

Catalyst 6500, 4500 및 3750 Series 스위치 EtherChannel 로드 밸런싱

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[Catalyst 6500 Series 스위치](#)

[Catalyst 4500 Series 스위치](#)

[Catalyst 3750 Series 스위치](#)

[잠재적 문제](#)

소개

이 문서에서는 Cisco Catalyst 6500, 4500 및 3750 Series 스위치에서 특정 트래픽 플로우에 사용되는 EtherChannel 멤버 링크를 식별하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

EtherChannel에 대한 기본적인 지식을 보유한 것이 좋습니다.

사용되는 구성 요소

이 문서의 정보는 다음 플랫폼을 기반으로 합니다. Cisco Catalyst 6500, 4500, 2960, 3750, 3750G, 3750X 및 3560 Series 스위치.

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

배경 정보

EtherChannel 로드 밸런싱의 작동 방식은 트래픽 유형에 대해 구성된 해시 방법(로드 밸런싱 알고리즘)에 따라 스위치가 0-7의 해시 결과를 할당하는 것입니다. 이 해시 결과는 일반적으로 RBH(결과 번들 해시)라고 합니다.

플로우 수는 구성된 로드 밸런싱 알고리즘에 따라 달라집니다. 이제 DDoS 공격의 실제 사례를 살펴 보겠습니다.

```
Source 192.168.1.1 (mac a.a.a ) sending a tcp stream to 172.16.1.1 ( mac b.b.b )  
with a source tcp port of 50 and destination port 2000
```

```
Source 192.168.1.1 (mac a.a.a ) sending a tcp stream to 209.165.201.1 ( mac c.c.c )  
with a source tcp port of 60 and destination 2000.
```

```
If configured load balancing algorithm is SRC_MAC  
Then no of flows = 1
```

```
If configured load balancing algorithm is DST_MAC  
Then no of flows = 2
```

```
If configured load balancing algorithm is DST_PORT  
Then no of flows= 1
```

Catalyst 6500 Series 스위치

1. 작동 부하 균형 알고리즘을 확인합니다.
2. SP(스위치 프로세서)에서 **show etherchannel load-balance**를 입력합니다.

```
6500#remote login sw  
Trying Switch ...  
Entering CONSOLE for Switch  
Type "^C^C^C" to end this session
```

```
6500-sp#show etherchannel load-balance  
EtherChannel Load-Balancing Configuration:  
    src-dst-ip  
    mpls label-ip
```

3. 원하는 흐름 사이의 패킷에 대해 선택한 RBH 값을 찾습니다.

```
6500-sp#test etherchannel load-balance interface port-channel
```

이 예에서 플로우는 192.168.1.1~172.16.1.1 사이이고 해당 포트 채널은 port-channel 1입니다. 1단계의 출력을 기반으로 명령의 속성을 선택합니다. 구성된 로드 밸런싱 알고리즘이 src_ip인 경우 패킷 192.168.1.1의 src-ip를 지정합니다. 이 예에서는 src-dst ip가 구성된 로드 밸런싱 알고리즘을 포함하므로 명령은 192.168.1.1~172.16.1.1을 모두 포함해야 합니다.

```
6500-sp#test etherchannel load-balance int port-channel 1 ip 192.168.1.1 172.16.1.1
```

```
Computed RBH: 0x5  
Would select Gi3/2 of Po1
```

4. RBH 값에 매핑된 물리적 포트를 찾습니다.(선택 사항)

Cisco IOS®의 특정 버전에서 명령 출력은 선택된 물리적 인터페이스를 제공하지 않습니다. 2단계에서 이그레스 인터페이스 정보가 생성되지 않은 경우에만 이 단계를 수행합니다.

```
6500-sp#test etherchannel load-balance int port-channel 1 ip 192.168.1.1 172.16.1.1
Computed RBH: 0x5
```

Route Processor 콘솔로 전환하고 **show interface port-channel <num> etherchannel** 명령을 입력합니다. 물리적 인터페이스에 해당하는 Load 열 출력을 확인합니다. Load 값을 이진으로 변환합니다(이 예 참조).

```
6500-sp#exit
```

```
[Connection to Switch closed by foreign host]
```

```
6500#show interface port-channel 1 etherchannel
```

```
Port-channel1 (Primary aggregator)
Age of the Port-channel = 0d:01h:05m:54s
Logical slot/port = 14/1 Number of ports = 2
HotStandBy port = null
Port state = Port-channel Ag-Inuse
Protocol = LACP
Fast-switchover = disabled
```

```
Ports in the Port-channel:
```

```
Index Load Port EC state No of bits
```

```
-----+-----+-----+-----+-----
```

```
 0    55   Gi3/1   Active    4
 1    AA   Gi3/2   Active    4
```

여기서 gi3/2의 로드 값은 AA이고 gi3/1의 로드 값은 55입니다.

```
          7654 3210
gig3/2 - AA - 1010 1010
          ----
          |    |
          A    A
```

```
gig3/1 - 55 - 0101 0101
          ----
          |    |
          5    5
```

For gi3/2 bits 1,3,5 and 7 are set. So RBH value of 1,3,5,and 7 chooses gi3/2.

For gi3/1 bits 0,2,4 and 6 are set. So RBH value of 0,2,4,and 6 chooses gi3/1.

출력은 두 인터페이스 각각에 대해 4비트가 설정되었음을 보여줍니다. 따라서 EtherChannel에 두 개의 링크가 있을 경우 각 링크가 동일한 사용 가능성을 갖습니다.

그러나 EtherChannel에 3개의 링크가 있는 경우 테스트 EtherChannel의 출력은 다음과 유사합니다.

```
6500#show interface port-channel 1 etherchannel
```

```
Port-channel1 (Primary aggregator)
Age of the Port-channel = 0d:01h:05m:54s
Logical slot/port = 14/1 Number of ports = 2
HotStandBy port = null
Port state = Port-channel Ag-Inuse
Protocol = LACP
Fast-switchover = disabled
Ports in the Port-channel:
```

Index	Load	Port	EC state	No of bits
0	49	Gi3/1	Active	3
1	92	Gi3/2	Active	3
2	24	Gi3/3	Active	2

-----+-----+-----+-----+-----

0	49	Gi3/1	Active	3
1	92	Gi3/2	Active	3
2	24	Gi3/3	Active	2

여기서 비트 공유 비율은 3:3:2입니다. 따라서 두 링크는 세 번째 링크와 비교했을 때(마지막에 있는 추가 섹션에서 더 많음) 더 많이 사용될 가능성이 높습니다.

Catalyst 4500 Series 스위치

1. `show etherchannel load-balancing` 명령을 사용하여 구성된 로드 밸런싱 알고리즘을 확인합니다.
2. 이그레스 인터페이스를 찾으려면 `show platform software etherchannel port-channel 1 map` 명령을 사용합니다.

```
4500#show platform software etherchannel port-channel 1 map ip 192.168.1.1
172.16.1.1
```

```
Map port for IP 192.168.1.1, 172.16.1.1 is Gi3/1(Po1)
NOTE: Software forwarded traffic uses Gi3/1(Po1)
```

Catalyst 3750 Series 스위치

Catalyst 3750 Series 스위치에서는 비슷한 8비트 해싱 알고리즘이 사용됩니다. 여기에서 EtherChannel의 링크 수가 2, 4 또는 8인 경우에도 트래픽 분배가 더 많습니다. 포트 채널에서 인터페이스를 확인하는 명령입니다.

```
test etherchannel load-balance interface port-channel
```

포트 채널 로드 밸런싱 알고리즘이 3750에서 `src-dst ip`(기본값 아님)로 구성되어 있다고 가정합니다. 이 예에서는 트래픽에서 172.16.1.1으로 이동하는 트래픽에 사용되는 링크를 식별하는 방법을 보여줍니다.

```
3750(config)#port-channel load-balance src-dst-ip
```

```
3750#show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
src-dst-ip
```

```
3750#test etherchannel load-balance interface port-channel 1 ip 192.168.1.1 172.16.1.1
Would select Gi3/1 of Po1
```

로드 밸런싱 알고리즘이 MAC 주소 기반인 경우 패킷의 소스 및 목적지 MAC 주소를 제공할 경우 이전 명령을 사용할 수 있습니다.

잠재적 문제

다음은 EtherChannel 멤버 인터페이스의 트래픽 분포를 다르게 일으킬 수 있는 몇 가지 시나리오와 로드 밸런싱을 최적화하기 위해 수행해야 하는 단계입니다.

- **시나리오:** EtherChannel에 두 개의 플로우와 두 개의 물리적 인터페이스가 있는 경우 한 플로우가 다른 플로우보다 더 수다적일 수 있습니다. 5개의 플로우가 있고 1개는 가장 수다적이면 이 흐름은 다른 흐름을 압도할 수 있습니다. 이 플로우가 선택하는 물리적 인터페이스 중 어느 것이 다른 인터페이스보다 사용률이 상대적으로 높습니다.
- **해결 방법:** Super Talker의 흐름을 제어합니다. 호스트 측면에서 봐야 합니다.
-
- **시나리오:** 일반적인 문제는 충분한 플로우가 없고 대부분의 적은 수의 플로우가 동일한 물리적 인터페이스에 해시된다는 것입니다.
- **해결 방법:** 플로우 수를 늘립니다. 해시 알고리즘을 트래픽에 가장 적합한 것으로 변경하려고 합니다.
-
- **시나리오:** EtherChannel에 물리적 링크가 3개, 5개, 6개 또는 7개 있을 경우, 일부 링크는 다른 링크보다 트래픽을 가져올 가능성이 높습니다(각 물리적 인터페이스에 할당된 해시 비트 수를 기준으로 함). 그러면 트래픽이 균등하게 분산될 가능성이 있습니다.
- **해결 방법:** EtherChannel에서 링크 2, 4 또는 8개를 사용합니다.