Catalyst 6500 S2T CEF 엔트리 이해

목차

소개

이 문서에서는 Cisco Catalyst 6500 with Supervisor Sup2T가 Cisco IOS 소프트웨어에 구성된 (Cisco Express Forwarding) CEF 항목을 패킷 포워딩을 수행하는 데 사용되는 라인 카드 하드웨어에 프로그래밍하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- CEF(Cisco Express Forwarding)
- Cisco Catalyst 6500 Series Switches
- Cisco DFC(Distributed Forwarding Card)

사용되는 구성 요소

이 문서의 정보는 다음 하드웨어 및 소프트웨어 버전을 기반으로 합니다.

- Cisco Catalyst 6500 WS-X6848-GE-TX(DFC4 포함) 라인 카드
- Cisco Catalyst 6500(IOS 버전 15.2.1SY5의 Supervisor 2T 포함)

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.

배경 정보

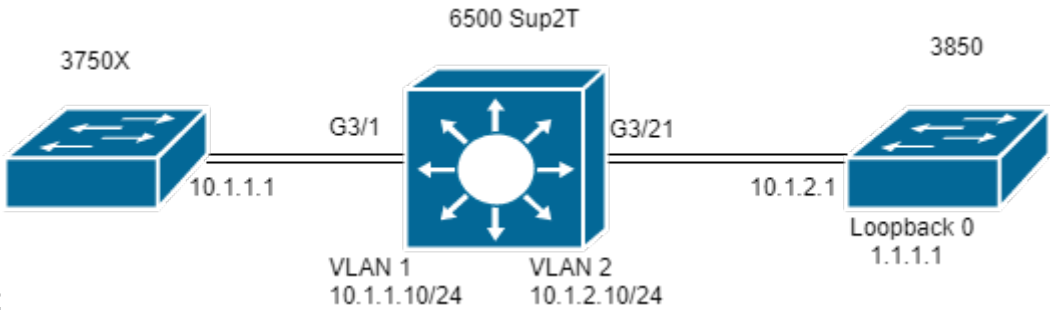
레이어 3 스위칭 메커니즘으로서 CEF는 대부분의 Cisco 멀티레이어 스위치에서 사용됩니다. 네트워크 엔지니어는 네트워크 중단, 패킷 손실 또는 패킷 지연 시나리오를 일상적으로 해결하기 위해 CEF가 작동하는 방식을 이해하는 것이 중요합니다.

Sup2T 수퍼바이저는 독립형 모드 또는 VSS로 현재 많은 엔터프라이즈 네트워크에서 코어 스위치로 구축되어 있으며, 사실상 다른 모든 라우팅 또는 스위칭 디바이스를 통합합니다. 이는 패킷을 대상에 성공적으로 전달하기 위해 대부분의 도메인 내 및 도메인 간 트래픽을 전달한다는 의미이기도 합니다. 이를 위해서는 Sup2T가 라우팅 프로토콜을 통해 정적으로 또는 동적으로 학습된 적절한 라우팅 정보가 있어야 합니다.

모듈형 섀시에는 수퍼바이저 외에 여러 포워딩 엔진이 있을 수 있습니다. 특정 라인 카드(특히 C6800-32P10G와 같은 신세대 라인 카드)는 패킷 스위칭 성능을 향상시키기 위해 자체 포워딩 엔진을 이미 포함하고 있으며, CEF 항목의 조회는 로컬로 실행되며 다른 라인 카드를 통해 들어오는 트래픽에 리소스를 가장 잘 분배합니다. 이를 DFC(Distributed Forwarding Card)라고 합니다.

모든 포워딩 엔진에서 공유되는 이러한 CEF 엔트리는 소프트웨어 결함 상태에서 높은 CPU 조건으로 리소스 소진 등 여러 가지 이유로 HW에서 할당되지 못할 수 있으며 스위치가 모든 엔트리를 업데이트하기에 충분한 시간을 확보하지 못하게 되므로 일련의 바람직하지 않은 이벤트가 발생할 수 있습니다.

네트워크 다이어그램



네트워크:

```
Switch#show module 3
---------------------- -----------------------------
Mod Ports Card Type Model Serial No.
--- ----- ------------------------------------ ------------------ -----------
3 48 CEF720 48 port 10/100/1000mb Ethernet WS-X6848-GE-TX SAL2003X5AH
---- -------------------------- ------------------ ----------- ------- -------
3 Distributed Forwarding Card WS-F6K-DFC4-A SAL2003X5AH 1.4 Ok
```

분산된 포워딩 엔진을 통해 CEF 항목 식별

다이어그램에서 독립형 6506 스위치에는 Supervisor 2T가 설치되어 있고 슬롯 3에 DFC가 포함된 라인 카드 WS-6848-GE-TX가 있습니다. 포트 G3/1을 통해 라인 카드에 연결된 호스트 3750X는 3850의 루프백 0 주소 1.1.1.1로 트래픽을 전송합니다.

이를 위해 3750X에는 다음 홉 10.1.1.1.10을 통해 IP 주소 1.1.1.1로 이동하는 고정 경로가 있습니다. 이는 Sup2T 스위치의 VLAN 1에 대한 SVI입니다. Sup2T 스위치는 VLAN 2의 Sup2T에 연결된 3850 인터페이스인 다음 홉 10.1.2.1을 통해 IP 1.1.1.1/32에 대한 고정 경로 항목에 따라 이 트래픽을 3850 스위치로 라우팅해야 합니다.

```
MXC.CALO.3750X#show ip route | inc 1.1.1.1
S 1.1.1.1 [1/0] via 10.1.1.10
```

```
MXC.CALO.Sup2T#show ip route | inc 1.1.1.1
S 1.1.1.1 [1/0] via 10.1.2.1
```

```
CALO.MXC.3850#show ip route | inc 1.1.1.1
C 1.1.1.1 is directly connected, Loopback1
```

간소화를 위해 3750X 및 3850 스위치 모두 동일한 라인 카드를 통해 6500에 연결됩니다. 즉, 트래픽은 로컬에서 조회되고 로컬에서도 전달됩니다.

패킷은 Gi3/1을 통해 Sup2T 스위치를 인그레스(ingress)하므로(DFC이므로) 결국 포워딩 엔진에 도달합니다. 포워딩 엔진은 이 패킷의 목적지 IP 주소 필드 및 최상의 일치(최장 마스크)를 위해 프로그래밍된 CEF 항목에 대한 조회를 구문 분석합니다.

이 카드는 DFC 카드이므로 자체 CEF 항목이 있으며 이를 확인하기 위해 VSS를 위해 명령 연결 [dec] 또는 스위치 [1-2] 모드 [dec]을 사용하여 라인 카드에 연결해야 합니다.

이제 DFC 프롬프트에 있어야 합니다. 명령 show platform hardware cef 또는 show platform hardware cef vpn 0은 일반 라우팅 테이블에 대해 프로그래밍된 모든 CEF 항목을 반환합니다(VPN 0/ No VRF).

목표는 접두사 1.1.1.1/32이므로 명령 show platform hardware cef vpn 0 lookup 1.1.1.1을 사용합니다. 이 명령은 접두사 1.1.1.1에 가장 일치하는 접두사와 실제 트래픽을 전달하는 데 사용하는 접두사를 반환합니다.

<#root>

```
MXC.CALO.Sup2T#attach 3
Trying Switch ...
Entering CONSOLE for Switch
Type "^C^C^C" to end this session

MXC.CALO.Sup2T-dfc3#
```

show platform hardware cef vpn 0

```
Codes: decap - Decapsulation, + - Push Label
Index Prefix Adjacency
32 0.0.0.0/32 receive
33 255.255.255.255/32 receive
34 10.1.85.254/32 glean
35 10.1.85.5/32 receive
36 10.1.86.5/32 receive
[snip...]
```

```
MXC.CALO.Sup2T-dfc3#
```

show platform hardware cef vpn 0 lookup 1.1.1.1

```
Codes: decap - Decapsulation, + - Push Label
Index Prefix Adjacency
262 1.1.1.1/32 Vl2 ,0c11.678b.f6f7
```

CEF 엔트리는 IOS 소프트웨어에서 명령 ip route 1.1.1.1 255.255.255.255 10.1.2.1을 통해 프로그래밍된 정적 엔트리의 결과로 프로그래밍되었습니다.

또한 이 항목이 적중률을 얻고 트래픽이 인접 항목을 반환하는 명령 show platform hardware cef 1.1.1.1 detail을 통해 이 항목과 함께 전달되는지 확인할 수 있습니다.

<#root>

```
MXC.CALO.Sup2T-dfc3#
```

show platform hardware cef 1.1.1.1 detail

```
Codes: M - mask entry, V - value entry, A - adjacency index, NR- no_route bit
LS - load sharing count, RI - router_ip bit, DF: default bit
CP - copy_to_cpu bit, AS: dest_AS_number, DGTv - dgt_valid bit
DGT: dgt/others value

Format:IPV4 (valid class vpn prefix)
M(262 ): 1 F 2FFF 255.255.255.255
V(262 ): 1 0 0 1.1.1.1
(A:114689, LS:0, NR:0, RI:0, DF:0 CP:0 DGTv:1, DGT:0)
```

마지막으로, 인접성 항목은 패킷이 재작성되는 방법과 이 인접성 항목에 의해 트래픽이 재작성되는 경우를 보여줍니다.

<#root>

```
MXC.CALO.Sup2T-dfc3#
```

show platform hardware cef adjacencies entry 114689 detail

```
RIT fields: The entry has a Layer2 Format
_____________________________________________________
|decr_ttl = YES | pipe_ttl = 0 | utos = 0
|_____________________|_____________________|_____________________
|l2_fwd = 0 | rmac = 0 | ccc = L3_REWRITE
|_____________________|_____________________|_____________________
|rm_null_lbl = YES| rm_last_lbl = YES| pv = 0
|_____________________|_____________________|_____________________
|add_shim_hdr= NO | rec_findex = N/A | rec_shim_op = N/A
|_____________________|_____________________|_____________________
|rec_dti_type = N/A | rec_data = N/A
|_____________________________________|_____________________
|modify_smac = YES| modify_dmac = YES| egress_mcast = NO
```

```
|_____________________________________|____________________
|ip_to_mac = NO
|_____________________________________________________________
|dest_mac = 0c11.678b.f6f7 | src_mac = d8b1.902c.9680
|_____________________________|_____________________________
|
Statistics: Packets = 642
Bytes = 75756                    <<<<
```

dest_mac 및 src_mac은 이 패킷에 대해 작성된 새 L2 헤더를 나타내는 주요 관심 값입니다. 대상 MAC 주소 0c11.678b.f6f7은 3850인 10.1.2.1입니다(1.1.1.1에 도달하는 다음 홉).

```
MXC.CALO.Sup2T#show ip arp 10.1.2.1
Protocol Address Age (min) Hardware Addr Type Interface
Internet 10.1.2.1 30 0c11.678b.f6f7 ARPA Vlan2
```

또한 Statistics 필드는 트래픽이 실제로 이 인접성 항목에 도달하고 그에 따라 L2 헤더가 다시 작성됨을 보여 줍니다.

CEF 항목 삭제

CEF 항목을 삭제하면 잘못 프로그래밍되었거나(예: 잘못된 인접성 항목에) 훈련 목적으로도 잘못된 항목을 삭제할 수 있습니다. 또한 라우팅 경로를 수정하는 방법도 제공합니다.

CEF 항목을 삭제하려면 CEF 항목이 순차적으로 프로그래밍되고 다음과 같이 하드웨어 인덱스가 할당되었음을 이해해야 합니다.

MXC.CALO.Sup2T-dfc3#show platform hardware cef vpn 0

코드: decapp - 역캡슐화, + - 푸시 레이블

<#root>

```
MXC.CALO.Sup2T-dfc3#show platform hardware cef vpn 0
...
Index Prefix Adjacency
259 10.1.2.255/32 receive
260 10.1.1.1/32 Vl1 ,a0ec.f930.3f40
261 10.1.2.1/32 Vl2 ,0c11.678b.f6f7
262
```

1.1.1.1/32 Vl2 ,0c11.678b.f6f7

```
    <<<< Our CEF entry of interest has a HW index of 262.
...
```

이 하드웨어 인덱스는 참조로 사용되므로 CEF 항목을 삭제하는 데 가장 중요한 요소입니다. 그러나 이를 변경하려면 소프트웨어 핸들로 변환해야 합니다. 명령 테스트 플랫폼 하드웨어 cef index-conv hw_to_sw [hw index]

<#root>

```
MXC.CALO.Sup2T-dfc3#

test platform hardware cef index-conv hw_to_sw 262

hw index: 262

---->

 sw handle: 101
```

이제 소프트웨어 핸들을 알고 있으므로 명령 테스트 플랫폼 하드웨어 cef v4-delete [sw handle] mask [mask length] vpn [dec]을 사용하여 CEF 항목 삭제를 진행할 수 있습니다

<#root>

```
MXC.CALO.s2TVSS-sw2-dfc3#

test platform hardware cef v4-delete 101 mask 32 vpn 0

test_ipv4_delete: done.
```

참고: 호스트 특정 경로이므로 마스크 길이 값은 32입니다(1.1.1.1/32).

이제 CEF 항목이 삭제됩니다.

<#root>

```
MXC.CALO.Sup2T-dfc3#

show platform hard cef vpn 0 1.1.1.1

Codes: decap - Decapsulation, + - Push Label
Index Prefix Adjacency


MXC.CALO.Sup2T-dfc3#

show platform hard cef vpn 0

[snip...]
259 10.1.2.255/32 receive
260 10.1.1.1/32 Vl1 ,a0ec.f930.3f40

261 10.1.2.1/32 Vl2 ,0c11.678b.f6f7
288 224.0.0.0/24 receive              <<<<<<< Index 262 no longer exists in the CEF entries.

289 10.1.85.0/24 glean
```

테스트 플랫폼 하드웨어 cef vpn 0 명령이 DFC 프롬프트에서 실행되었음을 확인합니다. 이렇게 하면 CEF 항목이 DFC의 CEF 테이블에서 제거되고 수퍼바이저가 아니라, 어떤 포워딩 엔진에서 항목이 제거되는지 주의해야 합니다.

트래픽의 변경은 가시성이 없는 위험이 있습니다(랩 테스트의 경우). 이는 다른 CEF 항목의 히트 때문일 수 있습니다. 가장 정확한 마스크와 항상 일치해야 합니다(가장 긴 마스크). 이 Lab에서는 다음 사항을 다룹니다.

<#root>

```
MXC.CALO.Sup2T-dfc3#

show plat hard cef vpn 0 lookup 1.1.1.1

Codes: decap - Decapsulation, + - Push Label
Index Prefix Adjacency
262048 0.0.0.0/0 glean
```

그러면 이 항목은 패킷과 실제로 어떤 관련이 있습니까?

```
MXC.CALO.Sup2T-dfc3#show platform hardware cef adjacencies entry 262048
RIT fields: The entry has a Recirc. Format
_____________________________________________________
|decr_ttl=NO | l2_fwd=NO | ccc = 6 | add_shim_hdr = YES
|____________|___________|_________|___________________
|rc_fidx=0 | rc_shimop=1 | rc_dti_type=4 | rc_data = 0x10B
|____________|_____________|_______________|_____________

Statistics: Packets = 2163
Bytes = 255234
```

<#root>

```
Taken from a CPU packet capture using Catlayst 6500 NETDR tool. For NETDR capture tool details refer to

------- dump of incoming inband packet -------

l2idb Po1, l3idb Vl1, routine inband_process_rx_packet, timestamp 01:00:17.841
dbus info: src_vlan 0x1(1), src_indx 0xB40(2880), len 0x82(130)
bpdu 0, index_dir 0, flood 0, dont_lrn 0, dest_indx 0x5FA4(24484), CoS 0
cap1 0, cap2 0
78020800 00018400 0B400100 82000000 1E000464 2E000004 00000010 5FA45BDD
destmac D8.B1.90.2C.96.80, srcmac A0.EC.F9.30.3F.40, shim ethertype CCF0
earl 8 shim header IS present:
version 0, control 64(0x40), lif 1(0x1), mark_enable 1,
feature_index 0, group_id 0(0x0), acos 0(0x0),
ttl 14, dti 4, dti_value 267(0x10B)
10000028 00038080 010B
ethertype 0800
protocol ip: version 0x04, hlen 0x05, tos 0x00, totlen 100, identifier 51573
df 0, mf 0, fo 0, ttl 255, src 10.1.1.1, dst 1.1.1.1
icmp type 8, code 0


------- dump of outgoing inband packet -------
```

```
l2idb NULL, l3idb Vl2, routine etsec_tx_pak, timestamp 01:03:56.989
dbus info: src_vlan 0x2(2), src_indx 0x380(896), len 0x82(130)
bpdu 0, index_dir 0, flood 0, dont_lrn 0, dest_indx 0x0(0), CoS 0
cap1 0, cap2 0
00020000 0002A800 03800000 82000000 00000000 00000000 00000000 00000000
destmac 0C.11.67.8B.F6.F7, srcmac D8.B1.90.2C.96.80, shim ethertype CCF0
earl 8 shim header IS present:
version 0, control 0(0x0), lif 16391(0x4007), mark_enable 0,
feature_index 0, group_id 0(0x0), acos 0(0x0),
ttl 15, dti 0, dti_value 540674(0x84002)
000800E0 0003C008 4002
ethertype 0800
protocol ip: version 0x04, hlen 0x05, tos 0x00, totlen 100, identifier 50407
df 0, mf 0, fo 0, ttl 254, src 10.1.1.1, dst 1.1.1.1
icmp type 8, code 0
```

이제 대상 1.1.1.1에서 라인 카드 3을 통해 들어오는 모든 트래픽은 shim 헤더로 재순환되어 CPU로 푸시됩니다. 때때로 이 CEF 항목 대신 드롭 인접성이 있는 다른 0.0.0.0/0가 표시되고 동일한 작업을 수행합니다.

참고: 어떤 CEF 항목이 제거되었는지 평가합니다. 이로 인해 CPU 사용률이 높아질 수 있습니다. 일반적으로 기본 경로 0.0.0.0/0이 구성되고 이에 따라 트래픽이 전달되며 패킷이 손실됩니다.

CEF 항목 추가

CEF 항목이 추가되면 대부분의 경우 패킷 손실, 패킷 지연 또는 높은 CPU 사용률을 유발하는 모든 잘못된 프로그래밍 문제를 해결합니다. 하드웨어에 CEF 항목을 설치하는 방법에 대한 지식은 잘못 프로그래밍된 항목을 수정할 뿐만 아니라, 패킷의 재순환을 통해 패킷 포워딩을 조작하고, 완전히 다른 인터페이스 또는 다음 홉으로 가리키고, 라우팅된 패킷을 원하는 대로 재작성하거나 삭제하는 등의 기능을 제공합니다. 이 모든 작업은 장비를 다시 로드할 필요 없이 컨피그레이션이나 명백한 변경을 제거하고 설정합니다. CEF 항목 추가는 컨피그레이션 모드에 들어가지 않고도 수행할 수 있습니다. (이전 섹션에서 설명한 CEF 항목 제거 절차에서도 그랬듯이)

기본적으로 다음 홉에 대한 유효한 ARP 항목이 있는 경우(이 경우 10.1.2.1)와 없는 경우(어떤 이유로든)에는 두 가지 상황이 있습니다. 두 번째 상황에서는 고정 ARP를 통해 유효한 ARP 항목을 실제로 생성해야 합니다.

1단계. 스위치에 1.1.1.1의 다음 홉인 10.1.2.1에 대한 ARP 항목이 있습니다.

```
<#root>

MXC.CALO.Sup2T#

show ip arp 10.1.2.1

Protocol Address Age (min) Hardware Addr Type Interface
Internet 10.1.2.1 2 0c11.678b.f6f7 ARPA Vlan2

MXC.CALO.Sup2T#show ip route | inc 1.1.1.1
S 1.1.1.1 [1/0] via 10.1.2.1
```

ARP 항목은 CEF 테이블에서 호스트 경로(/32)로 프로그래밍됩니다.

<#root>

```
MXC.CALO.Sup2T-dfc3#

show plat hard cef vpn 0 look 10.1.2.1

Codes: decap - Decapsulation, + - Push Label
Index Prefix Adjacency
53 10.1.2.1/32 Vl2 ,0c11.678b.f6f7

And of course, there is an index for this which again will tell us how a packet should be rewritten to r


MXC.CALO.Sup2T-sw2-dfc3#

show plat hard cef vpn 0 10.1.2.1 detail
[snip...]

Format:IPV4 (valid class vpn prefix)
M(53 ): 1 F 2FFF 255.255.255.255
V(53 ): 1 0 0 10.1.2.1
(A:114689, LS:0, NR:0, RI:0, DF:0 CP:0 DGTv:1, DGT:0)


Wait, wasn't 114689 adj entry the same used for 1.1.1.1?:


MXC.CALO.Sup2T-sw2-dfc3#show plat hard cef 1.1.1.1 de
[snip...]
Format:IPV4 (valid class vpn prefix)
M(54 ): 1 F 2FFF 255.255.255.255
V(54 ): 1 0 0 1.1.1.1
(A:114689, LS:0, NR:0, RI:0, DF:0 CP:0 DGTv:1, DGT:0)
```

데이터 링크 next hop이 동일한 목적지 IP 주소를 가진 패킷은 동일한 인터페이스를 통해 전달되어야 하며, 동일한 L2 헤더로 재작성되어야 합니다.

이 항목은 처음에 매우 명백해 보일 수 있지만 실제로 CEF 항목을 추가하는 가장 중요한 요소이며, 특정 CEF 인접성 항목을 사용하여 패킷을 재작성하는 방법을 알려줘야 합니다.

2단계. 이제 이에 대해 자동으로 생성된 ARP 항목이 없으므로 고정 ARP 항목을 생성해야 한다고 가정합니다.

이렇게 하려면 접두사 10.1.2.1에 대한 next-hop으로 사용되는 장치의 MAC 주소를 알아야 하므로 0c11.678b.f6f7로 전송됩니다. show mac address-table address 0c11.678b.f6f7 명령 출력에 이미 MAC 주소 항목이 있는 경우, 그렇지 않은 경우 고정 MAC 항목을 만들어야 합니다.

<#root>

```
MXC.CALO.Sup2T(config)#

mac address-table static 0c11.678b.f6f7 vlan 2 int Gi3/21

Displaying entries from DFC switch [2] linecard [3]:

vlan mac address type learn age ports
```

```
----+----+--------------+-------+-----+----------+----------------------------
2 0c11.678b.f6f7 static No - Gi3/21
```

3단계. 마지막으로, CEF 항목을 프로그래밍하려면 고정 ARP 항목을 만들어야 합니다.

<#root>

```
MXC.CALO.Sup2T(config)#

arp 10.1.2.1 0c11.678b.f6f7 arpa

    <<< Static ARP configuration

MXC.CALO.Sup2T#

show ip arp 10.1.2.1

Protocol Address Age (min) Hardware Addr Type Interface
Internet 10.1.2.1 - 0c11.678b.f6f7 ARPA                  <<< Now the static ARP entry is complete

// Attaching to DFC3...

MXC.CALO.Sup2T-sw2-dfc3#

show plat hard cef 10.1.2.1 detail

[snip...]
Format:IPV4 (valid class vpn prefix)
M(53 ): 1 F 2FFF 255.255.255.255
V(53 ): 1 0 0 10.1.2.1
(A:114689, LS:0, NR:0, RI:0, DF:0 CP:0 DGTv:1, DGT:0)

The ARP entry exist in CEF table for DFC3. Same Adjacency Index result as before...
```

이제 이러한 인접성 항목의 기능을 이해했으므로 CEF 항목을 추가할 수 있습니다. 마지막 섹션에서는 테스트 플랫폼 하드웨어 cef v4-delete 명령을 통해 접두사 1.1.1.1/32에 대한 CEF 항목을 삭제했습니다. 이제 명령 테스트 플랫폼 하드웨어 cef v4-insert [prefix] [mask length] vpn [vpn number] adjacency [adjacency index]를 통해 다시 추가합니다.

이를 확인하려면 명령 테스트 플랫폼 하드웨어 cef v4-insert 1.1.1.1 32 vpn 0 adjacency 114689를 사용합니다. 항목이 DFC CEF 테이블에 다시 추가되었습니다.

<#root>

```
MXC.CALO.Sup2T-sw2-dfc3#

test platform hardware cef v4-insert 1.1.1.1 32 vpn 0 adjacency 114689

test_ipv4_insert: done: sw_index = 42

MXC.CALO.Sup2T-sw2-dfc3#

show plat hard cef vpn 0 1.1.1.1

Codes: decap - Decapsulation, + - Push Label
```

```
Index Prefix Adjacency
54 1.1.1.1/32 Vl2 ,0c11.678b.f6f7

Ping from the 3750X to Loopback 0 is successful and HW forwarded by 6500 DFC.

MXC.CALO.Sup2T-sw2-dfc3#show platform hard cef adj entry 114689

Index: 114689 -- Valid entry (valid = 1) --

RIT fields: The entry has a Layer2 Format
_____________________________________________________
|decr_ttl=YES | l2_fwd=NO | ccc = 4 | add_shim_hdr = NO
|_____________|_____________|_________|_____________________

Statistics: Packets = 684
Bytes = 80712


// Logs in 3850

CALO.MXC.385024XU#show logging
[snip...]
*Jan 23 05:59:56.911: ICMP: echo reply sent, src 1.1.1.1, dst 10.1.1.1, topology BASE, dscp 0 topoid 0
*Jan 23 05:59:57.378: ICMP: echo reply sent, src 1.1.1.1, dst 10.1.1.1, topology BASE, dscp 0 topoid 0
*Jan 23 05:59:57.390: ICMP: echo reply sent, src 1.1.1.1, dst 10.1.1.1, topology BASE, dscp 0 topoid 0
```

VRF 라우팅 테이블에 대한 항목 추가 및 삭제

모든 이전 단계에서 구성한 컨피그레이션을 통해 show platform hardware cef 명령의 vpn 0 문자열이 적용되었습니다. 이 명령은 기본적으로 일반 라우팅 테이블 또는 vpn 0에 대한 항목을 반환하므로 완전히 필요하지 않은 것 같더라도, 이 작업은 CEF 항목 1.1.1.1/32을 추가하고 삭제한 문서를 통해 항상 특정 라우팅 테이블 인스턴스(VRF)에서 항목이 추가되거나 삭제된다는 점을 염두에 두기 위해 의도적으로 수행되었습니다. 그러나 특정 접두사는 다른 VRF에 존재할 가능성이 매우 높습니다(i. e. 10.x.x.x) 그리고 잘못된 VRF에 대한 CEF 항목을 삭제, 추가 또는 수정하면 부정적인 영향을 미칠 수 있습니다.

VRF TEST_VRF에 대해 접두사 1.1.1.1/32이 있는 CEF 항목을 삭제합니다. CEF 항목 추가에 대한 자세한 내용은 이 문서의 Add a CEF Entry(CEF 항목 추가) 섹션을 참조하십시오.

VRF를 추가하려면 6500 스위치의 SVI를 명령 ip vrf forwarding [VRF-NAME]을 사용하여 제안된 VRF로 변경하고 마지막으로 TEST_VRF 테이블에 동일한 고정 경로를 추가합니다.

<#root>

```
MXC.CALO.Sup2T(config)#ip vrf TEST_VRF
MXC.CALO.Sup2T(config-vrf)#int vlan 1
MXC.CALO.Sup2T(config-if)#ip vrf forwarding TEST_VRF
% Interface Vlan1 IPv4 disabled and address(es) removed due to enabling VRF TEST_VRF
MXC.CALO.Sup2T(config-if)#ip add 10.1.1.10 255.255.255.0
MXC.CALO.Sup2T(config-if)#int vlan 2
MXC.CALO.Sup2T(config-if)#ip vrf forwarding TEST_VRF
% Interface Vlan2 IPv4 disabled and address(es) removed due to enabling VRF TEST_VRF
MXC.CALO.Sup2T(config-if)#ip add 10.1.2.10 255.255.255.0
MXC.CALO.Sup2T(config)#ip route vrf TEST_VRF 1.1.1.1 255.255.255.255 10.1.2.1
```

```
MXC.CALO.Sup2T#
```

show ip vrf

```
 Name Default RD Interfaces
```

TEST_VRF

```
 <not set> Vl1
 Vl2
```

VRF도 순차적으로 프로그래밍됩니다. 이 VRF는 스위치의 첫 번째 VRF이므로(다른 VRF는 이전에 구성되지 않음) 이 VRF 인스턴스의 vpn 번호는 1이어야 합니다. show platform hardware cef vpn 1 명령을 실행하여 이 값이 올바른지 확인하십시오.

<#root>

```
MXC.CALO.Sup2T-sw2-dfc3#
```

show plat hard cef vpn 1

```
Codes: decap - Decapsulation, + - Push Label
Index Prefix Adjacency
34 10.1.1.10/32 receive
35 10.1.1.0/32 receive
36 10.1.1.255/32 receive
38 10.1.2.10/32 receive
43 10.1.2.0/32 receive
44 10.1.2.255/32 receive
53 10.1.2.1/32 Vl2 ,0c11.678b.f6f7
```

54 1.1.1.1/32 Vl2 ,0c11.678b.f6f7

```
[snip...]
```

**However, usually, switches have hundred or thousands of VRFs and just count them in the 'show ip vrf' co
it will return the actual vpn number for that VRF:**

```
Format:IPV4 (valid class vpn prefix)
M(54 ): 1  F 2FFF 255.255.255.255
V(54 ): 1  0
```

1

```
    1.1.1.1                                <<<<<<<<<<< The number in red determines the VPN this prefix
(A:114689, LS:0, NR:0, RI:0, DF:0 CP:0 DGTv:1, DGT:0)
```

이 항목의 실제 VPN 번호 및 소프트웨어 인덱스를 알아야 이 VRF 인스턴스를 삭제하거나 이 VRF 인스턴스에 추가할 수 있습니다.

<#root>

MXC.CALO.Sup2T-sw2-dfc3#

test platform hardware cef index-conv hw_to_sw 54

hw index: 54 ----> sw handle: 42

MXC.CALO.Sup2T-sw2-dfc3#

test platform hardware cef v4-delete 42 mask 32 vpn 1

test_ipv4_delete: done.

Result:

MXC.CALO.Sup2T-sw2-dfc3#

show platform hardware cef vpn 1 lookup 1.1.1.1

Codes: decap - Decapsulation, + - Push Label
Index Prefix Adjacency

262049 0.0.0.0/0 drop

Traffic is now getting punted, and the effects are seen in the 3750X pings to 1.1.1.1:

MXC.CALO.3750X#ping 1.1.1.1 repe 5000000

```
Sending 5000000, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!!!!!
[snip...]
```

// Packet loss

프로덕션 네트워크에서는 이러한 CEF 엔트리 조건으로 인해 패킷 손실 및 오디오 또는 비디오 품질이 저하된다고 가정합니다. 따라서 유지 관리 창에서 이러한 테스트를 수행하는 것이 좋습니다.

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.