Nexus 스위치의 순환 이중화 검사 오류 이해

목차

소개

이 문서에서는 인터페이스 카운터와 Cisco Nexus 스위치 통계에서 관찰된 CRC(Cyclic Redundancy Check) 오류에 대한 세부 정보를 설명합니다.

사전 요구 사항

요구 사항

Cisco에서는 이더넷 스위칭과 Cisco NX-OS CLI(Command Line Interface)의 기본 사항을 이해하는 것이 좋습니다. 자세한 내용은 다음 관련 문서 중 하나를 참조하십시오.

- Cisco Nexus 9000 NX-OS 기본 사항 컨피그레이션 가이드, 릴리스 10.2(x)
- Cisco Nexus 9000 Series NX-OS Fundamentals 컨피그레이션 가이드, 릴리스 9.3(x)
- Cisco Nexus 9000 Series NX-OS Fundamentals 컨피그레이션 가이드, 릴리스 9.2(x)
- Cisco Nexus 9000 Series NX-OS 기본 사항 컨피그레이션 가이드, 릴리스 7.x

- 이더넷 문제 해결

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- NX-OS 소프트웨어 릴리스 9.3(8)부터 시작하는 Nexus 9000 시리즈 스위치

- NX-OS 소프트웨어 릴리스 9.3(8)부터 시작하는 Nexus 3000 시리즈 스위치

이 문서의 정보는 특정 랩 환경의 디바이스에서 생성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

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

배경 정보

이 문서에서는 Cisco Nexus 시리즈 스위치의 인터페이스 카운터에서 관찰된 CRC(Cyclic Redundancy Check) 오류에 대한 세부 정보를 설명합니다. 이 문서에서는 CRC의 정의, 이더넷 프레임의 FCS(Frame Check Sequence) 필드에서 사용되는 방법, Nexus 스위치에서 CRC 오류가 발생하는 방법, Store-and-Forward 스위칭 및 Cut-Through 스위칭 시나리오에서 CRC 오류가 상호 작용하는 방법, CRC 오류의 근본 원인, CRC 오류의 문제 해결 및 해결 방법에 대해 설명합니다.

적용 가능한 하드웨어

이 문서의 정보는 모든 Cisco Nexus Series 스위치에 적용됩니다. 이 문서의 일부 정보는 Cisco Catalyst 라우터 및 스위치와 같은 다른 Cisco 라우팅 및 스위칭 플랫폼에도 적용할 수 있습니다.

CRC 정의

CRC는 전송 중에 변경되거나 손상된 데이터를 식별하기 위해 컴퓨터 및 저장소 네트워크에서 일반적으로 사용되는 오류 감지 메커니즘입니다. 네트워크에 연결된 장치가 데이터를 전송해야 할 경우 장치는 고정 길이 숫자를 생성하는 데이터에 대해 순환 코드를 기반으로 계산 알고리즘을 실행합니다. 이 고정 길이 숫자를 CRC 값이라고 하지만, 구어로는 단기간의 CRC라고 합니다. 이 CRC 값은 데이터에 추가되고 네트워크를 통해 다른 디바이스로 전송됩니다. 이 원격 디바이스는 데이터와 동일한 순환 코드 알고리즘을 실행하고 결과 값을 데이터에 추가된 CRC와 비교합니다. 두 값이 모두 일치하면 원격 디바이스는 데이터가 손상되지 않고 네트워크를 통해 전송된 것으로 가정합니다. 값이 일치하지 않으면 원격 디바이스는 네트워크를 통해 전송하는 동안 데이터가 손상된 것으로 간주합니다. 이 손상된 데이터는 신뢰할 수 없으며 삭제됩니다.

CRC는 이더넷(유무선 변형 모두), 토큰 링, ATM(Asynchronous Transfer Mode) 및 프레임 릴레이와 같은 여러 컴퓨터 네트워킹 기술 전반에서 오류 탐지에 사용됩니다. 이더넷 프레임에는 32비트 CRC 값이 삽입되는 프레임 끝(프레임의 페이로드 바로 후)에 32비트 FCS(Frame Check Sequence) 필드가 있습니다.

예를 들어, Host-A와 Host-B라는 두 호스트가 NIC(Network Interface Card)를 통해 서로 직접 연결되는 시나리오를 가정해보겠습니다. Host-A는 네트워크를 통해 Host-B에 "This is an example"이라는 문장을 보내야 합니다. Host-A는 Host-B로 향하는 이더넷 프레임을 "This is an example" 페이로드로 만들고 프레임의 CRC 값이 0xABCD의 16진수 값임을 계산합니다. Host-A는 0xCRC 값을 이더넷 프레임의 FCS 필드에 삽입한 다음 Host-A의 NIC에서 이더넷 프레임을 Host-B로 전송합니다.

Host-B가 이 프레임을 수신하면 Host-A와 정확히 동일한 알고리즘을 사용하여 프레임의 CRC 값을 계산합니다. Host-B는 프레임의 CRC 값이 0xABCD의 16진수 값임을 계산합니다. 이는 프레임이 Host-B로 전송되는 동안 이더넷 프레임이 손상되지 않았음을 Host-B에 나타냅니다.

CRC 오류 정의

CRC 오류는 디바이스(네트워크 디바이스 또는 네트워크에 연결된 호스트)가 프레임의 FCS 필드에 CRC 값이 있는 이더넷 프레임을 수신할 때 발생하며, 이는 해당 프레임의 디바이스에서 계산한 CRC 값과 일치하지 않습니다.

이 개념은 예를 통해 가장 잘 보여집니다. Host-A와 Host-B라는 두 호스트가 NIC(Network Interface Card)를 통해 서로 직접 연결되는 시나리오를 가정해 보겠습니다. Host-A는 네트워크를 통해 Host-B에 "This is an example"이라는 문장을 보내야 합니다. Host-A는 Host-B로 향하는 이더넷 프레임을 "This is an example" 페이로드로 만들고 프레임의 CRC 값이 16진수 값 0xABCD로 계산합니다. Host-A는 0xCRC 값을 이더넷 프레임의 FCS 필드에 삽입한 다음 Host-A의 NIC에서 이더넷 프레임을 Host-B로 전송합니다.

그러나 Host-A를 Host-B에 연결하는 물리적 미디어의 손상은 프레임 내의 문장이 "This is an example"이라는 원하는 페이로드 대신 "This was an example"으로 변경되도록 프레임의 내용을 손상시킵니다.

Host-B가 이 프레임을 수신하면 손상된 페이로드를 포함하여 프레임의 CRC 값이 계산됩니다. Host-B는 프레임의 CRC 값이 0xDEAD의 16진수 값인 0xDEAD로 계산하며, 이는 이더넷 프레임의 FCS 필드 내의 CRC 값과 다릅니다. CRC 값의 이 차이는 Host-B에 프레임이 Host-B로 전송되는 동안 이더넷 프레임이 손상되었음을 알려줍니다. 따라서 Host-B는 이 이더넷 프레임의 내용을 신뢰할 수 없으므로 삭제됩니다. Host-B는 일반적으로 "입력 오류", "CRC 오류" 또는 "RX 오류" 카운터와 같은 NIC(Network Interface Card)의 일부 오류 카운터를 증가시킵니다.

CRC 오류의 일반적인 증상

CRC 오류는 일반적으로 다음 두 가지 방법 중 하나로 나타납니다.

1. 네트워크에 연결된 디바이스의 인터페이스에 대한 오류 카운터 증가 또는 0이 아닌 오류 카운터
2. 네트워크 연결 디바이스가 손상된 프레임을 삭제하여 네트워크를 통과하는 트래픽의 패킷/프레임 손실.

이러한 오류는 작업 중인 디바이스에 따라 약간 다른 방식으로 나타납니다. 이러한 하위 섹션은 각 디바이스 유형에 대해 자세히 설명합니다.

Windows 호스트에서 수신된 오류

Windows 호스트의 CRC 오류는 일반적으로 명령 프롬프트의 netstat **-e** 명령 출력**에** 표시되는 0이 아닌 **받은 오류** 카운터로 나타납니다. Windows 호스트의 명령 프롬프트에서 0이 아닌 받은 오류 카운터의 예는 다음과 같습니다.

```
>netstat -e
Interface Statistics

                           Received            Sent
Bytes                    1116139893      3374201234
Unicast packets           101276400        49751195
Non-unicast packets               0               0
Discards                          0               0
Errors                        47294               0
```

```
Unknown protocols                 0
```
NIC와 해당 드라이버는 netstat -e 명령에서 보고한 수신 오류 수를 정확하게 기록하려면 NIC에서 수신한 CRC 오류**의** 어카운팅을 지원해야 합니다. 대부분의 최신 NIC 및 해당 드라이버는 NIC에서 수신한 CRC 오류를 정확하게 회계처리하는 것을 지원합니다.

Linux 호스트의 RX 오류

Linux 호스트의 CRC 오류는 일반적으로 ifconfig 명령 출력에 표시되는 0이 아닌 "RX 오류" 카운터로 나타납니다. Linux 호스트의 RX가 0이 아닌 오류 카운터 예는 다음과 같습니다.

```
$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.0.2.10  netmask 255.255.255.128  broadcast 192.0.2.255
       inet6 fe80::10  prefixlen 64  scopeid 0x20<link>
       ether 08:62:66:be:48:9b  txqueuelen 1000  (Ethernet)
       RX packets 591511682  bytes 214790684016 (200.0 GiB)
        RX errors 478920  dropped 0  overruns 0  frame 0
       TX packets 85495109  bytes 288004112030 (268.2 GiB)
       TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```
Linux 호스트의 CRC 오류는 **ip -s 링크 show** 명령 출력에 표시되는 0이 아닌 "RX 오류" 카운터로 나타날 수도 있습니다. Linux 호스트에서 0이 아닌 RX 오류 카운터의 예는 다음과 같습니다.

```
$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
   link/ether 08:62:66:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
   32246366102 444908978 478920       647     0       419445867
   TX: bytes  packets  errors  dropped carrier collsns
   3352693923 30185715 0       0       0       0
   altname enp11s0
```
ifconfig 또는 **ip -s 링크** show 명령이 정확하게 보고되는 RX 오류 수를 확인하려면 NIC와 해당 드라이버가 NIC에서 수신한 CRC 오류 어카운팅을 지원해야 합니다. 대부분의 최신 NIC 및 해당 드라이버는 NIC에서 수신한 CRC 오류를 정확하게 회계처리하는 것을 지원합니다.

네트워크 디바이스의 CRC 오류

네트워크 디바이스는 저장 및 전달 모드, 컷스루 전달 모드 중 하나로 작동합니다. 네트워크 디바이스에서 수신된 CRC 오류를 처리하는 방법은 전달 모드에 따라 다릅니다. 이 하위 섹션에서는 각 전달 모드에 대한 특정 동작을 설명합니다.

Store-and-Forward 네트워크 디바이스의 입력 오류

저장 및 전달 포워딩 모드에서 작동하는 네트워크 디바이스가 프레임을 수신하면 프레임의 CRC 값을 검증하고, 프레임에 포워딩 결정을 내리고, 인터페이스를 통해 프레임을 전송하기 전에 네트워크 디바이스가 전체 프레임("저장")을 버퍼링합니다("전달"). 따라서 Store-and-Forward 포워딩 모드에서 작동하는 네트워크 디바이스가 특정 인터페이스에서 잘못된 CRC 값으로 손상된 프레임을 수신하면 프레임이 삭제되고 인터페이스의 "Input Errors" 카운터가 증가합니다.

즉, 손상된 이더넷 프레임은 Store-and-Forward 포워딩 모드에서 작동하는 네트워크 디바이스에 의해 전달되지 않습니다. 인그레스(ingress)에 드롭됩니다.

Cisco Nexus 7000 및 7700 Series 스위치는 Store-and-Forward 포워딩 모드에서 작동합니다. Nexus 7000 또는 7700 Series 스위치에서 0이 아닌 입력 오류 카운터 및 0이 아닌 CRC/FCS 카운터의 예는 다음과 같습니다.

```
switch# show interface
<snip>
Ethernet1/1 is up
 RX
   241052345 unicast packets  5236252 multicast packets  5 broadcast packets
   245794858 input packets  17901276787 bytes
   0 jumbo packets  0 storm suppression packets
   0 runts  0 giants  579204 CRC/FCS  0 no buffer
   579204 input error  0 short frame  0 overrun   0 underrun  0 ignored
   0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
   0 input with dribble  0 input discard
   0 Rx pause
```

또한 CRC 오류는 **show interface counters** 오류의 출력에서 0이 아닌 "FCS-Err" 카운터로 나타날 수 있습니다. 이 명령의 출력에 있는 "Rcv-Err" 카운터에도 0이 아닌 값이 있습니다. 이는 인터페이스에서 받은 모든 입력 오류(CRC 또는 그 밖의 값)의 합계입니다. 이에 대한 예는 다음과 같습니다.

```
switch# show interface counters errors
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1                0     579204          0     579204          0            0
```

컷스루 네트워크 장치의 입력 및 출력 오류

컷스루 전달 모드에서 작동하는 네트워크 디바이스가 프레임을 수신하기 시작하면 네트워크 디바이스는 프레임의 헤더에 대해 포워딩 결정을 내리고 올바른 포워딩 결정을 내릴 수 있는 프레임을 충분히 수신하자마자 인터페이스에서 프레임 전송을 시작합니다. 프레임 및 패킷 헤더가 프레임의 시작 부분에 있으므로 이 전달 결정은 일반적으로 프레임의 페이로드를 수신하기 전에 수행됩니다
.

이더넷 프레임의 FCS 필드는 프레임의 페이로드 바로 뒤에 있는 프레임 끝에 있습니다. 따라서 컷스루 포워딩 모드에서 작동하는 네트워크 디바이스는 프레임의 CRC를 계산할 수 있을 때까지 다른 인터페이스에서 프레임을 전송하기 시작합니다. 프레임에 대한 네트워크 디바이스에서 계산된 CRC가 FCS 필드에 있는 CRC 값과 일치하지 않으면 네트워크 디바이스가 손상된 프레임을 네트워크로 전달했음을 의미합니다. 이 경우 네트워크 디바이스는 두 개의 카운터를 증가시킵니다.

1. 손상된 프레임이 원래 수신된 인터페이스의 "Input Errors" 카운터입니다.
2. 손상된 프레임이 전송된 모든 인터페이스의 "출력 오류" 카운터입니다. 유니캐스트 트래픽의 경우 일반적으로 단일 인터페이스가 됩니다. 그러나 브로드캐스트, 멀티캐스트 또는 알 수 없는 유니캐스트 트래픽의 경우 하나 이상의 인터페이스가 될 수 있습니다.

여기에 이 예제가 나와 있습니다. 여기서 **show interface** 명령의 출력은 네트워크 디바이스의 Ethernet1/1에서 여러 손상된 프레임을 수신하고 네트워크 디바이스의 컷스루 포워딩 모드 때문에 Ethernet1/2에서 전송되었음을 나타냅니다.

```
switch# show interface
<snip>
Ethernet1/1 is up
 RX
```

```
   46739903 unicast packets  29596632 multicast packets  0 broadcast packets
   76336535 input packets  6743810714 bytes
   15 jumbo packets  0 storm suppression bytes
   0 runts  0 giants  47294 CRC  0 no buffer
   47294 input error  0 short frame  0 overrun   0 underrun  0 ignored
   0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
   0 input with dribble  0 input discard
   0 Rx pause

Ethernet1/2 is up
 TX
   46091721 unicast packets  2852390 multicast packets  102619 broadcast packets
   49046730 output packets  3859955290 bytes
   50230 jumbo packets
   47294 output error  0 collision  0 deferred  0 late collision
   0 lost carrier  0 no carrier  0 babble  0 output discard
   0 Tx pause
```

또한 CRC 오류는 인그레스 인터페이스의 0이 아닌 "FCS-Err" 카운터로, show interface counters 오류의 출력에서 이그레스 인터페이스의 0이 아닌 "Xmit-Err" 카운터로 나타날 수 있습니다. 이 명령의 출력에 있는 인그레스 인터페이스의 "Rcv-Err" 카운터에도 0이 아닌 값이 있습니다. 이 값은 인터페이스에서 받은 모든 입력 오류(CRC 또는 그 밖의 값)의 합계입니다. 이에 대한 예는 다음과 같습니다.

```
switch# show interface counters errors
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1                0      47294          0      47294          0           0
Eth1/2                0          0      47294          0          0           0
```

또한 네트워크 디바이스는 프레임 FCS 필드의 CRC 값을 이 프레임이 손상되었음을 나타내는 특정 방식으로 수정합니다. 이 동작은 CRC를 "사용자 지정"한다고 합니다. CRC가 수정되는 정확한 방법은 한 플랫폼마다 다르지만 일반적으로 프레임의 FCS 필드에 있는 현재 CRC 값을 반전시키는 작업이 포함됩니다. 예를 들면 다음과 같습니다.

```
Original CRC: 0xABCD (1010101111001101)
Stomped CRC:  0x5432 (0101010000110010)
```

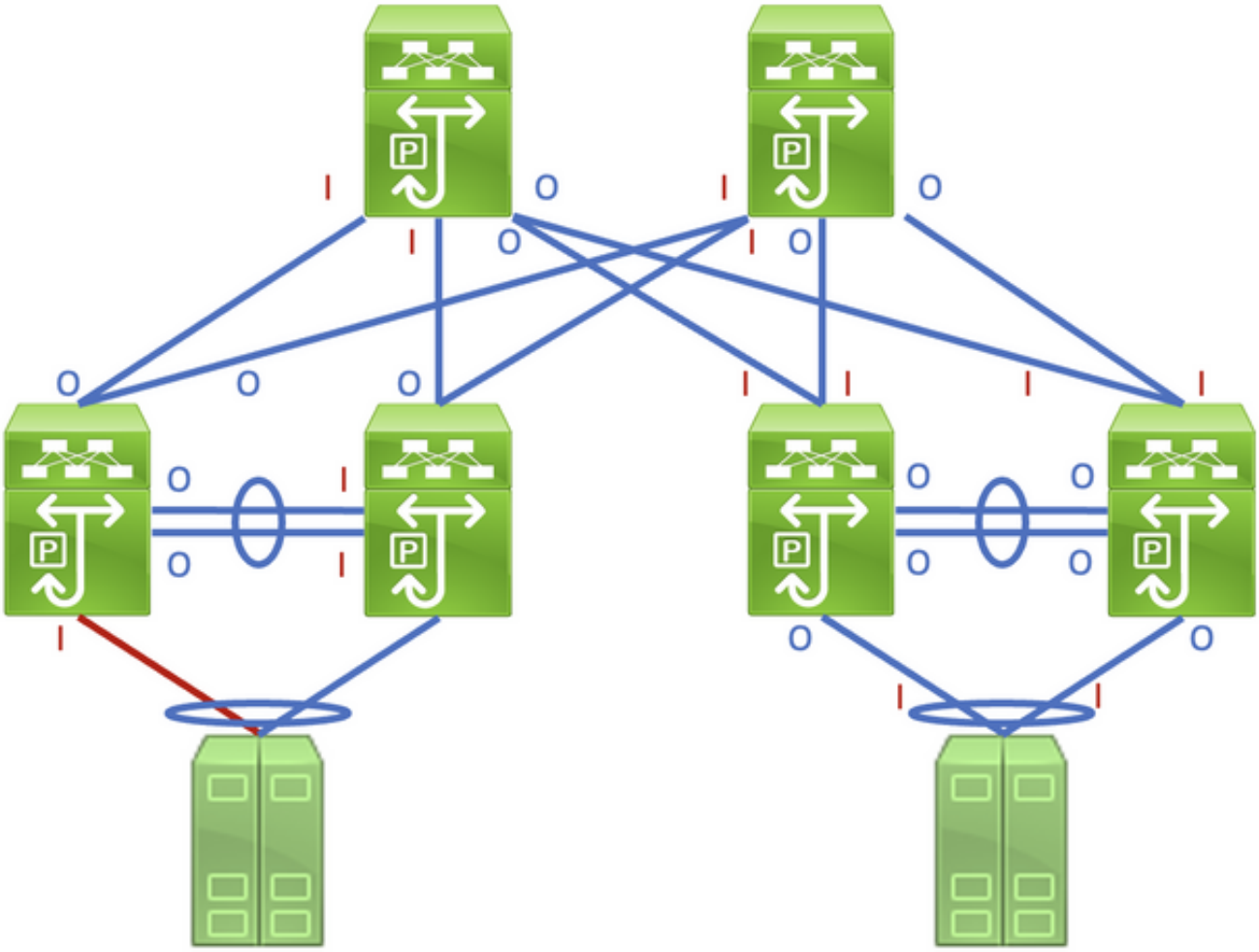
이러한 동작으로 인해 컷스루 전달 모드에서 작동하는 네트워크 장치가 손상된 프레임을 네트워크 전체에 전파할 수 있습니다. 컷스루 전달 모드에서 작동하는 여러 네트워크 장치로 구성된 네트워크가 손상된 경우 단일 프레임이 손상되면 네트워크 내의 여러 네트워크 장치에서 입력 오류 및 출력 오류 카운터가 증가할 수 있습니다.

CRC 오류 추적 및 격리

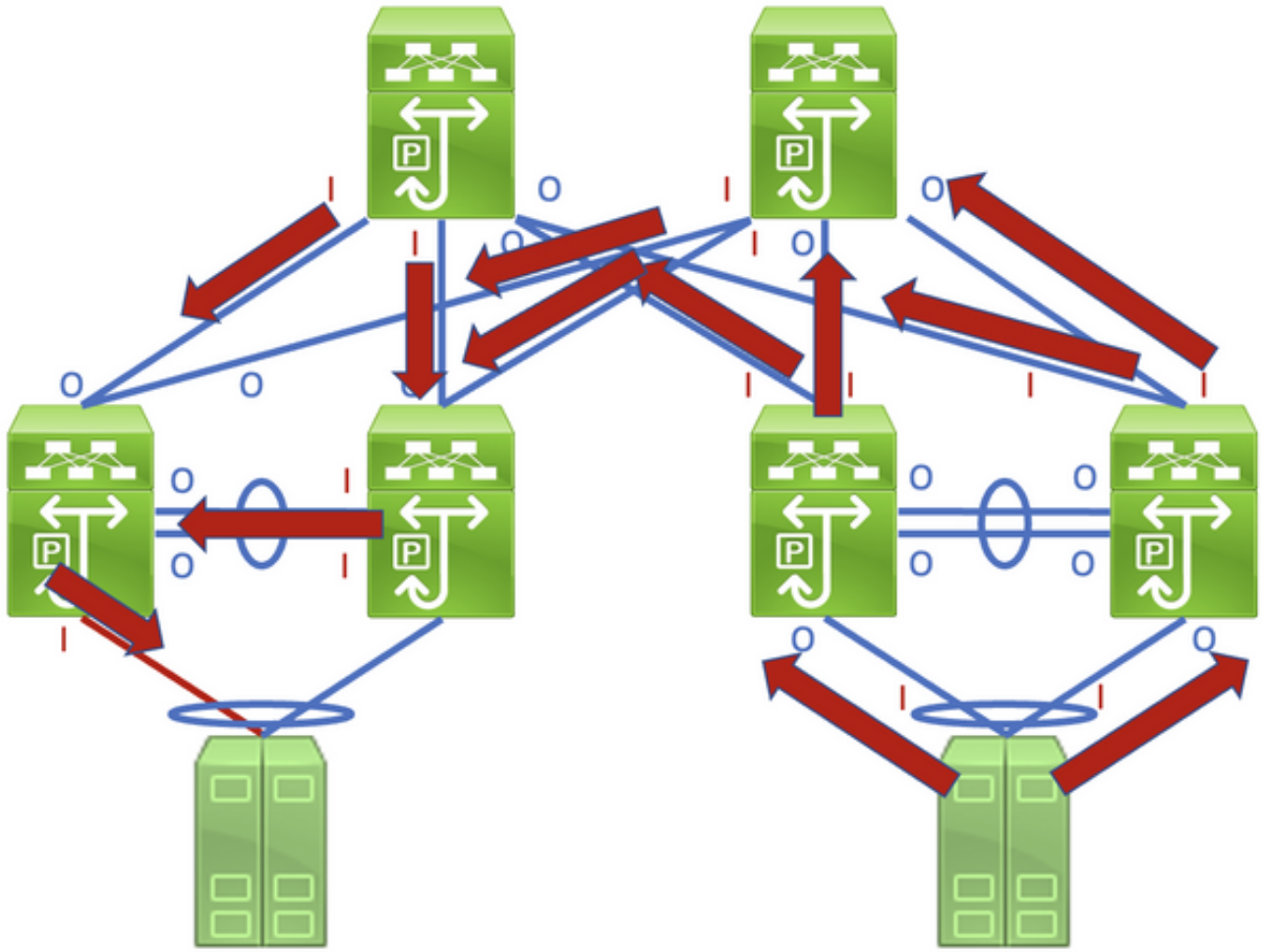
CRC 오류의 근본 원인을 식별하고 해결하기 위한 첫 번째 단계는 CRC 오류의 소스를 네트워크 내의 두 디바이스 간의 특정 링크로 분리하는 것입니다. 이 링크에 연결된 장치 하나에 값이 0이거나 증가하지 않는 인터페이스 출력 오류 카운터가 있는 반면, 이 링크에 연결된 다른 장치에는 0이 아니거나 증가된 인터페이스 입력 오류 카운터가 있습니다. 이는 트래픽이 한 디바이스의 인터페이스를 손상되지 않은 상태로 원격 디바이스로 전송할 때 손상되며, 링크에 있는 다른 디바이스의 인그레스 인터페이스에서 입력 오류로 간주된다는 것을 의미합니다.

스토어 및 전달 포워딩 모드에서 작동하는 네트워크 디바이스로 구성된 네트워크에서 이 링크를 식별하는 것은 간단한 작업입니다. 그러나 컷스루(Cut-Through) 포워딩 모드에서 작동하는 네트워크 디바이스로 구성된 네트워크에서 이 링크를 식별하기가 더 어렵습니다. 많은 네트워크 디바이스에

0이 아닌 입력 및 출력 오류 카운터가 있기 때문입니다. 이 현상의 예는 토폴로지에서 확인할 수 있습니다. 여기서 빨간색으로 강조 표시된 링크가 손상되어 링크를 통과하는 트래픽이 손상되었습니다. 빨간색 "I"로 레이블이 지정된 인터페이스는 0이 아닌 입력 오류가 있을 수 있는 인터페이스를 나타내며, 파란색 "O"로 레이블이 지정된 인터페이스는 0이 아닌 출력 오류가 발생할 수 있는 인터페이스를 나타냅니다.



결함이 있는 링크를 식별하려면 네트워크에서 0이 아닌 입력 및 출력 오류 카운터를 통해 뒤따르는 "경로" 손상된 프레임을 반복적으로 추적해야 하며, 0이 아닌 입력 오류는 네트워크의 손상된 링크를 가리키는 업스트림을 가리킵니다. 이 내용은 여기 다이어그램에 나와 있습니다.



손상된 링크를 추적 및 식별하는 자세한 프로세스는 예를 통해 가장 잘 보여집니다. 토폴로지는 다음과 같습니다.

이 토폴로지에서 Switch-1이라는 Nexus 스위치의 Ethernet1/1은 Host-1이라는 호스트에 Host-1을 통해 Host-1이라는 호스트에 연결됩니다. Switch-1의 Interface Ethernet1/2는 Switch-2의 인터페이스 Ethernet1/2를 통해 Switch-2라는 두 번째 Nexus 스위치에 연결됩니다. Switch-2의 Ethernet1/1 인터페이스는 Host-2를 통해 Host-2라는 호스트에 연결됩니다. eth0.

Host-1과 Switch-1 간의 링크가 Switch-1의 Ethernet1/1 인터페이스에 손상되어 링크를 통과하는 트래픽이 간헐적으로 손상됩니다. 그러나 이 링크가 손상되었는지 아직 알 수 없습니다. 손상된 프레임이 네트워크에 남아 있는 경로를 0이 아닌 것으로 추적하거나 입력 및 출력 오류 카운터를 증가시켜 이 네트워크에서 손상된 링크를 찾아야 합니다.

이 예에서 Host-2의 NIC는 CRC 오류를 수신한다고 보고합니다.

```
Host-2$ ip -s link show eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
   link/ether 00:50:56:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
   32246366102 444908978 478920       647     0       419445867
   TX: bytes  packets  errors  dropped carrier collsns
   3352693923 30185715 0       0       0       0
   altname enp11s0
```

Host-2의 NIC가 인터페이스 Ethernet1/1을 통해 Switch-2에 연결된다는 사실을 알고 있습니다. 인터페이스 Ethernet1/1에 **show interface** 명령과 함께 0이 아닌 출력 오류 카운터가 있는지 확인할 수 있습니다.

```
Switch-2# show interface
<snip>
Ethernet1/1 is up
admin state is up, Dedicated Interface
    RX
      30184570 unicast packets  872 multicast packets  273 broadcast packets
      30185715 input packets  3352693923 bytes
      0 jumbo packets  0 storm suppression bytes
      0 runts  0 giants 0 CRC  0 no buffer
      0 input error  0 short frame  0 overrun   0 underrun  0 ignored
      0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
      0 input with dribble  0 input discard
      0 Rx pause
    TX
      444907944 unicast packets  932 multicast packets  102 broadcast packets
      444908978 output packets  32246366102 bytes
      0 jumbo packets
      478920 output error  0 collision  0 deferred  0 late collision
      0 lost carrier  0 no carrier  0 babble  0 output discard
      0 Tx pause
```

인터페이스 Ethernet1/1의 출력 오류 카운터가 0이 아니므로 0이 아닌 입력 오류 카운터가 있는 Switch-2의 다른 인터페이스가 있을 가능성이 높습니다. **show interface counters errors non-zero** 명령을 사용하여 Switch-2의 인터페이스에 0이 아닌 입력 오류 카운터가 있는지 식별할 수 있습니다.

```
Switch-2# show interface counters errors non-zero
<snip>
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1                0          0     478920          0          0           0
Eth1/2                0     478920          0     478920          0           0

--------------------------------------------------------------------------------
Port         Single-Col  Multi-Col   Late-Col  Exces-Col  Carri-Sen       Runts
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port          Giants SQETest-Err Deferred-Tx IntMacTx-Er IntMacRx-Er Symbol-Err
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port         InDiscards
--------------------------------------------------------------------------------
```

Switch-2의 Ethernet1/2에 0이 아닌 입력 오류 카운터가 있는 것을 확인할 수 있습니다. 이는 Switch-2가 이 인터페이스에서 손상된 트래픽을 수신함을 의미합니다. Cisco CDP(Discovery Protocol) 또는 LLDP(Link Local Discovery Protocol) 기능을 통해 어떤 디바이스가 스위치-2의 Ethernet1/2에 연결되었는지 확인할 수 있습니다. 이 예제는 show cdp neighbors 명령과 함께 여기에 표시됩니다.

```
Switch-2# show cdp neighbors
<snip>
    Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
    S - Switch, H - Host, I - IGMP, r - Repeater,
    V - VoIP-Phone, D - Remotely-Managed-Device,
    s - Supports-STP-Dispute

Device-ID          Local Intrfce  Hldtme Capability  Platform      Port ID
Switch-1(FDO12345678)
                    Eth1/2         125    R S I s   N9K-C93180YC- Eth1/2
```

이제 Switch-2가 Switch-1의 Ethernet1/2 인터페이스에서 Ethernet1/2 인터페이스에서 손상된 트래픽을 수신하지만 Switch-1의 Ethernet1/2와 Switch-2의 Ethernet1/2 간의 링크가 손상되어 손상을 일으키는지 또는 Switch-1이 손상된 트래픽을 수신하는 컷스루 스위치 포워딩 스위치인지 알 수 없습니다. 이를 확인하려면 Switch-1에 로그인해야 합니다.

Switch-1의 Ethernet1/2 인터페이스에 **show interfaces** 명령을 사용하여 0이 아닌 출력 오류 카운터가 있는지 확인할 수 있습니다.

```
Switch-1# show interface
<snip>
Ethernet1/2 is up
admin state is up, Dedicated Interface
    RX
      30581666 unicast packets  178 multicast packets  931 broadcast packets
      30582775 input packets  3352693923 bytes
      0 jumbo packets  0 storm suppression bytes
      0 runts  0 giants 0 CRC  0 no buffer
      0 input error  0 short frame  0 overrun   0 underrun  0 ignored
      0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
      0 input with dribble  0 input discard
      0 Rx pause
    TX
      454301132 unicast packets  734 multicast packets  72 broadcast packets
      454301938 output packets  32246366102 bytes
      0 jumbo packets
      478920 output error  0 collision  0 deferred  0 late collision
      0 lost carrier  0 no carrier  0 babble  0 output discard
      0 Tx pause
```

Switch-1의 Ethernet1/2에 0이 아닌 출력 오류 카운터가 있는 것을 확인할 수 있습니다. 이는 Switch-1의 Ethernet1/2와 Switch-2의 Ethernet1/2 간의 링크가 손상되지 않았음을 의미합니다. 대신 Switch-1은 일부 다른 인터페이스에서 수신하는 손상된 트래픽을 전달하는 컷스루 스위치입니다. 이전에 Switch-2에서 설명한 것처럼, **show interface counters errors non-zero** 명령을 사용하여 Switch-1의 인터페이스에 0이 아닌 입력 오류 카운터가 있는지 확인할 수 있습니다.

```
Switch-1# show interface counters errors non-zero
<snip>
```

```
--------------------------------------------------------------------------------
Port          Align-Err    FCS-Err   Xmit-Err    Rcv-Err  UnderSize OutDiscards
--------------------------------------------------------------------------------
Eth1/1                0     478920          0     478920          0           0
Eth1/2                0          0     478920          0          0           0

--------------------------------------------------------------------------------
Port         Single-Col  Multi-Col   Late-Col  Exces-Col  Carri-Sen       Runts
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port          Giants SQETest-Err Deferred-Tx IntMacTx-Er IntMacRx-Er Symbol-Err
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Port         InDiscards
--------------------------------------------------------------------------------
```

Switch-1의 Ethernet1/1에 0이 아닌 입력 오류 카운터가 있는 것을 확인할 수 있습니다. 이는 Switch-1이 이 인터페이스에서 손상된 트래픽을 수신하고 있음을 나타냅니다. 이 인터페이스는 Host-1의 eth0 NIC에 연결된다는 것을 알고 있습니다. Host-1의 eth0 NIC 인터페이스 통계를 검토하여 Host-1이 이 인터페이스에서 손상된 프레임을 전송하는지 확인할 수 있습니다.

```
Host-1$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group
default qlen 1000
   link/ether 00:50:56:84:8f:6d brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors dropped overrun mcast
   73146816142 423112898 0       0     0       437368817
   TX: bytes  packets  errors  dropped carrier collsns
   3312398924 37942624 0       0       0       0
   altname enp11s0
```

Host-1의 eth0 NIC 통계는 호스트가 손상된 트래픽을 전송하지 않음을 나타냅니다. 이는 Host-1의 eth0과 Switch-1의 Ethernet1/1 간의 링크가 손상되었으며 이 트래픽 손상의 원인임을 나타냅니다. 이러한 손상을 일으키는 결함이 있는 구성 요소를 식별하고 교체하려면 이 링크에서 추가적인 트러블슈팅을 수행해야 합니다.

CRC 오류의 근본 원인

CRC 오류의 가장 일반적인 근본 원인은 두 디바이스 간의 물리적 링크의 손상된 또는 오작동 구성 요소입니다. 예를 들면 다음과 같습니다.

- 물리적 미디어(구리 또는 파이버) 또는 DAC(Direct Attach Cables)에 결함이 있거나 손상되었습니다.
- 장애 또는 손상된 트랜시버/옵틱
- 패치 패널 포트에 장애가 발생했거나 손상되었습니다.
- 네트워크 장치 하드웨어 오류(특정 포트, 라인 카드 ASIC(Application-Specific Integrated Circuits), MAC(Media Access Controls), 패브릭 모듈 등),
- 호스트에 삽입된 네트워크 인터페이스 카드 오류

잘못 구성된 하나 이상의 디바이스가 실수로 네트워크 내 CRC 오류를 일으킬 수도 있습니다. 한 가지 예는 네트워크 내에서 둘 이상의 디바이스 간에 MTU(Maximum Transmission Unit) 컨피그레이션이 일치하지 않아 큰 패킷이 잘못 잘렸습니다. 이 구성 문제를 식별하고 해결하면 네트워크 내에서 CRC 오류를 수정할 수도 있습니다.

CRC 오류 해결

제거 프로세스를 통해 특정 오작동 컴포넌트를 식별할 수 있습니다.

1. 물리적 미디어(구리 또는 파이버) 또는 DAC를 동일한 유형의 정상 작동이 확인된 물리적 미디어로 교체합니다.
2. 한 디바이스의 인터페이스에 삽입된 트렌시버를 동일한 모델의 정상 작동이 확인된 트랜시버로 교체합니다. 이렇게 해도 CRC 오류가 해결되지 않으면 다른 디바이스의 인터페이스에 삽입된 트렌시버를 동일한 모델의 정상 작동이 확인된 트랜시버로 교체합니다.
3. 손상된 링크의 일부로 패치 패널을 사용하는 경우 패치 패널의 정상 작동이 확인된 포트로 링크를 이동합니다. 또는 가능한 경우 패치 패널을 사용하지 않고 링크를 연결하여 잠재적인 근본 원인으로 패치 패널을 제거할 수 있습니다.
4. 손상된 링크를 각 장치의 정상 작동이 확인된 다른 포트로 이동합니다. MAC, ASIC 또는 라인 카드 장애를 격리하려면 서로 다른 여러 포트를 테스트해야 합니다.
5. 손상된 링크에 호스트가 포함된 경우 해당 링크를 호스트의 다른 NIC로 이동합니다. 또는 손상된 링크를 정상 작동이 확인된 호스트에 연결하여 호스트의 NIC 장애를 격리합니다.

오작동 중인 지원 계약이 적용되는 Cisco 제품(예: Cisco 네트워크 디바이스 또는 트랜시버)인 경우 문제 해결을 자세히 설명하는 Cisco TAC에서 지원 케이스를 열어 RMA(Return Material Authorization)를 통해 오작동 구성 요소를 교체할 수 있습니다.

관련 정보

- Nexus 9000 클라우드 확장 ASIC CRC ID 및 추적 절차
- 기술 지원 및 문서 – Cisco Systems