Comprendre les entrées CEF du commutateur Catalyst 6500 S2T

Table des matières

- Introduction
  - Conditions préalables
  - Exigences
  - Composants utilisés
- Informations générales
- Diagramme du réseau
- Identifier les entrées CEF sur les moteurs de transfert distribué
  - Supprimer les entrées CEF
- Ajouter une entrée CEF
  - Ajout et suppression d'entrées pour les tables de routage VRF

Introduction

Ce document décrit comment Cisco Catalyst 6500 avec Supervisor Sup2T programme les entrées CEF (Cisco Express Forwarding) configurées sur le logiciel Cisco IOS dans le matériel des cartes de ligne utilisé pour réaliser le transfert de paquets.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Cisco Express Forwarding (CEF)
- Commutateurs Cisco Catalyst, série 6500
- Carte DFC (Distributed Forwarding Card) Cisco

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions matérielles et logicielles suivantes :

- Carte de ligne Cisco Catalyst 6500 WS-X6848-GE-TX (avec DFC4).
- Cisco Catalyst 6500 avec Supervisor 2T sur IOS version 15.2.1SY5

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Informations générales

CEF en tant que mécanisme de commutation de couche 3 est utilisé par la plupart des commutateurs multicouches Cisco.Il est impératif pour les ingénieurs réseau de comprendre comment CEF fonctionne afin de dépanner les pannes de réseau, les pertes de paquets ou les scénarios de retard de paquets sur une base quotidienne.

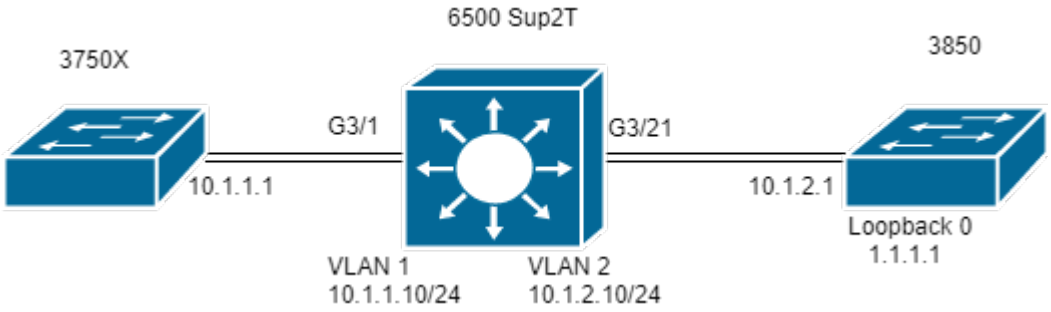
Le superviseur Sup2T en mode autonome ou VSS étant actuellement déployé par de nombreux réseaux d'entreprise en tant que commutateur principal, regroupe pratiquement tous les autres périphériques de routage ou de commutation. Cela signifie également que transfère la plupart du trafic intra et inter-domaines afin de livrer correctement les paquets à leurs destinations. Pour ce faire, Sup2T doit disposer d'informations de routage appropriées acquises de manière statique ou dynamique via des protocoles de routage.

Dans un châssis modulaire, plusieurs moteurs de transfert peuvent exister en plus de ceux du superviseur. Certaines cartes de ligne (en particulier celles de nouvelle génération telles que C6800-32P10G) incluent déjà leur propre moteur de transfert afin d'améliorer les performances de commutation de paquets. La recherche des entrées CEF est exécutée localement et entraîne une meilleure distribution des ressources pour le trafic entrant sur différentes cartes de ligne. Ces cartes sont appelées cartes de transfert distribuées (DFC).

Ces entrées CEF partagées entre tous les moteurs de transfert peuvent ne pas être allouées dans le matériel pour plusieurs raisons, d'une condition de défaut logiciel, d'épuisement des ressources à des conditions CPU élevées et empêche le commutateur d'avoir assez de temps pour mettre à jour toutes les entrées, cela peut provoquer une série d'événements indésirables.

Diagramme du réseau

Réseau:




```
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
```

Identifier les entrées CEF sur les moteurs de transfert distribué

Dans le schéma, un commutateur 6506 autonome dispose d'un Supervisor 2T installé ainsi que d'une carte de ligne WS-6848-GE-TX avec une carte DFC dans le logement 3. L'hôte 3750X qui est connecté à la carte de ligne via le port G3/1 envoie le trafic à l'adresse de bouclage 0 1.1.1.1 du 3850.

Pour cela, 3750X a une route statique vers l'adresse IP 1.1.1.1 à 10.1.1.10 de tronçon suivant qui est l'interface SVI de VLAN 1 dans le commutateur Sup2T. Le commutateur Sup2T doit acheminer ce trafic vers le commutateur 3850 basé sur une entrée de route statique pour IP 1.1.1.1/32 via le saut suivant 10.1.2.1 qui est l'interface 3850 connectée au Sup2T dans le VLAN 2.

```
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
```

Pour des raisons de simplicité, les commutateurs 3750X et 3850 sont tous deux connectés au 6500 via la même carte de ligne. Cela signifie que le trafic est recherché localement et transféré localement aussi.

Un paquet entre dans le commutateur Sup2T via Gi3/1 et finit par atteindre le moteur de transfert (puisqu'il s'agit d'une DFC). Le moteur de transfert analyse le champ d'adresse IP de destination dans ce paquet et recherche les entrées CEF programmées pour la meilleure correspondance (masque le plus long).

Puisqu'il s'agit d'une carte DFC, cela signifie qu'elle a ses propres entrées CEF et pour les vérifier, il est nécessaire pour nous d'attacher à la carte de ligne avec la commande attach [dec] ou attach switch [1-2] mod [dec] pour VSS.

À présent, vous devez être dans l'invite DFC, la commande show platform hardware cef ou show platform hardware cef vpn 0 et retourner toutes les entrées CEF programmées pour la table de routage générale (VPN 0/ No VRF).

Puisque l'objectif est le préfixe 1.1.1.1/32, vous utilisez la commande show platform hardware cef vpn 0 lookup 1.1.1.1. La commande retourne la meilleure correspondance pour le préfixe 1.1.1.1 et celle qu'elle utilise pour réellement transférer le trafic :

<#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
```

L'entrée CEF est là, elle a été programmée comme résultat de notre entrée statique programmée dans le logiciel IOS via la commande ip route 1.1.1.1 255.255.255.255 10.1.2.1.

Vous pouvez également vérifier que cette entrée obtient des résultats et que le trafic est transféré avec cette entrée via les commandes show platform hardware cef 1.1.1.1 detail qui renvoient une entrée de contiguïté :

```
<#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)
```

Enfin, l'entrée de contiguïté indique comment le paquet est réécrit et si le trafic est réécrit par cette entrée de contiguïté :

```
<#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                    <<<<
```

Les valeurs dest_mac et src_mac sont les valeurs d'intérêt principal, qui indiquent les nouveaux en-têtes L2 qui sont écrits pour ce paquet. L'adresse MAC de destination 0c11.678b.f6f7 est 10.1.2.1, qui est le 3850 (tronçon suivant pour atteindre 1.1.1.1) :

```
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
```

En outre, le champ Statistics montre que le trafic atteint réellement cette entrée de contiguïté et les en-têtes L2 sont réécrits en conséquence.

Supprimer les entrées CEF

Supprimer les entrées CEF peut nous aider à supprimer toute entrée qui pourrait être mal programmée (à une mauvaise entrée de contiguïté par exemple) ou même à des fins de formation. Il permet également de modifier un chemin de routage.

Pour supprimer une entrée CEF, vous devez comprendre que les entrées CEF sont programmées séquentiellement et qu'un index matériel leur est attribué, par exemple :

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

Codes : decap - Décapsulation, + - Push Label

<#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.
...
```

Cet index matériel est l'élément le plus important pour supprimer une entrée CEF puisqu'il est utilisé comme référence. Cependant, pour pouvoir y apporter des modifications, il faut le convertir en un descripteur logiciel. Pour ce faire, utilisez la commande test platform hardware 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

Maintenant que vous connaissez le handle logiciel, vous pouvez poursuivre la suppression de l'entrée CEF avec la commande test platform hardware cef v4-delete [sw handle] mask [mask length] vpn [dec]

<#root>

MXC.CALO.s2TVSS-sw2-dfc3#

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

test_ipv4_delete: done.

Remarque : La valeur de longueur de masque est 32 car il s'agit d'une route spécifique à l'hôte (1.1.1.1/32)

Maintenant, notre entrée CEF est supprimée :

<#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
```

Notez que la commande test platform hardware cef vpn 0 a été exécutée sous l'invite DFC. De cette façon, l'entrée CEF a été supprimée de la table CEF de la DFC et NON du superviseur, vous devez être vraiment prudent sur le moteur de transfert duquel les entrées sont supprimées.

Une modification du trafic risque de ne pas être visible (dans le cas d'un test en laboratoire), ce qui peut être dû au résultat d'une autre entrée CEF. Pensez à toujours faire correspondre le masque le plus exact (masque le plus long). Au cours de ces travaux pratiques, il atteint :

<#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
```

Que fait donc cette entrée avec le paquet ? :

```
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
```

Maintenant, tout le trafic avec la destination 1.1.1.1 qui entre par la carte de ligne 3 est recirculé avec l'en-tête de cale et envoyé au CPU. Parfois, au lieu de cette entrée CEF, un autre 0.0.0.0/0 avec drop adjaceny est vu et fait exactement la même chose.

Remarque : Évaluez les entrées CEF qui sont supprimées. Une utilisation CPU élevée peut être causée à cause de cela. Généralement, une route par défaut 0.0.0.0/0 est configurée et le trafic est transféré en fonction de celle-ci (et entraîne la perte de paquets).

Ajouter une entrée CEF

Lorsqu'une entrée CEF est ajoutée, dans la plupart des cas résout tout problème de mauvaise programmation qui entraîne une perte de paquets, un retard de paquets ou une utilisation élevée du CPU. La connaissance de la façon d'installer les entrées CEF dans le matériel, fournit non seulement la capacité de corriger une entrée mal programmée, mais aussi de manipuler tout transfert de paquet par la recirculation du paquet, le pointer vers une interface complètement différente ou le saut suivant, réécrire un paquet routé comme souhaité et/ou l'abandonner, etc. Tout cela,

sans rechargement de la boîte, supprimer et régler la configuration ou toute modification apparente. L'ajout d'une entrée CEF peut être effectué sans passer en mode de configuration. (Comme vous l'avez également fait avec la procédure de suppression d'entrée CEF expliquée dans la section précédente).

En fait, il y a deux situations ici, quand vous avez une entrée ARP valide pour le saut suivant, dans ce cas 10.1.2.1 et quand vous ne l'avez pas (Pour n'importe quelle raison). La deuxième situation vous oblige à créer une entrée ARP valide (via ARP statique) :

Étape 1. Il y a une entrée ARP dans le commutateur pour 10.1.2.1 qui est le saut suivant pour 1.1.1.1.

<#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
```

Une entrée ARP est programmée en tant que route hôte (/32) dans la table CEF :

<#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)
```

Tout paquet avec une adresse IP de destination qui a la même liaison de données que le tronçon suivant doit être transféré via la même interface et réécrit avec les mêmes en-têtes L2.

Même si cela peut sembler assez évident au début, c'est en fait l'élément le plus important pour ajouter une entrée CEF, vous devez lui dire comment un paquet doit être réécrit avec une entrée de contiguïté CEF spécifique.

Étape 2. Supposons maintenant qu'aucune entrée ARP n'est créée automatiquement pour cette opération. Vous devez donc créer une entrée ARP statique.

Pour ce faire, vous devez connaître l'adresse MAC du périphérique qui est utilisé comme tronçon suivant pour le préfixe 10.1.2.1, afin qu'il soit envoyé à 0c11.678b.f6f7. S'il y a déjà une entrée d'adresse MAC dans la sortie de commande show mac address-table address 0c11.678b.f6f7 qui est correcte, sinon vous devez créer une entrée MAC statique :

```
<#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
```

Étape 3. Enfin, une entrée ARP statique doit être créée pour qu'une entrée CEF soit programmée :

```
<#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...
```

Maintenant que vous comprenez ce que font ces entrées de contiguïté, vous pouvez enfin ajouter une entrée CEF. Dans la dernière section, l'entrée CEF pour le préfixe 1.1.1.1/32 a été supprimée via la commande test platform hardware cef v4-delete. À présent, ajoutez-le à nouveau via la commande test platform hardware cef v4-insert [prefix] [mask length] vpn [vpn number] adjacency [adjacency index]

Afin de vérifier ceci, utilisez la commande test platform hardware cef v4-insert 1.1.1.1 32 vpn 0 adjacency 114689. L'entrée a été rajoutée dans la table DFC CEF :

```
<#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
```

Ajout et suppression d'entrées pour les tables de routage VRF

Tout au long de la configuration effectuée à partir de toutes les étapes précédentes, la chaîne vpn

0 dans les commandes show platform hardware cef a été appliquée. Même si cela semble complètement inutile puisque la commande par défaut retourne les entrées pour la table de routage générale ou le vpn 0, ceci a été fait exprès afin d'avoir toujours à l'esprit que les entrées sont ajoutées ou supprimées à partir d'instances de table de routage spécifiques (VRF), par le document que vous avez ajouté et supprimé l'entrée CEF 1.1.1.1/32. Cependant, certains préfixes sont très susceptibles d'exister dans différents VRF (i. e. 10.x.x.x) et supprimer, ajouter ou modifier une entrée CEF pour un VRF incorrect peut avoir un impact négatif.

Supprimez une entrée CEF avec le préfixe 1.1.1.1/32 pour VRF TEST_VRF. Pour une description détaillée de l'ajout d'entrées CEF, référez-vous à la section Ajouter une entrée CEF de ce document.

Afin d'ajouter le VRF, changez les SVI dans le commutateur 6500 en VRF proposé avec la commande ip vrf forwarding [VRF-NAME] et enfin ajoutez la même route statique dans notre table TEST_VRF :

<#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
```

Les VRF sont également programmés séquentiellement. Il s'agissait du premier VRF dans le commutateur (aucun autre VRF n'a été configuré précédemment), donc le numéro de vpn pour cette instance de VRF doit être 1. Exécutez la commande show platform hardware cef vpn 1 afin de vérifier que ceci est vrai :

<#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)
```

Il est important de connaître le numéro VPN réel et l'index logiciel de cette entrée afin de pouvoir continuer à la supprimer ou l'ajouter à partir de/à cette instance VRF :

<#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
```

Considérez que dans un réseau de production, la perte de paquets et l'audio saccadé ou la mauvaise vidéo est subie en raison de ces entrées CEF condition. Par conséquent, il est recommandé d'effectuer ces tests dans une fenêtre de maintenance.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.