

Verständnis von CEF-Gewichtsverteilungen in der Lastverteilung zu gleichen Kosten

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[UCMP - Übersicht](#)

[Erstkonfiguration](#)

[Metrische Gewichtung/Last](#)

[Ermittlung der UCMP-Varianz](#)

[Gewichtungen verstehen](#)

[Bestimmen der Gewichtungswerte](#)

[Gewicht](#)

[Normalgewicht](#)

[Bearbeiten der CEF-Gewicht-/Lastverhältnisse](#)

[Beispiel 1: Gewicht/Lastverhältnis von 26/5](#)

[Beispiel 2: Gewicht/Lastverhältnis von 30/1](#)

Einführung

In diesem Dokument werden die Aspekte des Verständnisses, der Konfiguration und der Überprüfung von ungewöhnlichen Kosten-Multipath in IOS-XR beschrieben. Außerdem zeigen wir anhand von Beispielen für Gewichtsmanipulationen, wie die Pfadmetrik zu einem Ziel die Last einer Verbindung beeinflusst.

Voraussetzungen

Dieses Dokument enthält keine Voraussetzungen.

Anforderungen

Die folgenden Beispiele basieren auf IOS-XR 6.4.1.

Verwendete Komponenten

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

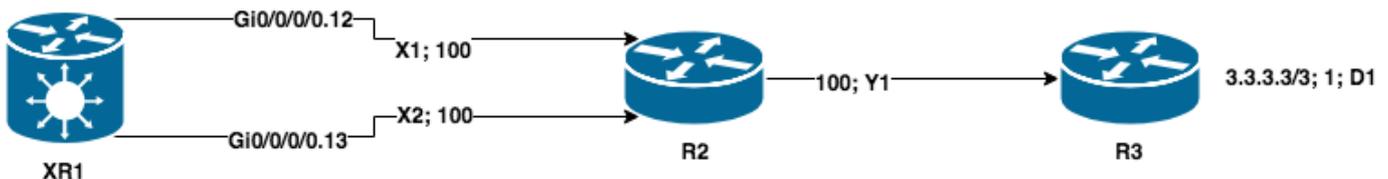
UCMP - Übersicht

Der Lastenausgleich bei ungleichen Kosten über mehrere Pfade hinweg bietet die Möglichkeit, den Datenverkehr proportional zum Lastenausgleich bei unterschiedlichen Kosten auszugleichen. Im Allgemeinen sind für Pfade mit höherer Bandbreite niedrigere IGP-Metriken (Interior Gateway Protocol) konfiguriert, sodass sie die kürzesten IGP-Pfade bilden.

Wenn der UCMP-Lastenausgleich aktiviert ist, können Protokolle für den Datenverkehr noch niedrigere Bandbreitenpfade oder kostenintensivere Pfade verwenden und diese Pfade zur Forwarding Information Base (FIB) installieren. Diese Protokolle installieren immer noch mehrere Pfade zum gleichen Ziel in FIB, aber jedem Pfad wird eine 'Lastmetrik/Gewicht' zugeordnet. Die FIB verwendet diese Lastmetrik/dieses Gewicht, um die Menge des Datenverkehrs zu bestimmen, der auf einem höheren Bandbreitenpfad gesendet werden muss, und die Menge des Datenverkehrs, der auf einem niedrigeren Bandbreitenpfad gesendet werden muss.

In der Vergangenheit war EIGRP das einzige IGP, das die UCMP-Funktion unterstützt. In IOS-XR wird jedoch UCMP für alle IGPs, statisches Routing und BGP unterstützt. In diesem Dokument wird die UCMP-Funktion erläutert, die OSPF als Grundlage für unsere Beispiele verwendet. Die Informationen hier gelten jedoch auch für IS-IS und andere UCMP-fähige Protokolle.

Topologiediagramm



Erstkonfiguration

```
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

```

Metrische Gewichtung/Last

Wenn in IOS-XR mehrere Pfade zu einem Ziel installiert werden, wird dem Ziel ein Gewichtswert zugewiesen, der die Lastverteilung für eine bestimmte Verbindung angibt. Dieser Wert ist umgekehrt proportional zur Pfadmetrik zum Ziel, je höher die Kosten sind, desto geringer wird das Gewicht zugewiesen. So kann CEF beim Routing zu Zielen die Lastverteilung von Links intelligent durchführen.

Wenn ECMP-Pfade installiert werden, werden die zugewiesenen Gewichtswerte für alle Pfade immer auf 0 gesetzt, d. h. der Datenverkehr wird gleichmäßig verteilt. Wenn wir CEF prüfen, können wir bestätigen, dass für jeden Pfad eine Gewichtung von 0 zugewiesen wurde.

```
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

Ermittlung der UCMP-Varianz

Wenn wir UCMP aktivieren möchten, legen wir zunächst die Kosten für XR1 anders fest. Hierzu werden wir die Kosten wie folgt festlegen:

```
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.
```

Um andere Pfade für UCMP in Betracht zu ziehen, müssen wir ermitteln, ob diese qualifiziert sind. IOS-XR verwendet prozentuale Kriterien für IS-IS und OSPF. Diese basieren auf dem Befehl **ucmp variance <value>** router process. Wir haben zwei Wege:

path metric 1 (pm1) = 151

path metric 2 (pm2) = 201

Loop free next-hops werden basierend auf UCMP $\leq (\text{Varianz} * \text{Primäre Pfadmetrik}) / 100$ installiert.

Der primäre Pfad muss wachsen, um die schlechteste Pfadmetrik zu erreichen (pm2), in diesem

Fall beträgt 134 Prozent von 151, was 2012 ergibt. Dies ist der genaue Varianzwert, den wir konfigurieren müssen, um den Pfad qualifiziert zu machen.

```
!  
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](#)

Hinweis: Der Varianzwert hat keine Auswirkungen auf die Gewichtsergebnisse. In diesem Fall hätte eine Mindestvarianz von 134 oder eine Varianz von 10000 (max. Wert) zu den gleichen Gewichtsergebnissen geführt, sondern die Kostenwerte sind diejenigen, die die resultierenden Gewichte beeinflussen, da diese umgekehrt proportional zueinander sind.

Hinweis: Der Varianzwert hat keine Auswirkungen auf die Gewichtsergebnisse. In diesem Fall hätte eine Mindestvarianz von 134 oder eine Varianz von 10000 (max. Wert) zu den gleichen Gewichtsergebnissen geführt, sondern die Kostenwerte sind diejenigen, die die resultierenden Gewichte beeinflussen, da diese umgekehrt proportional zueinander sind.

Gewichtungen verstehen

Es gibt zwei verschiedene Gewichtstypen in IOS-XR, **Gewicht** und **normalisiertes Gewicht**. Die Verwendung dieser Pakete basiert auf der Anzahl der Hash-Buckets, die auf einer bestimmten Plattform unterstützt werden. XRv9000 unterstützt 32 Hash-Buckets, ASR 9000 und CRS-X 64 Hash-Buckets. Das bedeutet, dass die Gewichtung bei der Programmierung der Gewichtungswerte die Hash-Bucket-Grenze der jeweiligen Plattform nicht überschreiten darf. Mithilfe des Befehls **show cef <prefix> detail location <location>** können wir beobachten, welche normalisierten Gewichte programmiert werden. Basierend auf den festgelegten Kostenwerten haben wir eine Lastverteilung von 18, 13, was bedeutet, dass 31 Hash-Buckets zugewiesen wurden (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
```

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)

Table with 4 columns: Hash, OK, Interface, Address. It lists 32 entries for hash buckets 0-31, each with 'Y' in the OK column and 'remote' in the Address column. The interface alternates between 12 and 13.

Wie wir feststellen können, stellt die Summe des normalisierten Gewichts die Anzahl der Hash-Buckets dar, die von der Plattform zugewiesen werden. In diesem Fall können wir 32 Hash-Buckets niemals überschreiten, wie es die Grenze dieser speziellen Plattform vorgibt. Das Gewicht des Primärpfads (pm1) ist immer auf 4294967295 festgelegt, das ist das Höchstgewicht (2^32) - 1.

Bestimmen der Gewichtungswerte

Gewicht

Wir können leicht die Gewichtungen mit der Formel **Gewicht = beste Kosten / schlimmste Kosten * 4294967295** berechnen. So werden beispielsweise die Gewichtungen für Pfad 1 und Pfad 2 unten berechnet:

Weight_path_1 = immer auf 4294967295 festgelegt

Weight_path_2 = $151/201 * 4294967295 = 3226567470$

[Spoiler](#)

Hinweis: Genauigkeitsverluste können bei der Berechnung der Werte auftreten, wie bei Gleitkomma-Berechnungen, und ganze Zahlen müssen in RIB und FIB installiert werden.

Hinweis: Genauigkeitsverluste können bei der Berechnung der Werte auftreten, wie bei Gleitkomma-Berechnungen, und ganze Zahlen müssen in RIB und FIB installiert werden.

Normalgewicht

Wie bereits erwähnt, können wir in der CEF-Tabelle keine Gewichtswerte einbauen, die die Anzahl der Hash-Buckets durch eine Plattform überschreiten. Daher müssen wir die Gewichte normalisieren, bevor wir sie in Hardware programmieren. Die Plattform berechnet das Normalgewicht nach der Formel **Normalized Weight = (Path weight/Total weight) * Maximum Bucket Size**. Basierend auf unserem Beispiel können wir Folgendes berechnen:

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

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

[Spoiler](#)

Hinweis: Wenn G.C.D gleich 1 ist, dann wird die oben genannte Methode verwendet, andernfalls, wenn G.C.D \neq 1, dann normalisiert Gewicht wird die Teilung der resultierenden G.C.D durch die Gewichtswerte.

Hinweis: Wenn G.C.D gleich 1 ist, dann wird die oben genannte Methode verwendet, andernfalls, wenn G.C.D \neq 1, dann normalisiert Gewicht wird die Teilung der resultierenden G.C.D durch die Gewichtswerte.

Bearbeiten der CEF-Gewicht-/Lastverhältnisse

In einigen Szenarien können wir festlegen, welcher spezifische Pfadmetrik-Wert konfiguriert werden muss, um eine resultierende Lastverteilung zu erreichen. Wir könnten die entsprechende Pfadmetrik ermitteln, indem wir die Kosten für die Links ändern und basierend auf, bis wir den erforderlichen Wert erreichen oder schätzen. Beachten Sie, dass nicht alle von uns benötigten Gewichte genau möglich sind, aber wir können die erforderliche Verteilung annähern.

Bevor Sie fortfahren, berücksichtigen Sie folgende Einschränkungen:

- a) Nicht alle Lastverteilungen sind genau möglich, aber wir können eine Annäherung vornehmen.
- b) Überschreiten Sie niemals die Grenzwerte für Hasheisen. - Das bedeutet, dass die Summe aller Pfadgewichte die Hash-Buckets nicht überschreiten darf, wenn dies geschieht, dann muss das Gewicht normalisiert werden. Das bedeutet, dass beim Addieren aller Gewichte die Hash-Eimer-Grenze nicht überschritten wird.

c) ASR 9000 und CRS-X haben einen Grenzwert von 64 Hash-Buckets, XRv9000 einen Grenzwert von 32 Hash-Buckets.

d) Bei Verwendung von vor 6.4.1 ist die Gewichtsverteilung unterschiedlich, und der Pfad mit dem geringsten Gewicht wird immer auf ein Gewicht von 1 gesetzt, während andere Pfade sind Vielfache dieses Pfades, was bedeutet, dass er größer als 1 sein kann.

Beispiel 1: Gewicht/Lastverhältnis von 26/5

Nach der gleichen Topologie wie zuvor möchten wir eine 26/5-Gewichtsverteilung zwischen den beiden Links vornehmen.

i) Anfänglich werden die Kosten auf allen Pfaden ($100 + 100 + 1$) = 201 gleich gesetzt.

ii) Wenn wir die UCMP-Varianz auf den maximalen Wert festlegen, berücksichtigen wir alle Next-Hops.

iii) Wenn Sie die RIB überprüfen, wird der Standardstatus angezeigt, in dem XR1 ECMP ausführt.

```
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
```

Für dieses Beispiel verwenden wir einen Fall, in dem Sie die folgenden Gewichte verwenden möchten:

W1 = 26 (wichtigste Kosten)

W2 = 5 (zweitbeste Kosten)

Wir müssen einen Beinweg nehmen, für diesen Pfad, sollten die Kosten bereits bekannt sein, in diesem Fall Referenzpfad ist der Pfad über Gi0/0/0/0.12. Der Beinpfad wird mit Kosten von Ende bis Ende vorberechnet, die Pfadmetrik und das für diesen Pfad erforderliche Gewicht sind:

i) $X1+Y1+D1 = 100 + 100 + 1 = 201$. (Beachten Sie die Variablen, die an die einzelnen Links in der Topologie angefügt sind.)

ii) Gewicht 1 = 26

iii) Gewicht 2 = 5

iv) $pm1 = 201$ (Primärer Beinpfad); Gewicht = 26

v) $pm2 =$ noch unbekannt (sekundärer Pfad); Gewicht = 5

Die Gewichtungen berechnen.

Pfadmetrik $pm2$: $pm2 = (26/5) * 201 = 1045$

Ermittlung der Kosten für Verbindung X2 auf XR1.

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

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

Konfigurieren der OSPF-Kosten für eine X2-Verbindung

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

Bei der Überprüfung der Gewichtsverteilung und Lastverteilung wird deutlich, dass die erforderlichen Gewichtungen in CEF entsprechend den Berechnungen zugewiesen wurden.

```
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
```


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