

排除埠通道負載均衡中的極化問題

目錄

[簡介](#)

[背景](#)

[必要條件](#)

[拓撲](#)

[組態](#)

[流量](#)

[疑難排解](#)

[因應措施](#)

簡介

本文說明在埠通道負載均衡中可能發生極化的情況，並提供如何防止極化的建議。

背景

極化 是一個雜湊演算法選擇網路中特定路徑而未使用冗餘路徑的問題

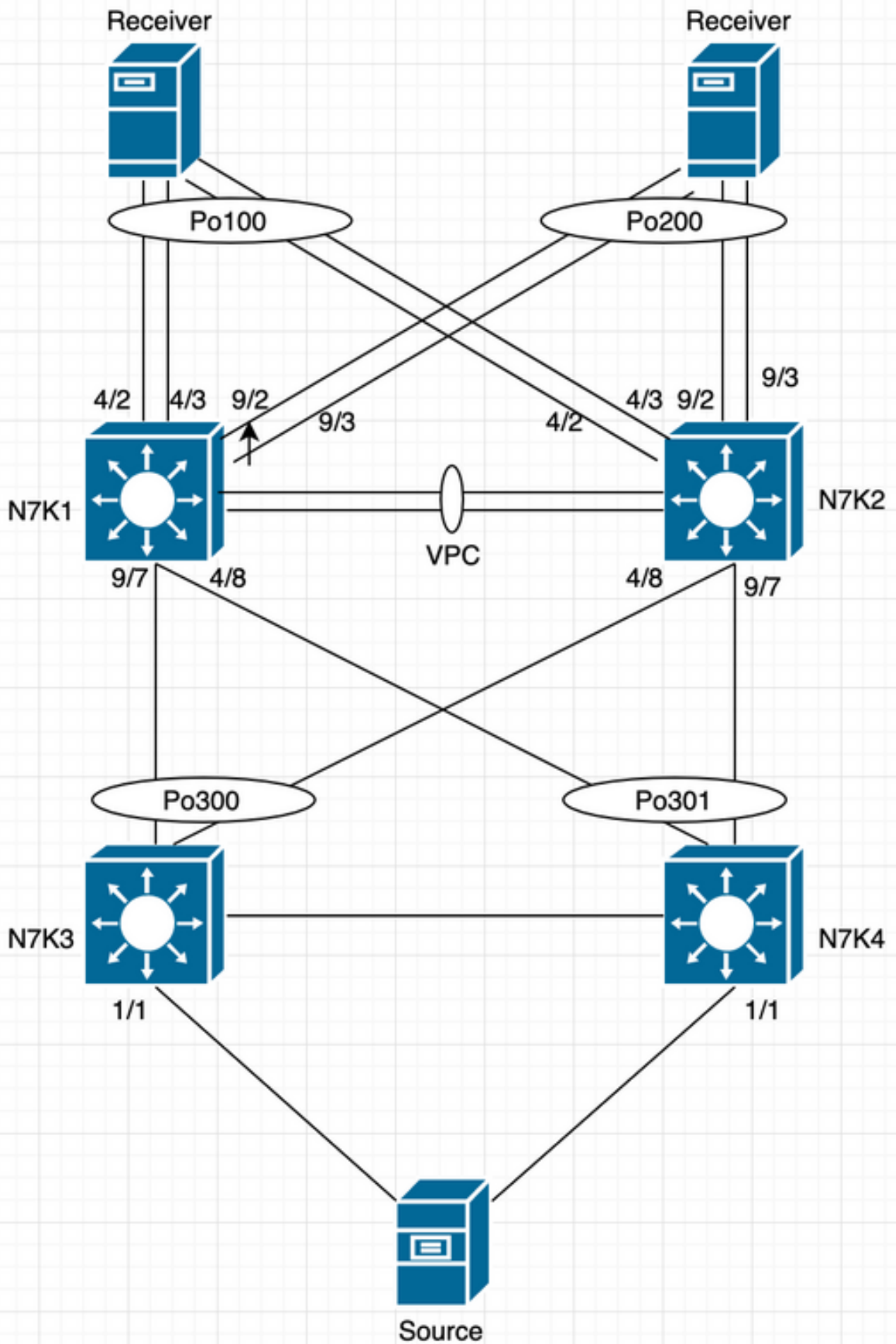
必要條件

建議瞭解下列主題。

[鏈路聚合控制協定](#)

Cisco Nexus平台

拓撲



組態

在VPC中連線的N7K1和N7K2以及Po100、Po200、Po300和Po301在VPC埠通道中。

N7K1和N7K2充當純L2交換機，這些交換機上不會發生路由。

所有交換機都運行相同的埠通道負載平衡演算法

從N7K1和N7K2傳出的流量出現極化問題，無論從源到目的地的流量是否在同一個vlan中（無路由），或者它們是否處於不同的vlan中，路由發生在N7K3或N7k4上。

流量

來源將多個串流傳送到目的地（具有多個來源和目的地IP位址，且第4層連線埠資訊也會因封包的不同而有所不同）。良好的流量組合可確保流量在理想的情況下均勻分佈在埠通道成員介面之間。

從N7k3/N7k4上的源地址發往目的地的流量，然後通過N7K1/N7K2發往目的地。

N7K1和N7K2上Po100和Po200的成員鏈路中的一個鏈路傳送了近99%的流量，另一個鏈路保持空閒。（例如，在每台交換機N7K1和N7K2上，4/2和4/3之間的一個鏈路傳輸99%的單播流量，而另一個鏈路傳輸不到1%，同樣，9/2和9/3之間的一個鏈路傳輸99%的流量，而另一個鏈路傳輸不到1%。故障排除部分中的輸出顯示N7K1上po100和po200成員介面上的流量，N7K2上可見類似的輸出）。

無論使用的埠通道負載均衡演算法的型別如何，只要在N7K1/N7K2對和N7K3/N7K4對上使用相同的埠通道負載均衡演算法，問題就會出現。下面給出了檢查埠通道負載平衡演算法的命令。

```
N7K1# show port-channel load-balance
Warning: Per Packet Load balance configuration has higher precedence
System config:
  Non-IP: src-dst mac
  IP: src-dst ip-l4port-vlan rotate 0
Port Channel Load-Balancing Configuration for all modules:
Module 1:
  Non-IP: src-dst mac
  IP: src-dst ip rotate 0
Module 2:
  Non-IP: src-dst mac
  IP: src-dst ip rotate 0
Module 3:
  Non-IP: src-dst mac
  IP: src-dst ip rotate 0
Module 4:
  Non-IP: src-dst mac
  IP: src-dst ip-l4port-vlan rotate 0
Module 7:
  Non-IP: src-dst mac
  IP: src-dst ip-l4port-vlan rotate 0
Module 8:
  Non-IP: src-dst mac
  IP: src-dst ip-l4port-vlan rotate 0
Module 9:
  Non-IP: src-dst mac
  IP: src-dst ip-l4port-vlan rotate 0
```

疑難排解

如果埠通道出現負載均衡不均衡，則可能是由於極化。

當流量到達N7K3和N7K4交換機時，會通過N7K4的Po301和N7K3的Po300將它們轉發到N7K1/N7K2交換機。此時將啟動負載平衡演算法，並將部分流轉發到N7K1，將其他流轉發到N7K2。

最初，所有流量進入交換機N7K3/N7K4（位於eth1/1上），根據src-dest ip和I4埠資訊，某些流在指向N7K1的鏈路上雜湊，其他流在指向N7K2的鏈路上雜湊。雜湊操作基於交換機計算的rbh值完成。為簡單起見，我們假設交換機根據使用的負載平衡演算法將傳入流量分為兩個流（流X和流Y）。流X從一個埠通道成員鏈路發出，流Y從另一個埠通道成員鏈路發出。

現在，當流量到達N7K1/N7K2對時，可能發生兩種情況。（認為X和Y可以互換）

案例1:

N7K3將流X傳送到N7K1，將流Y傳送到N7K2

和

N7K4將流Y傳送到N7K1並將流X傳送到N7K2

案例2:

N7K3將流X傳送到N7K1，將流Y傳送到N7K2

和

N7K4將流X傳送到N7K1，將流Y傳送到N7K2

在案例1中，N7K1和N7K2接收兩種型別的流（流X和流Y），即使使用與N7K3/N7K4相同的埠通道負載均衡演算法，也不會因為流從不同鏈路上的Po100和Po200輸出而出現極化，因此，我們可以看到埠通道成員介面之間更好的流量分配。

在案例2中，N7K1僅接收流X，N7K2僅接收流Y，如果這些交換機上使用的埠通道負載平衡演算法與N7K3/N7K4配對中使用的演算法相同，則這可能會產生極化。由於N7K1和N7K2使用相同的埠通道負載均衡演算法，因此N7K1僅在Po100/Po200的一個成員鏈路上傳送流X，而另一個成員鏈路將不會轉發任何流量。同樣，N7K2隻在Po100/Po200的一個成員鏈路上傳送流Y，而另一個成員鏈路不會轉發任何流量。

由於交換機N7K1和N7K2接收的流量已經分類為「開始」，因此僅使用一個埠通道成員鏈路將所有傳入流量從交換機N7K1/N7K2傳送出去，而從另一個成員鏈路傳送的任何流量都不傳送。如果傳入流量速率超過單埠通道鏈路的頻寬，則其他流量可能會被丟棄，因為其他埠通道成員鏈路不會轉發此流量。

在port-channel中使用了超過兩個鏈路時，也會出現類似的問題。例如，如果埠通道中使用了四個鏈路，則根據雜湊的情況，要麼不會發生極化，要麼會看到部分極化，其中四個埠通道成員鏈路中只有兩個鏈路用於轉發所有傳入流量，另外兩個鏈路不轉發任何流量

由於設計導致極化，因此對設計進行分析以確保不存在極化的現象非常重要。下面給出了在N7k1上發生Po100和Po200的極化輸出（在N7K2上也可以看到類似的輸出）。

```

N7K1# show port-channel summary | i 200
200 Po200(SU) Eth LACP Eth9/2(P) Eth9/3(P)

N7K1# show port-channel traffic interface port-channel 200
NOTE: Clear the port-channel member counters to get accurate statistics
ChanId Port Rx-Ucst Tx-Ucst Rx-Mcst Tx-Mcst Rx-Bcst Tx-Bcst
-----
200 Eth9/2 0.0% 99.99% 44.44% 4.00% 0.0% 100.00%
200 Eth9/3 0.0% 0.00% 55.55% 96.00% 0.0% 0.0%

N7K1# show port-channel summary | i 100
100 Po100(SU) Eth LACP Eth4/2(P) Eth4/3(P)

N7K1# show port-channel traffic interface port-channel 100
NOTE: Clear the port-channel member counters to get accurate statistics
ChanId Port Rx-Ucst Tx-Ucst Rx-Mcst Tx-Mcst Rx-Bcst Tx-Bcst
-----
100 Eth4/2 0.0% 99.99% 40.55% 7.00% 0.0% 100.00%
100 Eth4/3 0.0% 0.00% 54.44% 93.00% 0.0% 0.0%

```

[CSCvq26885](#) 已歸檔以獲取外部文檔。

因應措施

遵循一些用來確保極化不會發生的變通辦法。

1. 正確設計：由於極化的主要原因是設計不當，因此最好改變網路設計，確保拓撲中沒有極化空間

如果無法更改設計，我們可以執行以下操作。

2. 在每個交換機級別使用不同的埠通道負載平衡演算法（N7K1/N7k2對使用一個演算法，N7K3//N7k4對使用不同的演算法）。當負載均衡演算法發生更改時，N7k1/N7k2交換機現在會根據除N7k3/N7k4交換機所使用的資訊以外的某些其他資訊對傳入流量進行雜湊，因此傳出流量使用所有埠通道成員鏈路。（決定選擇哪種演算法取決於交換器接收的流量型別）
3. 如果客戶希望使用相同的負載均衡演算法，請在交換機的每一層使用不同的輪換值。Rotate命令通過抵消使用者配置的位元組的雜湊輸入，在雜湊演算法中引入了隨機性，並有助於避免極化。（對N7k1/N7k2對使用一個旋轉值，對N7k3/N7k4對使用不同的旋轉值）