

Introducción a las distribuciones de peso de CEF en la distribución de carga de costo desigual

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Descripción general de UCMP](#)

[Configuraciones iniciales](#)

[Pesos/carga de métricas](#)

[Determinación de la variación de UCMP](#)

[Comprensión de los pesos](#)

[Determinación de los valores de peso](#)

[Peso](#)

[Peso normalizado](#)

[Manipulación de las relaciones de carga/peso CEF](#)

[Ejemplo 1: Relación peso/carga de 26/5](#)

[Ejemplo 2: Relación peso/carga de 30/1](#)

Introducción

Este documento describe los aspectos de Comprensión, Configuración y Verificación de Trayectoria Múltiple de Coste Desigual en IOS-XR. Asimismo, pasamos por ejemplos de manipulaciones de peso para mostrar cómo la métrica de trayectoria a un destino influye en la carga en un link.

Prerequisites

Este documento no tiene requisitos previos.

Requirements

Los siguientes ejemplos se basan en IOS-XR 6.4.1.

Componentes Utilizados

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.

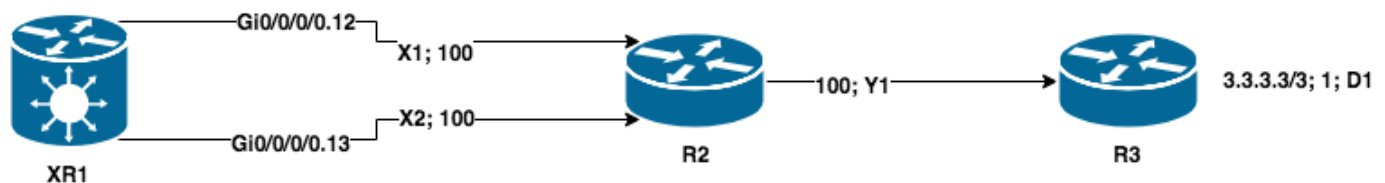
Descripción general de UCMP

El equilibrio de carga de múltiples rutas de costo desigual (UCMP) proporciona la capacidad de equilibrar la carga del tráfico proporcionalmente en varias rutas, con un coste diferente. Por lo general, las rutas de mayor ancho de banda tienen métricas de protocolo de gateway interior (IGP) más bajas configuradas, de modo que forman las rutas IGP más cortas.

Con el balanceo de carga UCMP habilitado, los protocolos pueden utilizar rutas de ancho de banda incluso más bajas o rutas de costes más altas para el tráfico, y pueden instalar estas rutas en la base de información de reenvío (FIB). Estos protocolos aún instalan varias trayectorias al mismo destino en FIB, pero cada trayectoria tendrá una 'métrica de carga/peso' asociada. FIB utiliza esta métrica/peso de carga para decidir la cantidad de tráfico que se debe enviar en una trayectoria de ancho de banda más alto y la cantidad de tráfico que se debe enviar en una trayectoria de ancho de banda más bajo.

Tradicionalmente, EIGRP ha sido el único IGP que soporta la función UCMP, pero en IOS-XR UCMP se soporta para todos los IGP, el ruteo estático y BGP. En este documento, explicaremos la función UCMP usando OSPF como base de nuestros ejemplos, pero la información aquí se aplica también a IS-IS y otros protocolos compatibles con UCMP.

Diagrama de topología



Configuraciones iniciales

```
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

```

Pesos/carga de métricas

En IOS-XR, cuando instalamos varias trayectorias a un destino, al destino se le asigna un valor de peso que indica la distribución de carga para un link determinado. Este valor es inversamente proporcional a la métrica de trayectoria al destino, cuanto mayor sea el costo, menor será el peso asignado. Esto permite a CEF realizar de forma inteligente el uso compartido de carga de links cuando se dirige a destinos.

Cuando se instalan las trayectorias ECMP, los valores de peso asignados siempre se configuran en 0 para todas las trayectorias, lo que significa que el tráfico se comparte por igual con la carga. Si verificamos CEF, podemos confirmar que se han asignado pesos de 0 para cada trayectoria.

```
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 |

Determinación de la variación de UCMP

Si queremos habilitar UCMP, empecemos por establecer los costes de forma diferente en XR1, para esto fijaremos los costes de la siguiente manera:

```

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.

```

Para considerar otras trayectorias para UCMP necesitamos determinar si son elegibles. IOS-XR utiliza un criterio de porcentaje para IS-IS y OSPF, esto se basa en el comando **ucmp varianza <valor>** router process. Los dos caminos que tenemos son:

path metric 1 (pm1) = 151

path metric 2 (pm2) = 201

Los saltos siguientes sin loops se instalarán en función de UCMP \leq (Varianza * Métrica de trayectoria principal) / 100.

En este caso, el 134% de 151, lo que resulta en 202, tiene que crecer la ruta principal para alcanzar la métrica de peor trayectoria (pm2). Este es el valor exacto de la varianza que necesitamos configurar para que la trayectoria sea elegible.

```
!  
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](#)

Nota: El valor de la varianza no tiene ningún impacto en los resultados del peso. En este caso, una varianza mínima de 134 o una varianza de 10000 (valor máximo) habrían producido los mismos resultados de peso, en cambio, los valores de costo son los que influyen en los pesos resultantes, ya que son inversamente proporcionales entre sí.

Nota: El valor de la varianza no tiene ningún impacto en los resultados del peso. En este caso, una varianza mínima de 134 o una varianza de 10000 (valor máximo) habrían producido los mismos resultados de peso, en cambio, los valores de costo son los que influyen en los pesos resultantes, ya que son inversamente proporcionales entre sí.

Comprensión de los pesos

Tenemos dos tipos diferentes de pesos en IOS-XR, **peso** y **pesos normalizados**. El uso de estos se basa en la cantidad de cubetas de troceo admitidas en una plataforma en particular, el XRv9000 admite 32 cubetas de troceo, ASR 9000 y CRS-X admiten 64 cubetas respectivamente. Esto significa que, cuando el router programa los valores de peso, la ponderación no puede exceder el límite de la cubeta hash de la plataforma en particular. Podemos observar qué pesos normalizados se programan ejecutando el comando **show cef <prefix> detail location <location>**. Basándonos en los valores de costo establecidos, tenemos una distribución de carga de 18, 13, lo que significa que se han asignado 31 cubetas (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 |

Como podemos observar, la suma del peso normalizado representa la cantidad de cubos de troceo asignados por la plataforma, en este caso, nunca podemos exceder los 32 cubos de troceo, según el límite de esta plataforma en particular. El peso de la ruta principal (pm1) siempre se establece en 4294967295, que es el peso máximo ($2^{32} - 1$).

Determinación de los valores de peso

Peso

Podemos calcular fácilmente los pesos con la fórmula **peso = mejor costo / peor costo * 4294967295**. Por ejemplo, los pesos para la trayectoria 1 y la trayectoria 2 se calculan a continuación:

Weight_path_1 = siempre establecido en 4294967295

Weight_path_2 = $151 / 201 * 4294967295 = 3226567470$

[Spoiler](#)

Nota: La pérdida de precisión puede ocurrir al calcular los valores, como estamos haciendo cálculos de punto flotante, y debemos instalar enteros en RIB y FIB.

Nota: La pérdida de precisión puede ocurrir al calcular los valores, como estamos haciendo cálculos de punto flotante, y debemos instalar enteros en RIB y FIB.

Peso normalizado

Como hemos mencionado, no podemos instalar en la tabla CEF valores de peso que excedan la cantidad de cubetas hash por una plataforma, debido a esto necesitamos normalizar los pesos antes de programarlos en hardware. La plataforma calcula los pesos normalizados según la fórmula **Peso normalizado = (peso de trayecto/peso total) * Tamaño máximo de depósito**. Basándonos en nuestro ejemplo, podemos calcular esto de la siguiente manera:

$\text{peso_normal_1} = (4294967295 * 32) / (3226567396 + 4294967295) = 18$

$\text{peso_normal_2} = (3226567396 * 32) / (3226567396 + 4294967295) = 13$

[Spoiler](#)

Nota: Cuando G.C.D es igual a 1, se utiliza el método anterior, de lo contrario si G.C.D \neq 1, entonces normalizar el peso será la división del G.C.D resultante por los valores de peso.

Nota: Cuando G.C.D es igual a 1, se utiliza el método anterior, de lo contrario si G.C.D \neq 1, el peso normal será la división del G.C.D resultante por los valores de peso.

Manipulación de las relaciones de carga/peso CEF

En algunos escenarios, es posible que queramos determinar qué valor de métrica de trayectoria particular necesitamos configurar para tener una distribución de peso/carga resultante. Podríamos determinar la métrica de trayectoria apropiada cambiando el costo de los links y basándonos en hasta que alcancemos o aproximemos el valor requerido. Tenga en cuenta que no todos los pesos que podamos necesitar son exactamente posibles, pero podemos aproximar la distribución requerida.

Antes de continuar, tenga en cuenta las siguientes restricciones:

r.) No todas las distribuciones de peso/carga son exactamente posibles, pero podemos hacer una aproximación.

b) Nunca exceda los límites de la cubeta hash. - Esto significa que la suma de todos los pesos de trayectoria no puede exceder las cubetas de troceo, si esto sucede, el peso debe ser normalizado. Lo que significa que, al añadir todos los pesos, no superamos el límite de la cubeta hash.

c.) ASR 9000 y CRS-X tienen un límite de 64 cubeta hash, XRv9000 tienen un límite de cubeta de troceo de 32.

d.) Cuando se utiliza pre-6.4.1, la distribución de peso es diferente, y la trayectoria con menor

peso siempre se fija en un peso de 1 mientras que otras trayectorias son múltiplos de esta trayectoria, lo que significa que puede ser mayor que 1.

Ejemplo 1: Relación peso/carga de 26/5

Siguiendo la misma topología antes, queremos tener una distribución de peso 26/5 entre los dos links.

- i) Inicialmente, los costes se fijan por igual en todas las trayectorias ($100 + 100 + 1 = 201$).
- ii) Si estableceremos la varianza de UCMP en el valor máximo, consideraremos todos los saltos siguientes.
- iii) Si verificamos el RIB, podemos ver el estado predeterminado en el que XR1 está realizando 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
```

Para este ejemplo, utilizaremos un caso en el que desee los siguientes pesos:

W1 = 26 (el mejor coste principal)

W2 = 5 (mejor coste secundario)

Necesitamos tomar un trayecto de tramas, para este trayecto, el costo ya debería conocerse, en este caso la trayectoria de referencia será la vía Gi0/0/0.12. La trayectoria del tramo se calculará previamente con el costo de extremo a extremo, la métrica de trayectoria y el peso necesarios

para esta trayectoria son:

i) $X1+Y1+D1 = 100 + 100 + 1 = 201$. (Observe las variables asociadas a cada link en la topología).

ii) Peso 1 = 26

iii) Peso 2 = 5

iv) $pm1 = 201$ (trayecto principal del tramo); Peso = 26

v) $pm2 =$ desconocido aún (trayecto secundario); Peso = 5

Computar los pesos.

Métrica de trayecto de $pm2$: $pm2 = (26/5) * 201 = 1045$

Determinación del costo del link X2 en XR1.

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

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

Configuración del costo OSPF en el link X2.

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

Verificando la distribución de peso/carga podemos ver que los pesos requeridos se han asignado apropiadamente en CEF como se predijo en los cálculos.

```
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 |

Ejemplo 2: Relación peso/carga de 30/1

Igual que antes, el costo predeterminado es 100 en ambas interfaces XR1.

W1 = 30 (el mejor coste principal)

W2 = 1 (mejor coste secundario)

i) $X1+Y1+D1 = 100 + 100 + 1 = 201$. (Observe las variables asociadas a cada link en la topología).

ii) Peso 1 = 30

iii) Peso 2 = 1

iv) pm1 = 201 (trayecto principal del tramo); Peso = 30

v) pm2 = desconocido aún (trayecto secundario); Peso = 1

Computar los pesos.

Métrica de trayecto de pm2: $pm2 = (30/1) * 201 = 6030$

Determinación del costo del link X2 en XR1.

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

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

Configuración del costo OSPF en el link X2.

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

Verificando la distribución de peso/carga podemos ver que los pesos requeridos se han asignado apropiadamente en CEF como se predijo en los cálculos.

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 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 |