

# Catalyst 6500、4500、および 3750 シリーズ スイッチでの EtherChannel ロードバランシング

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[Cisco Catalyst 6500 シリーズ スイッチ](#)

[『Catalyst 4500 Series Switches](#)

[Catalyst 3750 シリーズ スイッチ](#)

[潜在的な問題](#)

## 概要

このドキュメントでは、Cisco Catalyst 6500、4500、3750 シリーズ スイッチで特定のトラフィック フローによって使用される EtherChannel メンバー リンクを特定する方法について説明します。

## 前提条件

### 要件

EtherChannels に関する基本的な知識があることが推奨されます。

### 使用するコンポーネント

このドキュメントの情報は、次のプラットフォームに基いています。Cisco Catalyst 6500、4500、2960、3750、3750G、3750X、および 3560 シリーズ スイッチ。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

## 背景説明

EtherChannel ロード バランシングは、トラフィック タイプ用に設定されたハッシュ方式 (ロード バランシング アルゴリズム) に基づいて、スイッチが 0 ~ 7 からのハッシュ結果を割り当てることにより機能します。このハッシュ結果は通常、結果バンドル ハッシュ (RBH) と呼ばれます。

フローの数は、設定されているロード バランシング アルゴリズムによって異なることに注意してください。次に例を示します。

```
Source 192.168.1.1 (mac a.a.a ) sending a tcp stream to 172.16.1.1 ( mac b.b.b )  
with a source tcp port of 50 and destination port 2000
```

```
Source 192.168.1.1 (mac a.a.a ) sending a tcp stream to 209.165.201.1 ( mac c.c.c )  
with a source tcp port of 60 and destination 2000.
```

```
If configured load balancing algorithm is SRC_MAC  
Then no of flows = 1
```

```
If configured load balancing algorithm is DST_MAC  
Then no of flows = 2
```

```
If configured load balancing algorithm is DST_PORT  
Then no of flows= 1
```

## Cisco Catalyst 6500 シリーズ スイッチ

- 稼働しているロード バランシング アルゴリズムを確認します。
- スイッチ プロセッサ ( SP ) で、**show etherchannel load-balance** と入力します。

```
6500#remote login sw  
Trying Switch ...  
Entering CONSOLE for Switch  
Type "^C^C^C" to end this session
```

```
6500-sp#show etherchannel load-balance  
EtherChannel Load-Balancing Configuration:  
    src-dst-ip  
    mpls label-ip
```

- 目的のフロー間のパケットで選択されている RBH 値を見つけます。

```
6500-sp#test etherchannel load-balance interface port-channel
```

この例では、192.168.1.1 ~ 172.16.1.1の間のフローで、問題のポートチャンネルはport-channel 1です。ステップ1の出力に基づいてコマンドの属性を選択します。設定されたロードバランシングアルゴリズムがsrc\_ipのsrc-ipを指定します。src-dst ip が、172.16.1.1。

```
6500-sp#test etherchannel load-balance int port-channel 1 ip 192.168.1.1 172.16.1.1
```

```
Computed RBH: 0x5  
Would select Gi3/2 of Po1
```

- RBH 値にマッピングされている物理ポートを見つけます。( オプション )

一部のバージョンの Cisco IOS<sup>(R)</sup> では、コマンドの出力に、選択された物理インターフェイスが表示されません。このステップは、出カインターフェイス情報がステップ 2 で生成されない場合にのみ実行します。

```
6500-sp#test etherchannel load-balance int port-channel 1 ip 192.168.1.1 172.16.1.1
Computed RBH: 0x5
```

ルート プロセッサのコンソールに切り替え、**show interface port-channel <num> etherchannel** コマンドを入力します。出力で、物理インターフェイスに対応する Load カラムを調べます。Load の値をバイナリに変換します ( 次の例を参照 )。

```
6500-sp#exit
```

```
[Connection to Switch closed by foreign host]
```

```
6500#show interface port-channel 1 etherchannel
```

```
Port-channell1 (Primary aggregator)
Age of the Port-channel = 0d:01h:05m:54s
Logical slot/port = 14/1          Number of ports = 2
HotStandBy port = null
Port state = Port-channel Ag-Inuse
Protocol = LACP
Fast-switchover = disabled
```

```
Ports in the Port-channel:
```

```
Index Load Port EC state No of bits
```

```
-----+-----+-----+-----+-----
 0    55   Gi3/1   Active    4
 1    AA   Gi3/2   Active    4
```

ここで、gi3/2 の Load 値は AA、gi3/1 の Load 値は 55 です。

```
          7654 3210
gig3/2 - AA - 1010 1010
          ---- ----
          |      |
          A      A
```

```
gig3/1 - 55 - 0101 0101
          ---- ----
          |      |
          5      5
```

For gi3/2 bits 1,3,5 and 7 are set. So RBH value of 1,3,5,and 7 chooses gi3/2.

For gi3/1 bits 0,2,4 and 6 are set. So RBH value of 0,2,4,and 6 chooses gi3/1.

出力は、2つのインターフェイスのそれぞれに4ビットが設定されていることを示しています。したがって、EtherChannelに2つのリンクがある場合、各リンクが使用される確率は同じです。

ただし、EtherChannelに3つのリンクがある場合は、test etherchannelの出力が次のようになります。

```
6500#show interface port-channel 1 etherchannel
Port-channell1 (Primary aggregator)
Age of the Port-channel = 0d:01h:05m:54s
Logical slot/port = 14/1          Number of ports = 2
HotStandBy port = null
Port state = Port-channel Ag-Inuse
```

```
Protocol          = LACP
Fast-switchover   = disabled
Ports in the Port-channel:
```

Index	Load	Port	EC state	No of bits
0	49	Gi3/1	Active	3
1	92	Gi3/2	Active	3
2	24	Gi3/3	Active	2

ここでは、ビット共有比率は3:3:2です。したがって、2つのリンクは、3番目のリンク (最後の追加セクションの方が多く) と比較して、使用される確率が高くなります。

## 『Catalyst 4500 Series Switches

1. `show etherchannel load-balance` コマンドで、設定されているロード バランシング アルゴリズムを確認します。
2. `show platform software etherchannel port-channel 1 map` コマンドを使用して、出カインターフェイスを見つけます。

```
4500#show platform software etherchannel port-channel 1 map ip 192.168.1.1
172.16.1.1
```

```
Map port for IP 192.168.1.1, 172.16.1.1 is Gi3/1(Po1)
NOTE: Software forwarded traffic uses Gi3/1(Po1)
```

## Catalyst 3750 シリーズ スイッチ

Catalyst 3750シリーズスイッチでは、同様の8ビットハッシュアルゴリズムが使用されます。ここでは、EtherChannel内のリンク数が2、4、または8の場合でも、トラフィックの分散が行われます。ポートチャネル内のインターフェイスを確認するコマンド:

```
test etherchannel load-balance interface port-channel
```

3750 では、ポートチャネルのロード バランシング アルゴリズムが `src-dst ip` として設定されている (デフォルトではない) と想定します。次に、192.168.1.1 からの 172.16.1.1 へのトラフィックが使用するリンクを特定する方法の例を示します。

```
3750(config)#port-channel load-balance src-dst-ip
```

```
3750#show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
    src-dst-ip
```

```
3750#test etherchannel load-balance interface port-channel 1 ip 192.168.1.1 172.16.1.1
Would select Gi3/1 of Po1
```

ロード バランシング アルゴリズムが MAC アドレス ベースの場合は、パケットの送信元および

宛先 MAC アドレスを指定するために、前のコマンドを使用できます。

## 潜在的な問題

ここでは、EtherChannel メンバ インターフェイスのトラフィック分配の偏りを引き起こす可能性のあるシナリオと、ロード バランシングを最適化するために実行する必要がある手順を示します。

- **シナリオ** : EtherChannel に 2 つのフローと 2 つの物理インターフェイスがある場合、1 つのフローの方が負荷が高いことがあります。5 つのフローがあり、その 1 つが最も負荷が高い場合は、このフローによって他のフローが影響されることがあります。このフローが選択した物理インターフェイスは、他の物理インターフェイスに比べて使用率が高くなります。
- **解決策** : 負荷の高すぎるフローを制御します。これは、ホスト側から確認する必要があります。
- 
- **シナリオ** : 一般的な問題は、フローが不足しており、少数のフローのほとんどが、同じ物理インターフェイスにハッシュされることです。
- **解決策** : フローの数を増やします。ハッシュ アルゴリズムを、トラフィックに最適なアルゴリズムに変更してください。
- 
- **シナリオ** : EtherChannel の物理リンク数が 3、5、6、または 7 の場合、そのいくつかで、他のリンクよりトラフィックを分配される確立が高くなり ( 各物理インターフェイスに割り当てられているハッシュ ビット数に基づいて )、これにより、トラフィックの分配が偏る可能性があります。
- **解決策** : EtherChannel で使用する物理リンクの数を 2、4、または 8 にします。