

# 了解Catalyst 6500 S2T CEF条目

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[网络图](#)

[识别分布式转发引擎上的CEF条目](#)

[删除CEF条目](#)

[添加CEF条目](#)

[添加和删除VRF路由表的条目](#)

## 简介

本文档介绍配备管理引擎Sup2T的Cisco Catalyst 6500如何在用于实现数据包转发的线卡硬件中对Cisco IOS软件上配置的 ( 思科快速转发 ) CEF条目进行编程。

## 先决条件

## 要求

Cisco 建议您了解以下主题：

- Cisco 快速转发 (CEF)
- Cisco Catalyst 6500 系列交换机
- 思科分布式转发卡(DFC)

## 使用的组件

本文档中的信息基于下列硬件和软件版本：

- Cisco Catalyst 6500 WS-X6848-GE-TX ( 带DFC4 ) 线卡。
- IOS版本15.2.1SY5上带Supervisor 2T的Cisco Catalyst 6500

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始 ( 默认 ) 配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

## 背景信息

CEF作为第3层交换机制被大多数思科多层交换机使用。网络工程师必须了解CEF的工作方式，以便每天排除网络故障、丢包或数据包延迟情况。

独立模式下的Sup2T管理引擎或当许多企业网络当前部署VSS作为核心交换机时，实际上会聚合所有其他路由或交换设备。这也意味着转发大多数域内和域间流量以成功将数据包传送到其目的地。要实现此目的，Sup2T必须具有通过路由协议静态或动态获取的正确路由信息。

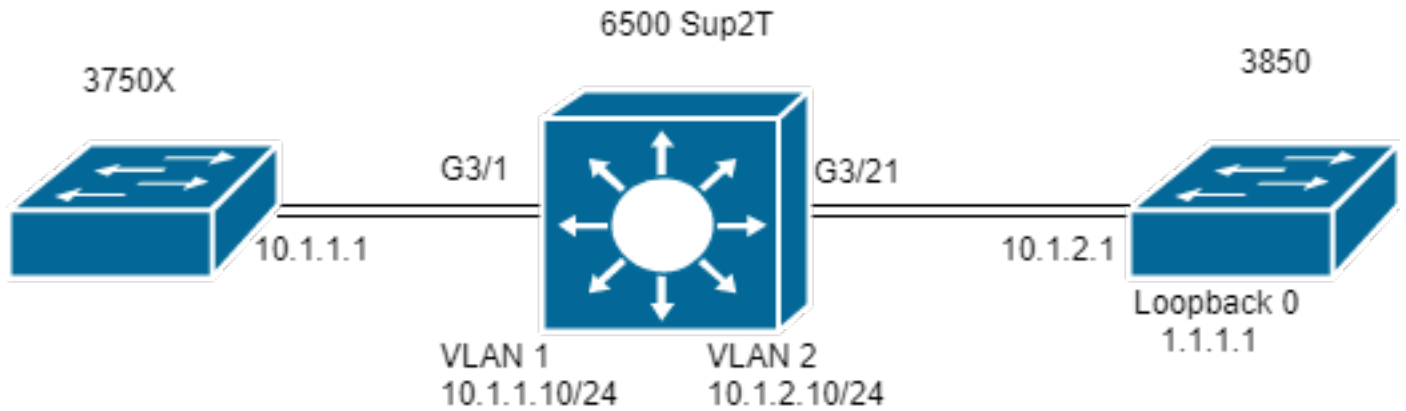
在模块化机箱中，除了管理引擎外可能还存在多个转发引擎。某些线卡 ( 特别是新一代线卡，如C6800-32P10G ) 已经包括了自己的转发引擎，以增强分组交换性能，CEF条目在本地执行查找，并导致资源最好地分配给通过不同线卡传入的流量。这些称为分布式转发卡(DFC)。

所有转发引擎共享的这些CEF条目可能因多种原因无法在硬件中分配，从软件缺陷状况、资源耗尽到CPU使用率过高，以及防止交换机有足够时间更新所有条目，这可能导致一系列不良事件。

## 网络图

网络

:



```
Switch#show module 3
```

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

## 识别分布式转发引擎上的CEF条目

在图中，独立式6506交换机安装了Supervisor 2T，并在插槽3中安装了DFC的线卡WS-6848-GE-TX。主机3750X通过端口G3/1连接到线卡，它将流量发送到3850的Loopback 0地址1.1.1.1

为此，3750X具有通过下一跳10.1.1.1 ( Sup2T交换机中VLAN 1的SVI ) 到IP地址1.1.1.1的静态路由。Sup2T交换机需要根据IP 1.1.1.1/32的静态路由条目，通过下一跳10.1.2.1将此流量路由到3850交换机，该下一跳是连接到VLAN 2中Sup2T的3850接口。

```
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
```

请注意，为简单起见，3750X和3850交换机都通过同一线卡连接到6500。这意味着流量在本地查找并在本地转发。

数据包通过Gi3/1接入Sup2T交换机，最终到达转发引擎（因为这是DFC）。转发引擎解析此数据包中的目标IP地址字段，并在已编程的CEF条目上查找最佳匹配（最长掩码）。

由于这是DFC卡，这意味着它有自己的CEF条目并且要验证它们，因此我们必须使用**attach [dec]命令连接到线卡，或为VSS连接交换机[1-2] mod [dec]命令。**

现在，您应该处于DFC提示符下，命令**show platform hardware cef** 或**show platform hardware cef vpn 0**返回为常规路由表（VPN 0/无VRF）编程的所有CEF条目。

由于目标是前缀1.1.1.1/32，因此您使用命令**show platform hardware cef vpn 0 lookup 1.1.1.1**。该命令返回前缀1.1.1.1的最佳匹配值，以及它实际转发流量的匹配值：

```
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 V12 ,0c11.678b.f6f7
```

CEF条目在IOS软件中，通过命令ip route 1.1.1.1 255.255.255.255 10.1.2.1编程静态条目，对它进行了编程。

您还可以通过命令show platform hardware cef 1.1.1.1 detail返回邻接条目，验证此条目是否被命中并且流量是通过此条目转发的：

```
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)
```

最后，邻接条目显示如何重写数据包以及是否由此邻接条目重写流量：

```
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和src\_mac是主要关注的值，表示为此数据包写入的新L2报头。目的MAC地址0c11.678b.f6f7是10.1.2.1，即3850（到达1.1.1.1的下一跳）：

```
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
```

此外，**Statistics**字段显示流量实际上已命中此邻接条目，并且L2报头会相应重写。

## 删除CEF条目

删除CEF条目有助于我们删除可能被错误编程（例如，指向错误的邻接条目）的任何条目，甚至用于培训目的。它还提供了修改路由路径的方法。

要删除CEF条目，您需要了解CEF条目是按顺序编程的，并且分配了硬件索引，例如：

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

代码：decap — 解封，+ — 推送标签

```
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 V11 ,a0ec.f930.3f40 261
10.1.2.1/32 V12 ,0c11.678b.f6f7 262 1.1.1.1/32 V12 ,0c11.678b.f6f7 <<<<< Our CEF entry of
interest has a HW index of 262.
...
```

此硬件索引是删除CEF条目的最重要元素，因为它用作参考。但是，要对其进行任何更改，必须将其转换为软件句柄。您可以使用命令test platform hardware cef index-conv hw\_to\_sw [hw index]实现此目的

```
MXC.CALO.Sup2T-dfc3#test platform hardware cef index-conv hw_to_sw 262
```

```
hw index: 262 ----> sw handle: 101
```

现在您知道了软件句柄，可以继续使用命令test platform hardware cef v4-delete [sw handle] mask [mask length] vpn [dec]删除CEF条目

```
MXC.CALO.s2TVSS-sw2-dfc3#test platform hardware cef v4-delete 101 mask 32 vpn 0
```

```
test_ipv4_delete: done.
```

**注意：**掩码长度值为32，因为这是主机特定路由(1.1.1.1/32)

现在，我们的CEF条目已删除：

```
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 V11 ,a0ec.f930.3f40
```

```
261 10.1.2.1/32 V12 ,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
```

请注意，**测试平台硬件cef vpn 0**命令是在DFC提示符下执行的。这样，CEF条目从DFC的CEF表中删除，而不是从管理引擎删除，您必须非常小心地从哪个转发引擎删除这些条目。

流量的更改可能会导致无法查看（在实验室测试中），这可能是由于另一个CEF条目的命中。请考虑始终匹配最精确的一个（最长掩码）。在本实验中，它符合：

```
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
```

那么，此条目实际上对数据包有何作用？：

```
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
```

Taken from a CPU packet capture using Catlayst 6500 NETDR tool. For NETDR capture tool details refer to: [Catalyst 6500 Series Switches Netdr Tool for CPU-Bound Packet Captures](#)

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

```
l2idb Po1, l3idb V11, 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 V12, 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
```

现在，所有通过线卡3进入的目的地为1.1.1.1的流量都会使用填充头重新循环并传送到CPU。有时，会看到另一个带丢弃邻接的0.0.0.0/0，而不是此CEF条目，并执行相同的操作。

**注意：**评估删除的CEF条目。这可能导致CPU使用率过高。通常配置默认路由0.0.0.0/0，并根据该路由转发流量（并导致数据包丢失）。



```
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...

现在，您了解了这些邻接条目的功能，可以继续添加CEF条目。在最后一部分，前缀1.1.1.1/32的CEF条目是通过测试平台硬件cef v4-delete命令删除的。现在，通过命令test platform hardware cef v4-insert [prefix]将其重新添加 [掩码长度] vpn [vpn编号]邻接[邻接索引]

为了验证此情况，请使用命令test platform hardware cef v4-insert 1.1.1.1 32 vpn 0 adjacency 114689。该条目已添加回DFC CEF表中：

```
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 V12 ,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
```

## 添加和删除VRF路由表的条目

在所有前面步骤中进行的整个配置中，已实施了show platform hardware cef命令中的vpn 0字符串。即使由于默认情况下该命令返回一般路由表或vpn 0的条目，因此似乎完全不必要，但这是为了始终牢记通过您添加和删除的文档从特定路由表实例(VRF)中添加或删除条目CEF条目1.1.1.1/32。但是，某些前缀很可能存在于不同的VRF(i)中e.10.x.x.x) 删除、添加或修改错误VRF的CEF条目可能会造成负面影响。

删除前缀为1.1.1.1/32的CEF条目，以用于VRFTEST\_VRF。有关CEF条目添加的详细说明，请参阅本文档的添加CEF条目部分。

要添加VRF，请使用命令ip vrf forwarding [VRF-NAME]将6500交换机中的SVI更改为建议的VRF，并最终在TEST\_VRF表中添加相同的静态路由：

```
MXC.CALO.Sup2T(config)#ip vrf TEST_VRF
```





