

# Entendendo as distribuições de peso de CEF no compartilhamento de carga de custo desigual

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Visão geral do UCMP](#)

[Configurações iniciais](#)

[Peso/carga da métrica](#)

[Determinação da variação de UCMP](#)

[Compreendendo os pesos](#)

[Determinando Valores de Peso](#)

[Peso](#)

[Peso normalizado](#)

[Manipulação do peso/carga do CEF](#)

[Exemplo 1: Relação peso/carga de 26/5](#)

[Exemplo 2: Relação peso/carga de 30/1](#)

## Introduction

Este documento descreve os aspectos de Entendendo, Configurando e Verificando o Multipath de Custo Desigual no IOS-XR. Também analisamos exemplos de manipulações de peso para mostrar como a métrica do caminho para um destino influencia a carga em um link.

## Prerequisites

Este documento não tem pré-requisitos.

## Requirements

Os exemplos abaixo são baseados no 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.

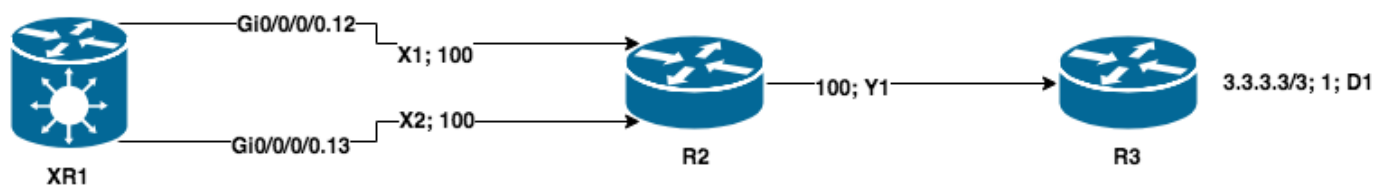
## Visão geral do UCMP

O balanceamento de carga de multipath de custo desigual (UCMP) fornece a capacidade de balancear carga de tráfego proporcionalmente em vários caminhos, com custos diferentes. Geralmente, os caminhos de largura de banda mais alta têm métricas de Interior Gateway Protocol (IGP) mais baixas configuradas, de modo que formem os caminhos IGP mais curtos.

Com o balanceamento de carga do UCMP ativado, os protocolos podem usar caminhos de largura de banda ainda menores ou caminhos de custo mais alto para o tráfego e podem instalar esses caminhos para a base de informações de encaminhamento (FIB). Esses protocolos ainda instalam vários caminhos para o mesmo destino em FIB, mas cada caminho terá uma "métrica/peso de carga" associada a ele. O FIB usa essa métrica/peso de carga para decidir a quantidade de tráfego que precisa ser enviada em um caminho de largura de banda mais alta e a quantidade de tráfego que precisa ser enviada em um caminho de largura de banda mais baixa.

Tradicionalmente, o EIGRP tem sido o único IGP que suporta o recurso UCMP, mas no IOS-XR UCMP é suportado para todos os IGPs, roteamento estático e BGP. Neste documento, explicaremos o recurso UCMP usando o OSPF como base de nossos exemplos, mas as informações aqui também se aplicam ao IS-IS e a outros protocolos compatíveis com UCMP.

## Diagrama de topologia



## Configurações iniciais

```
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

```

## Peso/carga da métrica

No IOS-XR, quando instalamos vários caminhos para um destino, o destino recebe um valor de peso que indica a distribuição de carga para um link específico. Esse valor é inversamente proporcional à métrica do caminho ao destino, quanto maior o custo, menor o peso atribuído. Isso permite que o CEF execute de forma inteligente o compartilhamento de carga de links ao rotear para os destinos.

Quando os caminhos ECMP são instalados, os valores de peso atribuídos são sempre definidos como 0 para todos os caminhos, isso significa que o tráfego é compartilhado com carga igualmente. Se verificarmos o CEF, podemos confirmar que foram atribuídos pesos de 0 para cada caminho.

```
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

## Determinação da variação de UCMP

Se quisermos habilitar o UCMP, vamos começar definindo o custo de forma diferente em XR1, para isso, definiremos o custo como abaixo:

```
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 outros caminhos para UCMP, precisamos determinar se eles são qualificados. O IOS-XR usa um critério de porcentagem para IS-IS e OSPF, baseado no comando de processo do roteador **ucmp variance <value>**. Os dois caminhos que temos são:

métrica de caminho 1 (pm1) = 151

métrica de caminho 2 (pm2) = 201

Os próximos saltos sem loop serão instalados com base em  $UCMP \leq (Variance * Primary Path metric) / 100$ .

Quanto caminho principal precisa crescer para alcançar a pior métrica de caminho (pm2) nesse caso é 134% de 151, o que resulta em 202. Esse é o valor exato da variação que precisamos configurar para tornar o caminho elegível.

```
!  
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:** O valor da variação não tem nenhum impacto nos resultados do peso. Neste caso, uma variância mínima de 134 ou uma variância de 10000 (valor máximo) levaria aos mesmos resultados ponderais, em vez disso, os valores dos custos são os que influenciam os pesos resultantes, uma vez que estes são inversamente proporcionais um ao outro.

Note: O valor da variação não tem nenhum impacto nos resultados do peso. Neste caso, uma variância mínima de 134 ou uma variância de 10000 (valor máximo) levaria aos mesmos resultados ponderais, em vez disso, os valores dos custos são os que influenciam os pesos resultantes, uma vez que estes são inversamente proporcionais um ao outro.

## Compreendendo os pesos

Temos dois tipos diferentes de pesos no IOS-XR, **peso** e **pesos normalizados**. O uso desses dados é baseado no número de hash buckets suportados em uma plataforma específica, o XRv9000 suporta 32 hash buckets, ASR 9000 e o CRS-X suportam 64 hash buckets, respectivamente. Isso significa que, quando o roteador programa os valores de peso, a ponderação não pode exceder o limite de hash bucket da plataforma específica. Podemos observar quais pesos normalizados estão programados emitindo o comando **show cef <prefix> detail location <location>**. Com base nos valores de custo definidos, temos uma distribuição de carga de 18, 13, o que significa que 31 hash buckets foram atribuídos (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, a soma do peso normalizado representa a quantidade de hash buckets atribuídos pela plataforma, neste caso, nunca podemos exceder 32 hash buckets, de acordo com o limite desta plataforma específica. O peso do caminho primário (pm1) é sempre definido como 4294967295, que é o peso máximo ( $2^{32} - 1$ ).

## Determinando Valores de Peso

### Peso

Podemos facilmente calcular os pesos com a fórmula **peso = melhor custo/pior custo \* 4294967295**. Por exemplo, pesos para o caminho 1 e o caminho 2 são calculados abaixo:

Weight\_path\_1 = sempre definido como 4294967295

$\text{Weight\_path\_2} = 151 / 201 * 4294967295 = 3226567470$

### [Spoiler](#)

**Note:** A perda de precisão pode ocorrer ao calcular os valores, à medida que fazemos cálculos de ponto flutuante, e devemos instalar números inteiros em RIB e FIB.

Note: A perda de precisão pode ocorrer ao calcular os valores, à medida que fazemos cálculos de ponto flutuante, e devemos instalar números inteiros em RIB e FIB.

## Peso normalizado

Como mencionamos, não podemos instalar na tabela CEF valores de peso que excedem a quantidade de hash buckets por uma plataforma, pois precisamos normalizar os pesos antes de programá-los no hardware. A plataforma calcula os pesos normalizáveis de acordo com a fórmula **Peso Normalizado = (Peso do caminho/Peso total) \* Tamanho máximo do bucket**. Com base em nosso exemplo, podemos calcular isso da seguinte forma:

$\text{normalized\_weight\_1} = (4294967295 * 32) / (3226567396 + 4294967295) = 18$

$\text{normal\_weight\_2} = (3226567396 * 32) / (3226567396 + 4294967295) = 13$

### [Spoiler](#)

**Note:** Quando G.C.D é igual a 1, então o método acima é usado, caso contrário, se  $\text{G.C.D} \neq 1$ , então normalizar o peso será a divisão do G.C.D resultante pelos valores de peso.

Note: Quando G.C.D é igual a 1, então o método acima é usado, caso contrário, se  $\text{G.C.D} \neq 1$ , então normalizar o peso será a divisão do G.C.D resultante pelos valores de peso.

## Manipulação do peso/carga do CEF

Em alguns cenários, podemos querer determinar qual valor de métrica de caminho específico precisamos configurar para ter uma distribuição de peso/carga resultante. Poderíamos determinar a métrica de caminho apropriada alterando o custo dos links e com base em até atingirmos ou aproximarmos o valor necessário. Observe que nem todos os pesos necessários são exatamente possíveis, mas podemos aproximar a distribuição necessária.

Antes de continuar, tenha em conta as seguintes restrições:

- r.) Nem todas as distribuições de peso/carga são exatamente possíveis, mas podemos fazer uma aproximação.
- b.) Nunca ultrapasse os limites de hash bucket. - Isso significa que a soma de todos os pesos de caminho não pode exceder os hash buckets, caso isso aconteça, o peso deve ser normalizado. Significa que, ao somar todos os pesos, não excedemos o limite de hash bucket.
- c.) O ASR 9000 e o CRS-X têm um limite de 64 hash bucket, o XRv9000 tem um limite de 32 hash bucket.
- d.) Ao usar o pré-6.4.1, a distribuição de peso é diferente, e o caminho com o menor peso é sempre definido com um peso de 1, enquanto outros caminhos são múltiplos desse caminho, o que significa que ele pode ser maior que 1.

## Exemplo 1: Relação peso/carga de 26/5

Seguindo a mesma topologia antes, queremos ter uma distribuição de peso 26/5 entre os dois links.

- i.) Inicialmente, os custos são igualmente definidos em todos os caminhos ( $100 + 100 + 1$ ) = 201.
- ii.) Se definirmos a variação de UCMP como o valor máximo, considere todos os próximos saltos.
- iii.) Se verificarmos o RIB, podemos ver o estado padrão onde XR1 está fazendo 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 exemplo, usaremos um caso em que você deseja os seguintes pesos:

W1 = 26 (custo principal)

W2 = 5 (melhor custo secundário)

Precisamos seguir um caminho de perna, para esse caminho, o custo já deve ser conhecido. Nesse caso, o caminho de referência será o caminho através de Gi0/0/0/0.12. O caminho da perna será pré-computado com o custo de ponta a ponta, a métrica do caminho e o peso necessários para esse caminho são:

- i.)  $X1+Y1+D1 = 100 + 100 + 1 = 201$ . (Observe as variáveis anexadas a cada link na



topologia).

ii.) Peso 1 = 26

iii.) Peso 2 = 5

iv.) pm1 = 201 (caminho da perna primária); Peso = 26

v.) pm2 = ainda desconhecido (caminho secundário); Peso = 5

Calculando os pesos.

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

Determinando o custo do link X2 em XR1.

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

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

Configurando o custo do OSPF no link X2.

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

Verificando a distribuição de peso/carga, podemos ver que os pesos necessários foram atribuídos adequadamente no CEF, como previmos nos 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
```





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