

Compréhension des distributions de poids de CEF dans partager de chargement de coût inégal

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Aperçu UCMP](#)

[Configurations initiales](#)

[Metric weights/chargement](#)

[Détermination de variance UCMP](#)

[Compréhension des poids](#)

[Détermination des valeurs de poids](#)

[Poids](#)

[Poids normal](#)

[Rapports manipulants de poids de CEF/chargement](#)

[Exemple 1 : Rapport de poids/chargement de 26/5](#)

[Exemple 2 : Rapport de poids/chargement de 30/1](#)

Introduction

Ce document décrit les aspects de la compréhension, configurant, et vérifie le coût inégal multivoie dans IOS-XR. Également nous passons par des exemples des manipulations de poids pour afficher comment la mesure de chemin à une destination influence le chargement sur un lien.

Conditions préalables

Ce document n'a pas des conditions préalables.

Exigences

Des exemples ci-dessous sont basés sur IOS-XR 6.4.1.

[Composants utilisés](#)

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

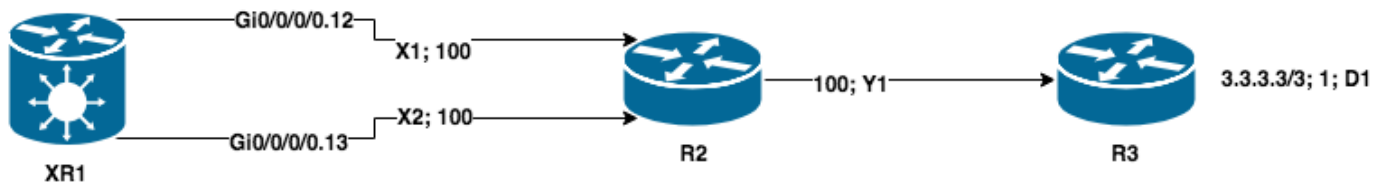
Aperçu UCMP

L'Équilibrage de charge multivoie de coût inégal (UCMP) fournit la capacité pour équilibrer la charge le trafic proportionnellement à travers des plusieurs chemins, avec le coût différent. Généralement, les chemins de bande passante élevée ont des mesures inférieures de Protocole IGP (Interior Gateway Protocol) configurées, de sorte qu'ils forment les chemins d'IGP les plus courts.

L'Équilibrage de charge UCMP étant activé, les protocoles peuvent utiliser encore des chemins de bande passante inférieure ou des chemins plus élevés de coût pour le trafic, et peuvent installer ces chemins sur le Forwarding Information Base (FIB). Ces protocoles installent toujours des plusieurs chemins sur le même FIB de destination in, mais chaque chemin aura une « mesure/poids de chargement » associés avec lui. Le FIB emploie ces mesure/poids de chargement pour décider le niveau de trafic qui doit être envoyé sur un chemin de bande passante élevée et le niveau de trafic qui doit être envoyé sur un chemin de bande passante inférieure.

Traditionnellement, l'EIGRP a été le seul IGP qui prend en charge la caractéristique UCMP, mais dans IOS-XR UCMP est pris en charge pour tous les IGP, routage statique, et BGP. Dans ce document, nous expliquerons la caractéristique UCMP utilisant l'OSPF comme base de nos exemples, mais les informations ici s'appliquent également à l'IS-IS et à d'autres protocoles UCMP-capables.

Diagramme de topologie



Configurations initiales

```
XR1
!
hostname XR1
!
interface GigabitEthernet0/0/0/0.12
description TO R2
ipv4 address 12.0.0.1 255.255.255.0
encapsulation dot1q 12
!
interface GigabitEthernet0/0/0/0.13
description TO R2
ipv4 address 13.0.0.1 255.255.255.0
encapsulation dot1q 13
! router ospf 1 address-family ipv4 area 0 ! interface GigabitEthernet0/0/0/0.12      cost 100
!
  interface GigabitEthernet0/0/0/0.13
    cost 100
!
!
!
end
```

```

R2
!
hostname R2
!
interface Ethernet0/0.12
  description TO XR1
  encapsulation dot1Q 12
  ip address 12.0.0.2 255.255.255.0
!
interface Ethernet0/0.13
  description TO XR1
  encapsulation dot1Q 13
  ip address 13.0.0.2 255.255.255.0
!
interface Ethernet0/1
  description TO R3
  ip address 172.16.23.2 255.255.255.0
  ip ospf cost 100
!
!
router ospf 1
  network 0.0.0.0 255.255.255.255 area 0
!
end

```

```

R3
!
hostname R3
!
interface Loopback0
  description FINAL_DESTINATION
  ip address 3.3.3.3 255.255.255.255
!
interface Ethernet0/0
  description TO R2
  ip address 172.16.23.3 255.255.255.0
!
router ospf 1
  network 0.0.0.0 255.255.255.255 area 0
!
end

```

Metric weights/chargement

Dans IOS-XR, quand nous installons des plusieurs chemins sur une destination, la destination est assignée une valeur de poids qui indique la répartition de charge pour un lien particulier. Cette valeur est inversement proportionnelle à la mesure de chemin à la destination, plus le coût est élevé, plus le poids est assigné humblement. Ceci permet au CEF pour exécuter intelligemment chargement-partager des liens en conduisant aux destinations.

Quand des chemins ECMP sont installés, des valeurs de poids assignées sont toujours placées à 0 pour tous les chemins, ceci signifie que le trafic chargement-est partagé également. Si nous vérifions le CEF nous pouvons confirmer que des poids de 0 ont été assignés pour chaque chemin.

```
RP/0/RP0/CPU0:XR1#show cef 3.3.3.3/32 detail
```

```

3.3.3.3/32, version 87, internal 0x1000001 0x0 (ptr 0xd689b50) [1], 0x0 (0xd820648), 0x0 (0x0)
Updated Nov 11 22:15:58.953
remote adjacency to GigabitEthernet0/0/0/0.12

```

```

Prefix Len 32, traffic index 0, precedence n/a, priority 1
gateway array (0xd6b32f8) reference count 2, flags 0x0, source rib (7), 0 backups
    [3 type 3 flags 0x8401 (0xd759758) ext 0x0 (0x0)]
LW-LDI[type=3, refc=1, ptr=0xd820648, sh-ldi=0xd759758]
gateway array update type-time 1 Nov 11 22:15:58.953
LDI Update time Nov 11 22:15:58.953
LW-LDI-TS Nov 11 22:15:58.953
  via 12.0.0.2/32, GigabitEthernet0/0/0/0.12, 4 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0xe14b0a0 0x0]
    next hop 12.0.0.2/32
    remote adjacency
  via 13.0.0.2/32, GigabitEthernet0/0/0/0.13, 4 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 1 NHID 0x0 [0xe14b128 0x0]
    next hop 13.0.0.2/32
    remote adjacency

Load distribution: 0 1 (refcount 3)

```

Hash	OK	Interface	Address
0	Y	GigabitEthernet0/0/0/0.12	remote
1	Y	GigabitEthernet0/0/0/0.13	remote

Détermination de variance UCMF

Si nous voulons activer UCMF, permettez-nous commençant en plaçant le coût différemment sur XR1, pour ceci, nous placera le coût en tant que ci-dessous :

```

router ospf 1
address-family ipv4
area 0
interface Loopback0
!
interface GigabitEthernet0/0/0/0.12
cost 50
!
interface GigabitEthernet0/0/0/0.13
cost 100
!
!
end

```

```
RP/0/RP0/CPU0:XR1#show route 3.3.3.3/32
```

```

Routing entry for 3.3.3.3/32
  Known via "ospf 1", distance 110, metric 151, type intra area
  Installed Nov 11 22:32:48.289 for 00:00:32
  Routing Descriptor Blocks
    12.0.0.2, from 3.3.3.3, via GigabitEthernet0/0/0/0.12
      Route metric is 151
  No advertising protos.

```

Pour considérer d'autres chemins pour UCMF que nous devons déterminer si ce sont éligibles. IOS-XR utilise des critères d'un pourcentage pour l'IS-IS et l'OSPF, ceci est basé sur la commande de processus de routeur de **<value> de variance d'ucmf**. Les deux chemins que nous avons sont :

mesure de chemin 1 (pm1) = 151

mesure de chemin 2 (pm2) = 201

Boucle libre prochain-saut être installé a basé sur UCMP \leq (variance \ast) de mesure de chemin primaire/100.

Combien de chemin primaire doit devenir la portée la plus mauvaise mesure de chemin (pm2) est dans ce cas de 134 pour cent de 151, qui a comme conséquence 202. C'est la valeur précise de variance que nous devons configurer pour rendre le chemin éligible.

```
!  
router ospf 1  
  ucmp variance 134  
!  
  
RP/0/RP0/CPU0:XR1#show route 3.3.3.3/32  
  
Routing entry for 3.3.3.3/32  
  Known via "ospf 1", distance 110, metric 151, type intra area  
  Installed Nov 11 22:36:45.720 for 00:00:09  
  Routing Descriptor Blocks  
    12.0.0.2, from 3.3.3.3, via GigabitEthernet0/0/0/0.12  
      Route metric is 151, Wt is 4294967295  
    13.0.0.2, from 3.3.3.3, via GigabitEthernet0/0/0/0.13  
      Route metric is 151, Wt is 3226567396  
  No advertising protos.
```

[Spoiler](#)

Note: La valeur de variance n'a aucune incidence sur les résultats de poids. Dans ce cas une variance minimum de 134 ou une variance de 10000 (valeur maximum) aurait pour mener aux mêmes résultats de poids, au lieu de cela, les valeurs de coût sont celle qui influencent les poids en résultant, en tant que ces derniers sont inversement proportionnelle entre eux.

Remarque: La valeur de variance n'a aucune incidence sur les résultats de poids. Dans ce cas une variance minimum de 134 ou une variance de 10000 (valeur maximum) aurait pour mener aux mêmes résultats de poids, au lieu de cela, les valeurs de coût sont celle qui influencent les poids en résultant, en tant que ces derniers sont inversement proportionnelle entre eux.

Compréhension des poids

Nous avons deux types différents de poids dans IOS-XR, **poids** et **poids normaux**. L'utilisation de ces derniers est basée sur combien de positions d'informations parasites sont prises en charge sur une plate-forme particulière, XRv9000 les positions d'informations parasites du support 32, des positions d'informations parasites du support 64 ASR 9000 et CRS-X respectivement. Ceci signifie que, quand le routeur programme les valeurs de poids, le coefficient ne peut pas dépasser la limite de position d'informations parasites de la plate-forme particulière. Nous pouvons observer quels poids normaux sont programmés en émettant la commande de **<location> d'emplacement de détail de <prefix> de show cef**. Basé sur la valeur définie de coût, nous avons des 18, la répartition de charge 13, qui signifie que 31 positions d'informations parasites ont été assignées (18+13).

```
RP/0/RP0/CPU0:XR1#show cef 3.3.3.3/32 detail
```

```
3.3.3.3/32, version 23, internal 0x1000001 0x0 (ptr 0xd3ecb50) [1], 0x0 (0xd583610), 0x0 (0x0)  
Updated Nov 11 22:36:45.723  
remote adjacency to GigabitEthernet0/0/0/0.12
```

```

Prefix Len 32, traffic index 0, precedence n/a, priority 1
gateway array (0xd4163d8) reference count 1, flags 0x0, source rib (7), 0 backups
    [2 type 3 flags 0x8401 (0xd4bc7b8) ext 0x0 (0x0)]
LW-LDI[type=3, refc=1, ptr=0xd583610, sh-ldi=0xd4bc7b8]
gateway array update type-time 1 Nov 11 22:36:45.723
LDI Update time Nov 11 22:36:45.729
LW-LDI-TS Nov 11 22:36:45.729
    via 12.0.0.2/32, GigabitEthernet0/0/0/0.12, 6 dependencies, weight 4294967295, class 0 [flags
0x0]
    path-idx 0 NHID 0x0 [0xe14b1b0 0x0]
    next hop 12.0.0.2/32
    remote adjacency
    via 13.0.0.2/32, GigabitEthernet0/0/0/0.13, 6 dependencies, weight 3226567396, class 0 [flags
0x0]
    path-idx 1 NHID 0x0 [0xe14b128 0x0]
    next hop 13.0.0.2/32
    remote adjacency

Weight distribution:
slot 0, weight 4294967295, normalized_weight 18, class 0
slot 1, weight 3226567396, normalized_weight 13, class 0

Load distribution: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 (refcount
2)

```

Hash	OK	Interface	Address
0	Y	GigabitEthernet0/0/0/0.12	remote
1	Y	GigabitEthernet0/0/0/0.12	remote
2	Y	GigabitEthernet0/0/0/0.12	remote
3	Y	GigabitEthernet0/0/0/0.12	remote
4	Y	GigabitEthernet0/0/0/0.12	remote
5	Y	GigabitEthernet0/0/0/0.12	remote
6	Y	GigabitEthernet0/0/0/0.12	remote
7	Y	GigabitEthernet0/0/0/0.12	remote
8	Y	GigabitEthernet0/0/0/0.12	remote
9	Y	GigabitEthernet0/0/0/0.12	remote
10	Y	GigabitEthernet0/0/0/0.12	remote
11	Y	GigabitEthernet0/0/0/0.12	remote
12	Y	GigabitEthernet0/0/0/0.12	remote
13	Y	GigabitEthernet0/0/0/0.12	remote
14	Y	GigabitEthernet0/0/0/0.12	remote
15	Y	GigabitEthernet0/0/0/0.12	remote
16	Y	GigabitEthernet0/0/0/0.12	remote
17	Y	GigabitEthernet0/0/0/0.12	remote
18	Y	GigabitEthernet0/0/0/0.13	remote
19	Y	GigabitEthernet0/0/0/0.13	remote
20	Y	GigabitEthernet0/0/0/0.13	remote
21	Y	GigabitEthernet0/0/0/0.13	remote
22	Y	GigabitEthernet0/0/0/0.13	remote
23	Y	GigabitEthernet0/0/0/0.13	remote
24	Y	GigabitEthernet0/0/0/0.13	remote
25	Y	GigabitEthernet0/0/0/0.13	remote
26	Y	GigabitEthernet0/0/0/0.13	remote
27	Y	GigabitEthernet0/0/0/0.13	remote
28	Y	GigabitEthernet0/0/0/0.13	remote
29	Y	GigabitEthernet0/0/0/0.13	remote
30	Y	GigabitEthernet0/0/0/0.13	remote

Comme nous pouvons observer, la somme du poids normal représente la quantité de positions d'informations parasites assignées par la plate-forme, dans ce cas, nous peut ne jamais dépasser 32 positions d'informations parasites, selon la limite de cette plate-forme particulière. Le poids du chemin primaire (pm1) est toujours placé à 4294967295, qui est le poids maximum (2^{32}) - 1.

Détermination des valeurs de poids

Poids

Nous pouvons facilement calculer les poids avec le **poids de formule = coût mieux coûté/plus mauvais * 4294967295**. Par exemple, des poids pour le chemin 1 et le chemin 2 sont calculés ci-dessous :

Weight_path_1 = toujours réglé à 4294967295

Weight_path_2 = $151/201 * 4294967295 = 3226567470$

[Spoiler](#)

Note: La perte de précision peut se produire en calculant les valeurs, car nous faisons des calculs de virgule flottante, et nous devons installer des entiers dans la NERVURE et MENTIR.

Remarque: La perte de précision peut se produire en calculant les valeurs, car nous faisons des calculs de virgule flottante, et nous devons installer des entiers dans la NERVURE et MENTIR.

Poids normal

Comme nous avons mentionné, nous ne pouvons pas installer en valeurs de poids de table CEF dépassant la quantité de positions d'informations parasites par une plate-forme, due à ceci que nous devons normaliser les poids avant de les programmer dans le matériel. La plate-forme calcule les poids de normalisation selon le **poids normalisé par formule = (poids de poids de chemin/total) * taille maximum de position**. Basé sur notre exemple, nous pouvons calculer ceci comme suit :

$normalized_weight_1 = (4294967295 * 32)/(3226567396 + 4294967295) = 18$

$normalized_weight_2 = (3226567396 * 32)/(3226567396 + 4294967295) = 13$

[Spoiler](#)

Note: Quand le G.C.D est égal à 1, alors au-dessus de la méthode est utilisé, autrement si G.C.D = ! 1, normalisent alors le poids sera division du G.C.D en résultant par les valeurs de poids.

Remarque: Quand le G.C.D est égal à 1, alors au-dessus de la méthode est utilisé, autrement si G.C.D = ! 1, normalisent alors le poids sera division du G.C.D en résultant par les valeurs de poids.

Rapports manipulants de poids de CEF/chargement

Dans quelques scénarios nous pourrions vouloir déterminer quelle valeur métrique particulière de chemin nous devons configurer pour avoir un poids en résultant/répartition de charge. Nous pourrions déterminer la mesure appropriée de chemin en changeant le coût des liens et basé en fonction jusqu'à ce que nous atteignons ou rapprochions la valeur priée. Notez que non tous les poids que nous pourrions exiger sont exactement possibles, mais nous pouvons rapprocher la distribution priée.

Avant la continuation, prenez en considération les restrictions suivantes :

l'A.) non tous les poids/répartitions de charge sont exactement possible, mais nous pouvons

faire une approximation.

le B.) ne dépassent jamais les limites de position d'informations parasites. - Ceci signifie que la somme de tous les poids de chemin ne peut pas dépasser les positions d'informations parasites, si ceci se produit, alors le poids doit être normal. Signifiant que, en ajoutant vers le haut de tous les poids, nous ne dépassons pas la limite de position d'informations parasites.

c.) ASR 9000 et CRS-X ont une limite de position des informations parasites 64, XRv9000 ont une limite de position des informations parasites 32.

le D.) en utilisant pre-6.4.1, distribution de poids est différent, et le chemin avec le moins poids est toujours placé à un poids de 1 tandis que d'autres chemins sont des multiples de ce chemin qui signifie que ce peut être le supérieur à 1.

Exemple 1 : Rapport de poids/chargement de 26/5

Après la même topologie avant, nous voulons avoir une distribution de 26/5 poids entre les deux liens.

i.) au commencement, les coûts sont également placés sur tous les chemins (100 + 100 + 1) = 201.

ii.) si nous placerons la variance UCMP à la valeur maximale, pour considérer tous les prochain-sauts.

iii.) si nous vérifions la NERVURE, nous pouvons voir l'état par défaut où XR1 fait ECMP.

```
RP/0/RP0/CPU0:XR1#show cef 3.3.3.3/32 detail
```

```
3.3.3.3/32, version 27, internal 0x1000001 0x0 (ptr 0xd3ecb50) [1], 0x0 (0xd583610), 0x0 (0x0)
Updated Nov 11 23:08:25.290
remote adjacency to GigabitEthernet0/0/0/0.12
Prefix Len 32, traffic index 0, precedence n/a, priority 1
gateway array (0xd416218) reference count 2, flags 0x0, source rib (7), 0 backups
      [3 type 3 flags 0x8401 (0xd4bc6f8) ext 0x0 (0x0)]
LW-LDI[type=3, refc=1, ptr=0xd583610, sh-ldi=0xd4bc6f8]
gateway array update type-time 1 Nov 11 23:08:25.290
LDI Update time Nov 11 23:08:25.297
LW-LDI-TS Nov 11 23:08:25.297
  via 12.0.0.2/32, GigabitEthernet0/0/0/0.12, 4 dependencies, weight 4294967295, class 0 [flags
0x0]
    path-idx 0 NHID 0x0 [0xe14b1b0 0x0]
    next hop 12.0.0.2/32
    remote adjacency
  via 13.0.0.2/32, GigabitEthernet0/0/0/0.13, 4 dependencies, weight 4294967295, class 0 [flags
0x0]
    path-idx 1 NHID 0x0 [0xe14b128 0x0]
    next hop 13.0.0.2/32
    remote adjacency

Weight distribution:
slot 0, weight 4294967295, normalized_weight 1, class 0
slot 1, weight 4294967295, normalized_weight 1, class 0

Load distribution: 0 1 (refcount 3)
```


Hash	OK	Interface	Address
0	Y	GigabitEthernet0/0/0/0.12	remote
1	Y	GigabitEthernet0/0/0/0.13	remote

Pour cet exemple, nous utiliserons un cas où vous voulez les poids suivants :

W1 = 26 (meilleur coût primaire)

W2 = 5 (meilleur coût secondaire)

Nous devons prendre un chemin de tronçon, pour ce chemin, le coût si connu déjà, dans ce cas mettre en référence le chemin sera le chemin par l'intermédiaire de Gi0/0/0/0.12. Le chemin de tronçon pré-sera calculé avec le coût de bout en bout, la mesure de chemin et le poids exigé pour ce chemin sont :

i.) $X1+Y1+D1 = 100 + 100 + 1 = 201$. (Notez les variables reliées à chaque lien dans la topologie).

ii.) poids 1 = 26

iii.) poids 2 = 5

iv.) pm1 = 201 (chemin primaire de tronçon) ; Poids = 26

v.) pm2 = inconnu pourtant (chemin secondaire) ; Poids = 5

Calculer les poids.

Mesure de chemin de pm2 : $pm2 = (26/5) * 201 = 1045$

Détermination du coût du lien X2 sur XR1.

$X2 = pm2 - (x2 + y1 + d1)$

$1045 - (100 + 100 + 1) = 844$

Configurer le coût OSPF sur le lien X2.

```
router ospf 1
  ucmp variance 10000
  area 0
  !
  interface GigabitEthernet0/0/0/0.13
    cost 844
```

Vérifiant le poids/répartition de charge nous pouvons voir que les poids exigés ont été assignés convenablement dans le CEF comme nous avons prévu dans les calculs.

RP/0/RP0/CPU0:XR1#show cef 3.3.3.3/32 detail

```
3.3.3.3/32, version 37, internal 0x1000001 0x0 (ptr 0xd3ecce0) [1], 0x0 (0xd5835d8), 0x0 (0x0)
Updated Nov 11 23:17:47.945
remote adjacency to GigabitEthernet0/0/0/0.12
Prefix Len 32, traffic index 0, precedence n/a, priority 1
gateway array (0xd4163d8) reference count 1, flags 0x0, source rib (7), 0 backups
```


i.) $X1+Y1+D1 = 100 + 100 + 1 = 201$. (Notez les variables reliées à chaque lien dans la topologie).

ii.) poids 1 = 30

iii.) poids 2 = 1

iv.) pm1 = 201 (chemin primaire de tronçon) ; Poids = 30

v.) pm2 = inconnu pourtant (chemin secondaire) ; Poids = 1

Calculer les poids.

Mesure de chemin de pm2 : $pm2 = (30/1) * 201 = 6030$

Détermination du coût du lien X2 sur XR1.

$X2 = pm2 - (x2 + y1 + d1)$

$6030 - (100 + 100 + 1) = 5829$

Configurer le coût OSPF sur le lien X2.

```
router ospf 1
 ucmp variance 10000
 area 0
 !
 interface GigabitEthernet0/0/0/0.13
  cost 5829
```

Vérifiant le poids/répartition de charge nous pouvons voir que les poids exigés ont été assignés convenablement dans le CEF comme nous avons prévu dans les calculs.

```
RP/0/RP0/CPU0:XR1#show cef 3.3.3.3/32 detail
```

```
3.3.3.3/32, version 40, internal 0x1000001 0x0 (ptr 0xd3ecce0) [1], 0x0 (0xd5835d8), 0x0 (0x0)
 Updated Nov 11 23:31:58.207
 remote adjacency to GigabitEthernet0/0/0/0.12
 Prefix Len 32, traffic index 0, precedence n/a, priority 1
 gateway array (0xd416218) reference count 1, flags 0x0, source rib (7), 0 backups
 [2 type 3 flags 0x8401 (0xd4bc6f8) ext 0x0 (0x0)]
 LW-LDI[type=3, refc=1, ptr=0xd5835d8, sh-ldi=0xd4bc6f8]
 gateway array update type-time 1 Nov 11 23:31:58.207
 LDI Update time Nov 11 23:31:58.208
 LW-LDI-TS Nov 11 23:31:58.208
 via 12.0.0.2/32, GigabitEthernet0/0/0/0.12, 6 dependencies, weight 4294967295, class 0 [flags
 0x0]
 path-idx 0 NHID 0x0 [0xe14b1b0 0x0]
 next hop 12.0.0.2/32
 remote adjacency
 via 13.0.0.2/32, GigabitEthernet0/0/0/0.13, 6 dependencies, weight 140784018, class 0 [flags
 0x0]
 path-idx 1 NHID 0x0 [0xe14b128 0x0]
 next hop 13.0.0.2/32
 remote adjacency
```

Weight distribution:

