp#
```

```
debug spanning-tree switch rx process
```

Spanning Tree Switch Shim process receive bpdu debugging is on

```
Nov 6 12:23:20: SP: STP SW: PROC RX: 0180.c200.0000<-0006.525f.0e50 type/len 0026
Nov 6 12:23:20: SP:      encap SAP linktype ieee-st vlan 1 len 52 on v1 Gi2/1
Nov 6 12:23:20: SP:      42 42 03 SPAN
Nov 6 12:23:20: SP:      CFG P:0000 V:00 T:00 F:00 R:0000 0007.4f1c.e847 00000013
Nov 6 12:23:20: SP:      B:8048 0006.525f.0e40 80.10 A:0100 M:1400 H:0200 F:0F00

Nov 6 12:23:22: SP: STP SW: PROC RX: 0180.c200.0000<-0006.525f.0e50 type/len 0026
Nov 6 12:23:22: SP:      encap SAP linktype ieee-st vlan 1 len 52 on v1 Gi2/1
Nov 6 12:23:22: SP:      42 42 03 SPAN
Nov 6 12:23:22: SP:      CFG P:0000 V:00 T:00 F:00 R:0000 0007.4f1c.e847 00000013
Nov 6 12:23:22: SP:      B:8048 0006.525f.0e40 80.10 A:0100 M:1400 H:0200 F:0F00
```

MST 모드의 경우 thisdebugcommand를 사용하여 자세한 BPDU 디코딩을 활성화할 수 있습니다.

<#root>

cat-sp#

debug spanning-tree mstp bpdu-rx

Multiple Spanning Tree Received BPDUs debugging is on

```
Nov 19 14:37:43: SP: MST:BPDU DUMP [
```

```
rcvd_bpdu Gi3/2
```


```
Repeated]
```

```
Nov 19 14:37:43: SP: MST:  Proto:0 Version:3 Type:2 Role: DesgFlags[ F ]
Nov 19 14:37:43: SP: MST:  Port_id:32897 cost:2000019
Nov 19 14:37:43: SP: MST:  root_id :0007.4f1c.e847 Prio:0
Nov 19 14:37:43: SP: MST:  br_id   :00d0.003f.8800 Prio:32768
Nov 19 14:37:43: SP: MST:  age:2 max_age:20 hello:2 fwdelay:15
Nov 19 14:37:43: SP: MST:  V3_len:90 PathCost:30000 region:STATIC rev:1
Nov 19 14:37:43: SP: MST:  ist_m_id :0005.74
Nov 19 14:37:43: SP: MST:BPDU DUMP [
```

```
rcvd_bpdu Gi3/2
```

```
Repeated]
```

```
Nov 19 14:37:43: SP: MST:  Proto:0 Version:3 Type:2 Role: DesgFlags[ F ]
Nov 19 14:37:43: SP: MST:  Port_id:32897 cost:2000019
Nov 19 14:37:43: SP: MST:  root_id :0007.4f1c.e847 Prio:0
Nov 19 14:37:43: SP: MST:  br_id   :00d0.003f.8800 Prio:32768
Nov 19 14:37:43: SP: MST:  age:2 max_age:20 hello:2 fwdelay:15
Nov 19 14:37:43: SP: MST:  V3_len:90 PathCost:30000 region:STATIC rev:1
Nov 19 14:37:43: SP: MST:  ist_m_id :0005.7428.1440 Prio:32768 Hops:18
  Num Mrec: 1
Nov 19 14:37:43: SP: MST: stci=3  Flags[ F ] Hop:19 Role:Desg [Repeated]
Nov 19 14:37:43: SP: MST:      br_id:00d0.003f.8800 Prio:32771 Port_id:32897
  Cost:2000028.1440 Prio:32768 Hops:18 Num Mrec: 1
Nov 19 14:37:43: SP: MST: stci=3  Flags[ F ] Hop:19 Role:Desg [Repeated]
Nov 19 14:37:43: SP: MST:      br_id:00d0.003f.8800 Prio:32771 Port_id:32897
  Cost:20000
```

 됩니다. 즉, 포트별 또는 VLAN별로 수신되거나 전송된 BPDU를 디버깅할 수 있습니다.

debug condition vlan vlan_num 또는 debug condition interface interface 명령을 실행하여 디버깅 출력의 범위를 인터페이스별 또는 VLAN별로 제한합니다.

전달 루프로부터 네트워크 보호

Cisco는 STP에서 특정 장애를 관리할 수 없을 때 포워딩 루프로부터 네트워크를 보호하기 위해 다양한 기능과 향상된 기능을 개발했습니다.

STP의 문제를 해결할 때 특정 장애의 원인을 찾아내고 격리하는 데 도움이 되며, 이러한 개선 사항의 구현은 포워딩 루프로부터 네트워크를 보호하는 유일한 방법입니다.

다음은 포워딩 루프로부터 네트워크를 보호하기 위한 방법입니다.


1. 모든 스위치 간 링크에서 UDLD(Unidirectional Link Detection)를 활성화합니다

UDLD에 대한 자세한 내용은 단방향 [링크 탐지 프로토콜 기능의 이해 및 구성을 참조하십시오](#).


2. 모든 스위치에서 Loop Guard 활성화

Loop Guard에 대한 자세한 내용은 Loop [Guard 및 BPDU Skew Detection 기능을 사용한 스페닝 트리 프로토콜 개선 사항을 참조하십시오](#).

활성화된 경우 UDLD 및 Loop Guard는 포워딩 루프가 발생하는 대부분의 원인을 제거합니다. 전달 루프를 생성하지 않고 결함이 있는 링크(또는 결함이 있는 하드웨어에 종속된 모든 링크)가 종료되거나 차단됩니다.


 참고: 이 두 기능은 다소 중복되어 보이지만, 각각 고유한 기능을 갖추고 있습니다. 따라서 두 기능을 동시에 사용하여 최고 수준의 보호를 제공합니다. UDLD와 Loop Guard의 자세한 비교는 [Loop Guard vs. Unidirectional Link Detection](#)을 참조하십시오.

공격적인 UDLD를 사용해야 하는지, 정상적인 UDLD를 사용해야 하는지에 대해서는 이견이 있다. 어그레시브 UDLD는 일반 모드 UDLD에 비해 루프를 더 많이 보호할 수 없습니다. Aggressive UDLD는 포트 스택(port-stuck) 시나리오를 탐지합니다(링크가 작동하지만 연결된 트래픽 블랙홀이 없는 경우). 이 추가된 기능의 단점은 일관성 있는 장애가 없을 때 공격적인 UDLD가 잠재적으로 링크를 비활성화할 수 있다는 것입니다. UDLDhellointerval의 수정과 공격적인 UDLD 기능을 혼동하는 경우가 많습니다. 이것은 틀렸습니다. 타이머는 두 UDLD 모드에서 모두 수정할 수 있습니다.

 참고: 드문 경우이지만 공격적인 UDLD는 모든 업링크 포트를 종료하여 스위치를 네트워크의 나머지 부분과 격리시킬 수 있습니다. 예를 들어, 두 업스트림 스위치의 CPU 사용률이 매우 높으며 적극적인 모드 UDLD가 사용되는 경우 이 문제가 발생할 수 있습니다. 따라서 스위치에 대역 외 관리가 없는 경우 시간 제한을 구성하여 시간을 줄일 수 없는 것이 좋습니다.

3. 모든 엔드 스테이션 포트에서 Portfast 활성화

네트워크 성능에 영향을 줄 수 있는 TC 및 후속 플러딩의 양을 제한하려면 portfast를 활성화해야 합니다. 엔드 스테이션에 연결하는 포트에만 이 명령을 사용합니다. 그렇지 않으면 실수로 인한 토폴로지 루프로 인해 데이터 패킷 루프가 발생하여 스위치 및 네트워크 운영이 중단될 수 있습니다.

 주의: no spanning-tree portfast 명령을 사용할 때는 주의해야 합니다. 이 명령은 포트 관련 portfast 명령만 제거합니다. 이 명령은 글로벌 컨피그레이션 모드에서 spanning-tree portfast default 명령을 정의하고 포트가 트렁크 포트가 아닌 경우 portfast를 암시적으로 활성화합니다. portfast를 전역적으로 구성하지 않으면 no spanning-tree portfast 명령은 spanning-tree portfast disable 명령과 같습니다.

4. EtherChannel을 양측(지원되는 경우) 및 무음 옵션에서 바람직한 모드로 설정합니다.

바람직한 모드는 채널링 피어 간의 런타임 일관성을 보장하기 위해 PAgP(Port Aggregation Protocol)를 활성화할 수 있습니다. 따라서 특히 채널 재컨피그레이션(예: 링크가 채널에 연결되거나 이탈하는 경우, 링크 장애 탐지) 중에 루프에 대한 추가적인 보호 기능이 제공됩니다. 기본적으로 활성화되어 있으며 채널 구성 오류 또는 기타 조건으로 인한 전달 루프를 방지하는 Channel Misconfiguration Guard가 내장되어 있습니다. 이 기능에 대한 자세한 내용은 EtherChannel [불일치 탐지 이해를 참조하십시오](#).

5. 스위치 간 링크에서 자동 협상(지원되는 경우)을 비활성화하지 마십시오

자동 협상 메커니즘은 원격 측에서 장애를 탐지하는 가장 빠른 방법인 원격 결함 정보를 전달할 수 있습니다. 원격측에서 장애가 감지되면 링크가 펄스를 얻더라도 로컬 측에서 링크를 내려 놓습니다. UDLD와 같은 상위 레벨 탐지 메커니즘에 비해 자동 협상은 매우 빠르지만(마이크로초 이내) UDLD의 엔드 투 엔드 범위가 부족합니다(예: 전체 데이터 경로: CPU—포워딩 로직—port1—port2—포워딩 로직—CPU vs port1—port2). Aggressive UDLD 모드는 장애 탐지와 관련하여 자동 협상과 유사한 기능을 제공합니다. 링크의 양쪽에서 협상이 지원되는 경우 어그레시브 모드 UDLD를 활성화할 필요가 없습니다.

6. STP 타이머를 조정할 때는 주의하십시오

STP 타이머는 서로 및 네트워크 토폴로지에 따라 달라집니다. STP는 타이머에 대한 임의의 수정 사항으로 올바르게 작동하지 않습니다. STP 타이머에 대한 자세한 내용은 스패닝 트리 [프로토콜 타이머 이해 및 튜닝을 참조하십시오](#).

7. DoS(Denial of Service) 공격이 가능한 경우, Root Guard를 사용하여 네트워크 STP 경계를 보호합니다

Root Guard 및 BPDU Guard를 사용하면 외부로부터의 영향을 받지 않도록 STP를 보호할 수 있습니다. 이러한 공격이 가능하다면 루트 가드 및 BPDU 가드를 사용하여 네트워크를 보호해야 합니다. Root Guard 및 BPDU Guard에 대한 자세한 내용은 다음 문서를 참조하십시오.

- [스패닝 트리 프로토콜 루트 가드 개선](#)

- [스패닝 트리 PortFast BPDU 가드 개선](#)

8. Portfast 지원 포트에서 BPDU Guard를 활성화하여 STP가 포트에 연결된 인증되지 않은 네트워크 장치(예: 허브, 스위치 및 브리징 라우터)의 영향을 받지 않도록 합니다

Root Guard를 올바르게 구성하면 STP가 외부에서 영향을 받지 않습니다. BPDU Guard가 활성화된 경우 BPDU를 수신하는 포트를 종료합니다. 이는 BPDU Guard가 syslog 메시지를 생성하고 포트를 종료하므로 인시던트를 조사하는 데 유용합니다. 루트 또는 BPDU 가드가 짧은 사이클 루프를 방지하지 않는 경우, 고속 활성화된 포트 2개가 직접 또는 허브를 통해 연결됩니다.

9. 관리 VLAN에서 사용자 트래픽 방지

관리 VLAN은 전체 네트워크가 아닌 구성 요소에 포함됩니다.

스위치 관리 인터페이스는 관리 VLAN에서 브로드캐스트 패킷을 수신합니다. 과도한 브로드캐스트(예: 브로드캐스트 스톱 또는 오작동 애플리케이션)가 발생하면 스위치 CPU가 오버로드되어 STP 작업이 왜곡될 수 있습니다.

10. 예측 가능한(하드코딩된) STP 루트 및 백업 STP 루트 배치

STP 루트 및 백업 STP 루트는 장애 발생 시 컨버전스가 예측 가능한 방식으로 발생하고 모든 시나리오에서 최적의 토폴로지를 구축하도록 구성해야 합니다. 예측할 수 없는 루트 스위치 선택을 방지하기 위해 STP 우선순위를 기본값으로 두지 마십시오.

관련 정보

- [LAN 제품 지원](#)
- [LAN 스위칭 기술 지원](#)

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.