

Vérification des affectations VQI ASR9000 dans CEF

Table des matières

[Introduction](#)

[Informations générales](#)

[Vérifier les affectations VQI](#)

Introduction

Ce document décrit comment vérifier les index de file d'attente virtuelle (VQI) et les attribuer correctement dans Cisco Express Forwarding (CEF) dans un routeur à services d'agrégation 9000 (ASR9K).

Informations générales

Pour que les paquets puissent être transférés d'une interface à une autre dans un ASR9K, ils doivent traverser le fabric. Il n'y a pas de commutation locale dans un ASR9K. Mais comment un paquet passe-t-il d'une interface à une autre ? Pour ce faire, des interfaces VQI sont attribuées à chaque interface. Ainsi, le fabric sait quelle carte de ligne (LC) et quel processeur réseau (NP) acheminer le paquet.

Parfois cependant, comme dans le cas de [CSCvc83681](#), une VQI incorrecte peut être attribuée et le trafic peut faire l'objet d'un blackholing dans le routeur.

Vérifier les affectations VQI

Reportez-vous à cette section afin de vérifier les affectations VQI.

Tout d'abord, identifiez les interfaces d'entrée et de sortie pour l'adresse IP (Internet Protocol) de flux, source et de destination, avec la commande `show cef <prefix> detail`.

Cela permet d'identifier les LC à examiner pour les attributions VQI.

Voici l'adresse source :

```
RP/0/RSP0/CPU0:ASR9006-H#show cef 123.29.62.12 detail
Tue May  1 10:54:50.356 EDT
123.29.62.12/32, version 325561, internal 0x1000001 0x0 (ptr 0x76a07a40) [1], 0x0 (0x73ffbf50),
0xa28 (0x75e3133c)
Updated May  1 10:26:51.592
remote adjacency to TenGigE0/1/0/5
Prefix Len 32, traffic index 0, precedence n/a, priority 1
gateway array (0x74bff484) reference count 3, flags 0x68, source lsd (5), 1 backups
[2 type 5 flags 0x8401 (0x7216f3d0) ext 0x0 (0x0)]
LW-LDI[type=5, refc=3, ptr=0x73ffbf50, sh-ldi=0x7216f3d0]
```

```

gateway array update type-time 1 May 1 10:26:51.592
LDI Update time May 1 10:26:51.592
LW-LDI-TS May 1 10:26:51.592
via 10.94.1.182/32, TenGigE0/1/0/5, 6 dependencies, weight 0, class 0 [flags 0x0]
path-idx 0 NHID 0x0 [0x7181cf4 0x0]
next hop 10.94.1.182/32
remote adjacency
local label 24088 labels imposed {86}
via 10.94.1.150/32, TenGigE0/1/0/7, 6 dependencies, weight 0, class 0 [flags 0x0]
path-idx 1 NHID 0x0 [0x7181d018 0x0]
next hop 10.94.1.150/32
remote adjacency
local label 24088 labels imposed {86}

```

Load distribution: 0 1 (refcount 2)

Hash	OK	Interface	Address
0	Y	TenGigE0/1/0/5	remote
1	Y	TenGigE0/1/0/7	remote

Voici l'adresse de destination :

```

RP/0/RSP0/CPU0:ASR9006-H#show cef 123.29.62.1 detail
Tue May 1 10:53:14.531 EDT
123.29.62.1/32, version 334286, internal 0x1000001 0x0 (ptr 0x74bf1a04) [1], 0x0 (0x73ffbeb0),
0xa20 (0x75e310d4)
Updated May 1 10:53:12.459
remote adjacency to TenGigE0/0/0/2
Prefix Len 32, traffic index 0, precedence n/a, priority 1
gateway array (0x74c025ec) reference count 27, flags 0x68, source lsd (5), 1 backups
[19 type 4 flags 0x8401 (0x7216f390) ext 0x0 (0x0)]
LW-LDI[type=1, refc=1, ptr=0x73ffbeb0, sh-ldi=0x7216f390]
gateway array update type-time 1 Apr 30 17:03:05.246
LDI Update time Apr 30 17:03:05.246
LW-LDI-TS Apr 30 17:03:05.247
via 10.94.0.10/32, TenGigE0/0/0/2, 4 dependencies, weight 0, class 0 [flags 0x0]
path-idx 0 NHID 0x0 [0x7181ce20 0x7181d06c]
next hop 10.94.0.10/32
remote adjacency
local label 24012 labels imposed {ImplNull}
via 10.94.2.9/32, TenGigE0/0/0/3, 4 dependencies, weight 0, class 0 [flags 0x0]
path-idx 1 NHID 0x0 [0x7181ce74 0x7181d0c0]
next hop 10.94.2.9/32
remote adjacency
local label 24012 labels imposed {ImplNull}

```

Load distribution: 0 1 (refcount 19)

Hash	OK	Interface	Address
0	Y	TenGigE0/0/0/2	remote
1	Y	TenGigE0/0/0/3	remote

À partir de ces sorties, vous voyez que LC 1 est le LC d'entrée et LC 0 est le LC de sortie, les deux ont deux ports afin d'équilibrer la charge du trafic.

Ensuite, vous devez identifier combien de NP sont sur le LC d'entrée et de sortie avec la commande **show controller np ports all loc <LC>**.

La LC d'entrée comporte 8 NP :

```
RP/0/RSP0/CPU0:ASR9006-H#show controller np ports all loc 0/1/CPU0
Tue May 1 10:56:57.996 EDT
```

```
Node: 0/1/CPU0:
```

```
NP Bridge Fia Ports
-- -----
0 -- 0 TenGigE0/1/0/0 - TenGigE0/1/0/2
1 -- 0 TenGigE0/1/0/3 - TenGigE0/1/0/5
2 -- 1 TenGigE0/1/0/6 - TenGigE0/1/0/8
3 -- 1 TenGigE0/1/0/9 - TenGigE0/1/0/11
4 -- 2 TenGigE0/1/0/12 - TenGigE0/1/0/14
5 -- 2 TenGigE0/1/0/15 - TenGigE0/1/0/17
6 -- 3 TenGigE0/1/0/18 - TenGigE0/1/0/20
7 -- 3 TenGigE0/1/0/21 - TenGigE0/1/0/23
```

La LC de sortie a 2 NP :

```
RP/0/RSP0/CPU0:ASR9006-H#show controller np ports all loc 0/0/cPU0
Tue May 1 10:55:27.661 EDT
```

```
Node: 0/0/CPU0:
```

```
NP Bridge Fia Ports
-- -----
0 -- 0 TenGigE0/0/0/0 - TenGigE0/0/0/3
1 -- 1 TenGigE0/0/1/0 - TenGigE0/0/1/3
```

Ensuite, vérifiez le LC d'entrée avec la commande **show cef <préfixe de destination> hardware ingress detail loc <ingress lc> | i vqi** et le LC de sortie avec la commande **show cef <dst prefix> hardware egress detail loc <egress lc> | vqi**.

Ces informations nous renseignent sur la façon dont chaque processeur réseau est programmé pour atteindre les interfaces de sortie. Dans ce cas, comme il y a huit NP sur la LC d'entrée et deux liaisons ECMP (Equal Cost Multi-Path) sur la LC de sortie, il y a 16 entrées. Les huit premières entrées concernent la première liaison ECMP et les huit suivantes concernent la deuxième liaison ECMP. Chaque jeu de huit doit correspondre, ce qui signifie que chaque processeur réseau est programmé pour faire de même. Chaque ensemble doit être différent, bien qu'il existe deux interfaces distinctes. S'ils sont identiques, vous risquez de rencontrer un problème de mauvaise programmation VQI CEF.

```
RP/0/RSP0/CPU0:ASR9006-H#show cef 123.29.62.1 hardware ingress loc 0/1/CPU0 | i vqi
Tue May 1 10:56:27.064 EDT
sfp/vqi      : 0x58
sfp/vqi      : 0x59
```

```
sfp/vqi      : 0x59  
sfp/vqi      : 0x59
```

Vérifiez le LC de sortie pour vous assurer qu'il est programmé correctement. Dans ce cas, il y a deux NP et deux liaisons ECMP, il y a donc deux ensembles de deux VQI qui doivent être programmés.

```
RP/0/RSP0/CPU0:ASR9006-H#show cef 123.29.62.1 hardware egress loc 0/0/CPU0 | i vqi  
Tue May  1 10:57:29.221 EDT  
    out_lbl_invalid: 0          match: 0          vqi/lag-id: 0x0  
    out_lbl_invalid: 0          match: 0          vqi/lag-id: 0x0  
    sfp/vqi      : 0x58  
    sfp/vqi      : 0x58  
        out_lbl_invalid: 0      match: 0          vqi/lag-id: 0x0  
        out_lbl_invalid: 0      match: 0          vqi/lag-id: 0x0  
    sfp/vqi      : 0x59  
    sfp/vqi      : 0x59
```

La dernière chose à vérifier est l'affectation VQI sur les interfaces.

Ici, vous pouvez vérifier la variable switch_fabric_port et la convertir de décimal en hexadécimal. 88 étant 58 et 89 étant 59, ces valeurs correspondent aux affectations VQI de ces commandes, ce qui signifie que CEF est programmé correctement pour le transport VQI dans l'ASR9K.

```
RP/0/RSP0/CPU0:ASR9006-H#show controller pm interface ten 0/0/0/2  
Tue May  1 10:58:52.024 EDT
```

```
Ifname(1): TenGigE0_0_0_2, ifh: 0x4000140 :  
iftype          0x1e  
egress_uidb_index 0x7, 0x7  
ingress_uidb_index 0x7, 0x7  
port_num        0x2  
subslot_num     0x0  
ifsubinst       0x0  
ifsubinst_port  0x2  
phy_port_num   0x2  
channel_id      0x0  
channel_map     0x0  
lag_id          0x0  
virtual_port_id 0x0  
switch_fabric_port 88  
in_tm_qid_fid0 0x20002  
in_tm_qid_fid1 0xffffffff  
in_qos_drop_base 0x690001  
out_tm_qid_fid0 0x20022  
out_tm_qid_fid1 0xffffffff  
np_port         0x6  
  
out_qos_drop_base 0x6900a1  
bandwidth       10000000 kbps  
ing_stats_ptr 0x53016a, 0x0  
egr_stats_ptr 0x53017b, 0x0  
l2_transport    0x0  
ac_count        0x0  
parent_ifh     0x0  
parent_bundle_ifh 0x0  
L2 protocols bmap 0x1000000  
Cluster interface 0
```

```
RP/0/RSP0/CPU0:ASR9006-H#show controller pm interface ten 0/0/0/3  
Tue May  1 10:59:08.886 EDT
```

```
Ifname(1): TenGigE0_0_0_3, ifh: 0x4000180 :
iftype          0x1e
egress_uidb_index 0x8, 0x8
ingress_uidb_index 0x8, 0x8
port_num        0x3
subslot_num     0x0
ifsubinst       0x0
ifsubinst port   0x3
phy_port_num    0x3
channel_id      0x0
channel_map     0x0
lag_id          0x0
virtual_port_id 0x0
switch_fabric_port 89
in_tm_qid_fid0 0x30002
in_tm_qid_fid1 0xffffffff
in_qos_drop_base 0x6e0001
out_tm_qid_fid0 0x30022
out_tm_qid_fid1 0xffffffff
np_port         0x7

out_qos_drop_base 0x6e00a1
bandwidth        10000000 kbps
ing_stats_ptrs   0x530183, 0x0
egr_stats_ptrs   0x530194, 0x0
l2_transport     0x0
ac_count         0x0
parent_ifh       0x0
parent_bundle_ifh 0x0
L2 protocols bmap 0x1000000
Cluster interface 0
```

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.