

Comprendre les répartitions de poids CEF dans le partage de charge à coût inégal

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Présentation UCMP](#)

[Paramètres de configuration initiaux](#)

[Poids/charge de la métrique](#)

[Détermination des écarts UCMP](#)

[Comprendre les poids](#)

[Détermination des valeurs de poids](#)

[Poids](#)

[Poids normalisé](#)

[Manipulation des ratios poids/charge CEF](#)

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

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

Introduction

Ce document décrit les aspects de Comprendre, configurer et vérifier les chemins multiples à coût inégal dans IOS-XR. Nous passons également en revue des exemples de manipulations de poids pour montrer comment la métrique de chemin vers une destination influence la charge sur une liaison.

Conditions préalables

Ce document n'a pas de conditions préalables.

Conditions requises

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

Components Used

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. If your network is live, make sure that you understand the potential impact of any command.

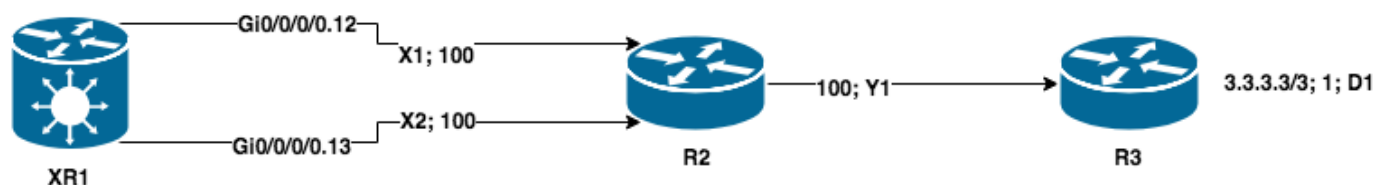
Présentation UCMP

L'équilibrage de charge UCMP (UnEqual Cost Multipath) permet d'équilibrer proportionnellement la charge du trafic sur plusieurs chemins, avec un coût différent. Généralement, les chemins à bande passante plus élevée ont des métriques IGP (Interior Gateway Protocol) plus basses configurées, de sorte qu'ils forment les chemins IGP les plus courts.

Avec l'équilibrage de charge UCMP activé, les protocoles peuvent utiliser des chemins de bande passante encore plus faibles ou des chemins de coût plus élevés pour le trafic, et peuvent installer ces chemins vers la base d'informations de transfert (FIB). Ces protocoles installent toujours plusieurs chemins vers la même destination dans FIB, mais chaque chemin sera associé à une 'métrique de charge/poids'. FIB utilise cette métrique de charge/poids pour déterminer la quantité de trafic à envoyer sur un chemin de bande passante plus élevée et la quantité de trafic à envoyer sur un chemin de bande passante plus faible.

Traditionnellement, le protocole EIGRP était le seul protocole IGP prenant en charge la fonctionnalité UCMP, mais dans IOS-XR, l'UCMP est pris en charge pour tous les protocoles IGP, le routage statique et le protocole BGP. Dans ce document, nous expliquerons la fonctionnalité UCMP en utilisant OSPF comme base de nos exemples, mais les informations ici s'appliquent également aux IS-IS et aux autres protocoles compatibles UCMP.

Schéma de topologie



Paramètres de configuration initiaux

```
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

```

Poids/charge de la métrique

Dans IOS-XR, lorsque nous installons plusieurs chemins vers une destination, une valeur de poids est attribuée à la destination qui indique la distribution de charge pour une liaison particulière. Cette valeur est inversement proportionnelle à la métrique de chemin vers la destination, plus le coût est élevé, plus le poids est faible. Cela permet à CEF d'effectuer intelligemment le partage de charge des liaisons lors du routage vers des destinations.

Lorsque des chemins ECMP sont installés, les valeurs de poids attribuées sont toujours définies sur 0 pour tous les chemins, ce qui signifie que le trafic est partagé de manière égale. Si nous vérifions CEF, nous pouvons confirmer que des poids de 0 ont été attribués 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 des écarts UCMP

Si nous voulons activer UCMP, commençons par définir le coût différemment sur XR1. Pour cela, nous définirons le coût comme suit :

```
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
!
!
```

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 envisager d'autres voies pour l'UCMP, nous devons déterminer si elles sont admissibles. IOS-XR utilise un critère de pourcentage pour IS-IS et OSPF, basé sur la commande de processus `ucmp variance <value>` du routeur. Les deux voies que nous avons sont :

métrique de chemin 1 (pm1) = 151

métrique de chemin 2 (pm2) = 201

Les sauts suivants sans boucle seront installés en fonction de l'UCMP $\leq (\text{Variance} * \text{Mesure du chemin principal}) / 100$.

Dans ce cas, 134 % des 151 chemins principaux doivent augmenter pour atteindre la pire métrique de chemin (pm2), ce qui donne lieu à 202. Il s'agit de la valeur de variance exacte 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.
```

Chauffeur

Note: La valeur de variance n'a aucune incidence sur les résultats de poids. Dans ce cas, une variance minimale de 134 ou une variance de 10000 (valeur maximale) aurait conduit aux mêmes résultats de pondération, au lieu de cela, les valeurs de coût sont celles qui influencent les pondérations résultantes, car celles-ci sont inversement proportionnelles les unes aux autres.

Note: La valeur de variance n'a aucune incidence sur les résultats de poids. Dans ce cas, une variance minimale de 134 ou une variance de 10000 (valeur maximale) aurait conduit aux mêmes résultats de pondération, au lieu de cela, les valeurs de coût sont celles qui influencent les pondérations résultantes, car celles-ci sont inversement proportionnelles les unes aux autres.

Comprendre les poids

Nous avons deux types différents de poids dans IOS-XR, **poids** et **poids normalisés**. L'utilisation de ces derniers est basée sur le nombre de compartiments de hachage pris en charge sur une plate-forme particulière, XRv9000 prend en charge 32 compartiments de hachage, ASR 9000 et CRS-X 64 compartiments de hachage respectivement. Cela signifie que, lorsque le routeur programme les valeurs de poids, la pondération ne peut pas dépasser la limite de groupement de hachage de la plate-forme particulière. Nous pouvons observer quels poids normalisés sont programmés en exécutant la commande **show cef <prefix> detail location <location>**. Sur la base des valeurs de coût définies, nous avons une distribution de charge de 18, 13, ce qui signifie que 31 segments de hachage ont été attribués (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 (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 le constater, la somme du poids normalisé représente la quantité de compartiments de hachage attribuée par la plate-forme, dans ce cas, nous ne pouvons jamais dépasser 32 compartiments de hachage, conformément à la limite de cette plate-forme particulière. Le poids du chemin principal (pm1) est toujours défini sur 4294967295, soit le poids maximal ($2^{32} - 1$).

Détermination des valeurs de poids

Poids

Nous pouvons facilement calculer les poids avec la formule **poids = meilleur coût / pire coût * 4294967295**. Par exemple, les poids des chemins 1 et 2 sont calculés ci-dessous :

Weight_path_1 = toujours défini sur 4294967295

Weight_path_2 = 151 / 201 * 4294967295 = 3226567470

Chauffeur

Note: Une perte de précision peut se produire lors du calcul des valeurs, comme nous faisons des calculs à virgule flottante, et nous devons installer des entiers dans RIB et FIB.

Note: Une perte de précision peut se produire lors du calcul des valeurs, comme nous faisons des calculs à virgule flottante, et nous devons installer des entiers dans RIB et FIB.

Poids normalisé

Comme nous l'avons mentionné, nous ne pouvons pas installer dans la table CEF des valeurs de poids supérieures à la quantité de compartiments de hachage par une plate-forme, en raison de cela nous devons normaliser les poids avant de les programmer dans le matériel. La plate-forme calcule les poids normalisés selon la formule **Poids normalisé = (Poids du chemin/Poids total) * Taille maximale du compartiment**. D'après notre exemple, nous pouvons calculer ceci comme suit :

$\text{poids_normalisé_1} = (4294967295 * 32) / (3226567396 + 4294967295) = 18$

$\text{poids_normalisé_2} = (3226567396 * 32) / (3226567396 + 4294967295) = 13$

Chauffeur

Note: Lorsque G.C.D est égal à 1, alors la méthode ci-dessus est utilisée, sinon si G.C.D ≠ 1, puis normaliser le poids sera la division de la masse résultante G.C.D par les valeurs de poids.

Note: Lorsque G.C.D est égal à 1, alors la méthode ci-dessus est utilisée, sinon si G.C.D ≠ 1, puis normaliser le poids sera la division de la DDC résultante par les valeurs de poids.

Manipulation des ratios poids/charge CEF

Dans certains scénarios, nous pourrions vouloir déterminer la valeur de métrique de chemin que nous devons configurer pour obtenir une distribution de poids/charge résultante. Nous pourrions déterminer la métrique de chemin appropriée en modifiant le coût des liaisons et en nous basant sur jusqu'à ce que nous atteignons ou approchions la valeur requise. Notez que tous les poids que nous pourrions exiger ne sont pas exactement possibles, mais nous pouvons approximer la distribution requise.

Avant de continuer, tenez compte des restrictions suivantes :

- a.) Toutes les distributions de poids/charge ne sont pas exactement possibles, mais nous pouvons faire une approximation.
- b.) Ne jamais dépasser les limites du seau de hachage. - Cela signifie que la somme de tous les poids de chemin ne peut pas dépasser les compartiments de hachage, si cela se produit, alors le poids doit être normalisé. Ce qui signifie que, lors de l'addition de tous les poids, nous ne dépassons pas la limite du compartiment de hachage.
- c.) L'ASR 9000 et le CRS-X ont une limite de regroupement de 64 hachages, le XRv9000 a une limite de regroupement de 32 hachages.

d.) Lors de l'utilisation de pre-6.4.1, la distribution de poids est différente, et le chemin avec le poids le plus faible est toujours défini sur un poids de 1 tandis que les autres chemins sont des multiples de ce chemin, ce qui signifie qu'il peut être supérieur à 1.

Exemple 1 : Rapport poids/charge de 26/5

En suivant la même topologie précédemment, nous voulons disposer d'une distribution de poids 26/5 entre les deux liaisons.

- i.) Initialement, les coûts sont également fixés sur tous les chemins ($100 + 100 + 1$) = 201.
- ii) Si nous allons définir la variance UCMP sur la valeur maximale, prenez en compte tous les sauts suivants.
- iii) Si nous vérifions le RIB, nous pouvons voir l'état par défaut où XR1 effectue 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

Dans cet exemple, nous allons utiliser un cas où vous souhaitez obtenir les poids suivants :

W1 = 26 (meilleur coût principal)

W2 = 5 (meilleur coût secondaire)

Nous devons prendre un chemin de jambe, pour ce chemin, le coût devrait déjà savoir, dans ce cas le chemin de référence sera le chemin via Gi0/0/0/0.12. Le chemin de la jambe sera précalculé avec le coût de bout en bout, la métrique du chemin et le poids requis pour ce chemin

sont les suivants :

i.) $X1+Y1+D1 = 100 + 100 + 1 = 201$. (Notez les variables associées à chaque liaison de la topologie).

ii) Poids 1 = 26

iii) Poids 2 = 5

iv.) $pm1 = 201$ (trajectoire de la jambe principale); Poids = 26

v.) $pm2 = \text{inconnu}$ (chemin secondaire); Poids = 5

Calcul des poids.

Mesure de chemin de $pm2$: $pm2 = (26/5) * 201 = 1\ 045$

Détermination du coût de la liaison X2 sur XR1.

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

$1\ 045 - (100+100+1) = 844$

Configuration du coût OSPF sur la liaison X2.

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

En vérifiant la répartition poids/charge, nous pouvons voir que les poids requis ont été attribués de façon appropriée dans le CEF comme nous l'avions 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
  [2 type 3 flags 0x8401 (0xd4bc7b8) ext 0x0 (0x0)]
LW-LDI[type=3, refc=1, ptr=0xd5835d8, sh-ldi=0xd4bc7b8]
gateway array update type-time 1 Nov 11 23:17:47.945
LDI Update time Nov 11 23:17:47.956
LW-LDI-TS Nov 11 23:17:47.956
  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 913532538, 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 26, class 0

slot 1, weight 913532538, normalized_weight 5, class 0

Load distribution: 0 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	GigabitEthernet0/0/0/0.12	remote
19	Y	GigabitEthernet0/0/0/0.12	remote
20	Y	GigabitEthernet0/0/0/0.12	remote
21	Y	GigabitEthernet0/0/0/0.12	remote
22	Y	GigabitEthernet0/0/0/0.12	remote
23	Y	GigabitEthernet0/0/0/0.12	remote
24	Y	GigabitEthernet0/0/0/0.12	remote
25	Y	GigabitEthernet0/0/0/0.12	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

Exemple 2 : Rapport poids/charge de 30/1

Comme précédemment, nous avons un coût par défaut de 100 sur les deux interfaces XR1.

W1 = 30 (meilleur coût principal)

W2 = 1 (meilleur coût secondaire)

i.) $X1+Y1+D1 = 100 + 100 + 1 = 201$. (Notez les variables associées à chaque liaison de la topologie).

ii) Poids 1 = 30

iii) Poids 2 = 1

iv.) pm1 = 201 (trajectoire de la jambe principale); Poids = 30

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

Calcul des poids.

Mesure de chemin de pm2 : pm2 = (30/1) * 201 = 6 030

Détermination du coût de la liaison X2 sur XR1.

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

$6\ 030 - (100 + 100 + 1) = 5\ 829$

Configuration du coût OSPF sur la liaison X2.

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

En vérifiant la répartition poids/charge, nous pouvons voir que les poids requis ont été attribués de façon appropriée dans le CEF comme nous l'avions 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:
 slot 0, weight 4294967295, normalized_weight 30, class 0
 slot 1, weight 140784018, normalized_weight 1, class 0

Load distribution: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	GigabitEthernet0/0/0/0.12	remote
19	Y	GigabitEthernet0/0/0/0.12	remote
20	Y	GigabitEthernet0/0/0/0.12	remote
21	Y	GigabitEthernet0/0/0/0.12	remote
22	Y	GigabitEthernet0/0/0/0.12	remote
23	Y	GigabitEthernet0/0/0/0.12	remote
24	Y	GigabitEthernet0/0/0/0.12	remote
25	Y	GigabitEthernet0/0/0/0.12	remote
26	Y	GigabitEthernet0/0/0/0.12	remote
27	Y	GigabitEthernet0/0/0/0.12	remote
28	Y	GigabitEthernet0/0/0/0.12	remote
29	Y	GigabitEthernet0/0/0/0.12	remote
30	Y	GigabitEthernet0/0/0/0.13	remote