

排除Catalyst交換機上的STP故障

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[慣例](#)

[背景資訊](#)

[STP故障的原因](#)

[排除轉發環路故障](#)

- [1. 辨識環路](#)
- [2. 發現環路的拓撲 \(範圍 \)](#)
- [3. 中斷回圈](#)
- [4. 尋找並修正回圈的原因](#)
- [5. 恢復冗餘](#)

[調查拓撲更改](#)

[找出泛洪的原因](#)

[尋找TC的來源](#)

[採取措施防止過度的TC](#)

[解決收斂時間相關問題](#)

[使用STP Debug命令](#)

[保護網路免受轉發環路的影響](#)

- [1. 在所有交換器到交換器連結上啟用單向連結偵測\(UDLD\)](#)
- [2. 在所有交換機上啟用環路防護](#)
- [3. 在所有終端站連線埠上啟用Portfast](#)
- [4. 在兩側 \(如果支援 \) 和Non-SilentOption上將EtherChannel設定為DesirableMode](#)
- [5. 不要停用交換器對交換器連結上的自動交涉 \(如果支援 \)](#)
- [6. 調整STP計時器時要格外小心](#)
- [7. 如果可能發生拒絕服務攻擊，請透過根防護保護網路STP周邊](#)
- [8. 在啟用了Portfast的埠上啟用BPDU防護，以防止STP受到連線到埠的未授權網路裝置 \(如集線器、交換機和橋接路由器 \) 的影響](#)
- [9. 避免管理VLAN上的使用者流量](#)
- [10. 可預測 \(硬編碼 \) 的STP根和備份STP根放置](#)

[相關資訊](#)

簡介

本檔案介紹如何使用Cisco IOS®軟體對跨距樹狀目錄通訊協定(STP)問題進行疑難排解。

必要條件

需求

思科建議您瞭解以下主題：

- 各種生成樹型別及其配置方法。有關詳細資訊，請參閱[配置STP和IEEE 802.1s MST](#)。
- 各種生成樹功能及其配置方法。有關詳細資訊，請參閱[配置STP功能](#)。

採用元件

本文中的資訊係根據以下軟體和硬體版本：

- 含Supervisor 2引擎的Catalyst 6500
- Cisco IOS 軟體版本 12.1(13)E

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

慣例

如需更多文件慣例的相關資訊，請參閱[思科技術提示慣例](#)。

背景資訊

有特定命令僅適用於Catalyst 6500/6000；但是，您可以將大多數原則應用於運行Cisco IOS軟體的任何思科Catalyst交換機。

大多數STP都存在以下三個問題：

- 轉發環路。
- 由於STP拓撲更改(TC)率高而導致過度泛洪。
- 與收斂時間相關的問題。

因為網橋沒有機制跟蹤某個資料包是轉發多次（例如，IP生存時間[TTL]），還是用於丟棄網路中循環時間過長的流量。同一第2層(L2)域中的兩台裝置之間只能存在一條路徑。

STP的目的是根據STP演算法阻塞冗餘埠，並將冗餘物理拓撲解析為樹狀拓撲。轉發環路（例如STP環路）發生在冗餘拓撲中沒有埠被阻塞，並且流量無限循環轉發。

一旦轉發環路開始，它將導致沿其路徑的最低頻寬鏈路擁塞。如果所有鏈路的頻寬相同，則所有鏈路都會擁塞。這種擁塞會導致資料包丟失，並導致受影響的L2域中的網路發生故障。

過度泛濫後，症狀不再那麼明顯。慢速鏈路可能會因泛洪流量而擁塞，這些擁塞鏈路後面的裝置或使用者可能會遇到速度緩慢或完全失去連線的情況。

STP故障的原因

STP對其操作環境做出某些假設。以下為與本檔案最相關的假設：

- 兩個網橋之間的每個鏈路都是雙向的。這意味著，如果A直接連線到B，則A接收B傳送的消息，B接收A傳送的消息，只要它們之間的鏈路處於接通狀態。
- 每個運行STP的網橋都可以定期接收、處理和傳輸STP網橋協定資料單元(BPDU)，也稱為STP資料包。

儘管這些假設似乎是合乎邏輯和顯而易見的，但有些情況下它們並未得到滿足。其中大多數情況都涉及某種硬體問題；但是，軟體缺陷也會導致STP故障。各種硬體故障、配置錯誤、連線問題導致大多數STP故障，而軟體故障則佔少數。由於交換機之間存在不必要的其他連線，也可能出現STP故障。由於這些額外連線，VLAN進入關閉狀態。要解決此問題，請刪除交換機之間所有不需要的連線。

當其中一個假設不滿足時，一個或多個網橋將無法接收或處理BPDU。這表示網橋（或多個網橋）未發現網路拓撲。如果不知道正確的拓撲，交換機將無法阻塞環路。因此，泛洪流量會透過環路拓撲循環，消耗所有頻寬，並中斷網路。

交換機無法接收BPDU的原因示例包括收發器故障或千兆介面轉換器(GBIC)、電纜問題或埠、板卡或Supervisor引擎上的硬體故障。STP故障的一個常見原因是網橋之間的單向鏈路。在這種情況下，一個網橋會傳送BPDU，但下游網橋永遠不會收到這些BPDU。由於交換機無法處理收到的BPDU，STP處理也可能被超載CPU（99%或以上）中斷。BPDU可能會在從網橋到另一個網橋的路徑中損壞，這也導致無法正常執行STP行為。

除了轉發環路以外，當沒有埠被阻塞時，也存在只有某些資料包透過阻塞流量的埠被錯誤轉發的情況。在大多數情況下，這是由軟體問題引起的。此類行為可能導致慢環路。這表示某些封包已循環，但大部分流量仍流經網路，因為連結沒有擁塞。

排除轉發環路故障

轉發環路在起源（原因）和效果方面差異很大。由於存在多種可能影響STP的問題，本文檔僅提供有關如何排除轉發環路故障的一般指南。

在開始故障排除之前，您需要以下資訊：

- 詳細顯示所有交換機和網橋的實際拓撲圖。
- 它們相應的埠號（互連）。
- STP配置詳細資訊，例如哪台交換機是根和備用根，哪些鏈路具有非預設的成本或優先順序，以及阻塞流量的埠的位置。

1. 辨識環路

當網路中形成轉發環路時，通常會出現以下症狀：

- 與受影響網路區域的連線、來自和經過這些區域的連線丟失。
- 連線到受影響網段或VLAN的路由器上的CPU使用率較高，可能導致各種症狀，例如路由協定鄰居抖動或熱備用路由器協定(HSRP)活動路由器抖動。

- 鏈路利用率高 (通常為100%)。
- 高交換機背板利用率 (與基線利用率相比)。
- 系統日誌消息，指示網路中資料包循環 (例如HSRP重複IP地址消息)。
- Syslog消息指示持續地址重新學習或MAC地址擺動消息。
- 許多介面上的輸出丟棄數量增加。

其中任何原因都可單獨指出不同的問題 (或根本不指出問題)。但是，當同時觀察到其中許多情況時，網路中極有可能已形成轉發環路。驗證這一點的最快方法是檢查交換機背板流量的利用率：

```
<#root>
```

```
cat#
```

```
show catalyst6000 traffic-meter
```

```
traffic meter = 13%
```

```
Never cleared
```

```
peak = 14%
```

```
reached at 12:08:57 CET Fri Oct 4 2002
```



注意：帶有Cisco IOS軟體的Catalyst 4000當前不支援此命令。

如果當前流量級別過高或基線級別未知，請檢查最近是否達到了峰值級別，以及它是否接近當前流量級別。例如，如果峰值流量級別為15%，而它是在兩分鐘前到達的，並且當前流量級別為14%，則意味著交換機具有異常高的負載。如果流量負載處於正常水準，則這可能意味著沒有環路或此裝置未參與環路。但是，它仍然可能涉及慢循環。

2. 發現環路的拓撲 (範圍)

一旦確定網路中斷的原因是轉發環路，最高優先順序就是停止環路並恢復網路運行。

要停止環路，您必須知道哪些埠參與了環路：檢視鏈路使用率最高的埠 (每秒資料包數)。show interface Cisco IOS軟體命令可以顯示每個介面的利用率。

要僅顯示利用率資訊和介面名稱 (以便快速分析)，請使用Cisco IOS軟體過濾正規表示式輸出。發出show interface | include line|\vseccommand僅顯示每秒資料包統計資訊和介面名稱：

```
<#root>
```

```
cat#
```

```
show interface | include line|\sec
```

```
GigabitEthernet2/1 is up, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/2 is up, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec

GigabitEthernet2/3 is up, line protocol is up
  5 minute input rate 99765230 bits/sec, 24912 packets/sec

  5 minute output rate 0 bits/sec, 0 packets/sec

GigabitEthernet2/4 is up, line protocol is up

  5 minute input rate 1000 bits/sec, 27 packets/sec

  5 minute output rate 101002134 bits/sec, 25043 packets/sec

GigabitEthernet2/5 is administratively down, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/6 is administratively down, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/7 is up, line protocol is down
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec

GigabitEthernet2/8 is up, line protocol is up


  5 minute input rate 2000 bits/sec, 41 packets/sec


  5 minute output rate 99552940 bits/sec, 24892 packets/sec
```


注意鏈路利用率最高的介面。在本例中，這些埠是介面g2/3、g2/4和g2/8；它們是參與環路的埠。


3. 中斷回圈

若要中斷回圈，您必須關閉或中斷相關連線埠。不僅要停止環路，還要找出並修復環路的根本原因，這一點尤為重要。相對而言，打破循環比較容易

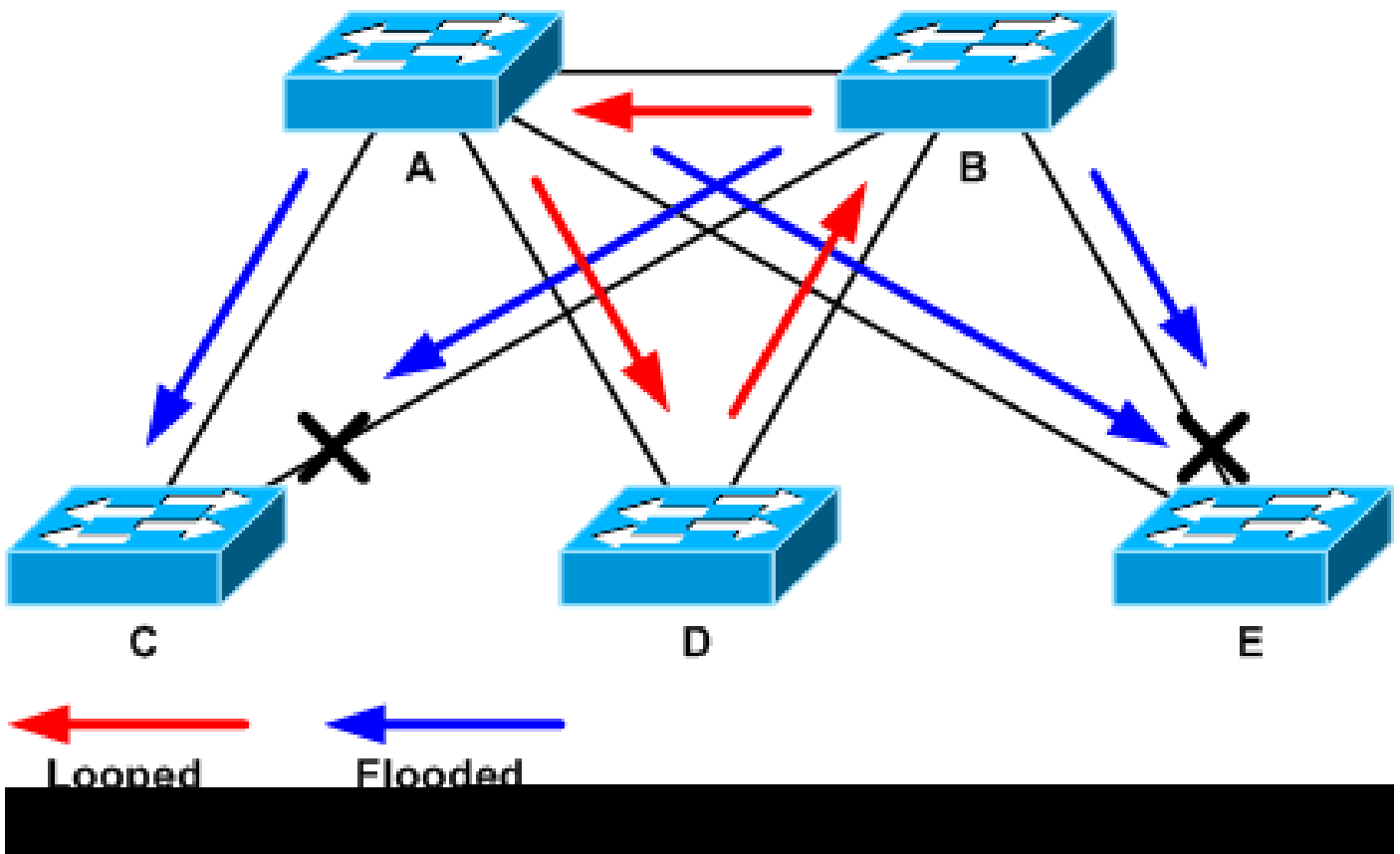
 注意：不必同時關閉或斷開所有埠。你可以一次關掉一個。在受環路影響的匯聚點（例如分佈或核心交換機）關閉埠會更好。如果一次關閉所有埠，並逐一啟用或重新連線這些埠，則不會起作用；環路會停止，並且在故障埠重新連線後無法立即啟動。因此，很難將故障與任何特定埠相關聯。

 注意：若要中斷回圈，建議您在重新啟動交換器之前收集資訊。否則，後續根本原因分析將非常困難。停用或斷開每個埠後，必須檢查交換機背板利用率是否恢復到正常水準。

 注意：請記住，埠不維持環路，而是將隨環路到達的流量泛洪。關閉此類泛洪埠時，僅將背板

 利用率降低一小部分，但不會停止環路。

在下一個示例拓撲中，環路位於交換機A、B和D之間。因此，鏈路AB、AD和BD是持續的。如果關閉其中任何鏈路，則會停止環路。鏈路AC、AE、BC和BE只是泛洪隨環路到達的流量。



循環和泛洪流量

在支援連線埠關閉後，背板使用率會降至正常值。您需要知道哪個埠的關閉使背板利用率（以及其他埠的利用率）達到正常水準。此時，環路停止，網路操作改善；但是，由於環路的原始原因未修復，仍然存在其他問題。

4. 尋找並修正回圈的原因

循環停止後，您需要確定循環開始的原因。這是整個過程的難點，因為原因可能不同。要正式確定在每種情況下都有效的確切程式也很困難。

準則：

- 檢查拓撲圖以查詢冗餘路徑。其中包括上一步中找到的返回同一交換機的支援埠（在環路期間交換的路徑資料包）。在前面的示例拓撲中，此路徑為AD-DB-BA。
- 對於冗餘路徑上的每台交換機，檢查該交換機是否知道正確的STP根。

L2網路中的所有交換機必須商定一個通用的STP根。當網橋一致地為特定VLAN或STP例項中的STP根顯示不同的ID時，這就是問題的明顯症狀。發出show spanning-tree vlan vlan-id 命令，以顯示給定VLAN的根網橋ID：

```
<#root>
```

```
cat#
```

```
show spanning-tree vlan 333
```

```
MST03
```

```
Spanning tree enabled protocol mstp
```

```
Root ID      Priority      32771
            Address      0050.14bb.6000
            Cost        20000
            Port        136 (GigabitEthernet3/8)
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID    Priority      32771 (priority 32768 sys-id-ext 3)
            Address      00d0.003f.8800
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
```

Interface	Role	Sts	Cost	Prio.Nbr	Status
Gi3/8	Root	FWD	20000	128.136	P2p
Po1	Desg	FWD	20000	128.833	P2p

您可以從連線埠找到VLAN編號，因為回圈中涉及的連線埠是在先前步驟中建立的。如果所討論的埠是TRUNK，則通常涉及該TRUNK上的所有VLAN。如果實際情況不是這樣（例如，環路似乎發生在一個VLAN上），則可以嘗試發出show interfaces | include L2|line|broadcastcommand（僅適用於Catalyst 6500/6000系列交換機上的Supervisor 2和更高版本的引擎，因為Supervisor 1不提供每VLAN交換統計資料）。請僅檢視VLAN介面。交換封包數量最多的VLAN通常是發生回圈的VLAN：

```
<#root>
```

```
cat#
```

```
show interface | include L2|line|broadcast
```

```
Vlan1 is up, line protocol is up
  L2 Switched: ucast: 653704527 pkt, 124614363025 bytes - mcast:
    23036247 pkt, 1748707536 bytes
    Received 23201637 broadcasts, 0 runts, 0 giants, 0 throttles

Vlan10 is up, line protocol is up
  L2 Switched: ucast: 2510912 pkt, 137067402 bytes - mcast:
    41608705 pkt, 1931758378 bytes
    Received 1321246 broadcasts, 0 runts, 0 giants, 0 throttles

Vlan11 is up, line protocol is up
  L2 Switched: ucast: 73125 pkt, 2242976 bytes - mcast:
    3191097 pkt, 173652249 bytes
    Received 1440503 broadcasts, 0 runts, 0 giants, 0 throttles

Vlan100 is up, line protocol is up
  L2 Switched: ucast: 458110 pkt, 21858256 bytes - mcast:
    64534391 pkt, 2977052824 bytes
    Received 1176671 broadcasts, 0 runts, 0 giants, 0 throttles

Vlan101 is up, line protocol is up
  L2 Switched: ucast: 70649 pkt, 2124024 bytes - mcast:
```

```
2175964 pkt, 108413700 bytes
Received 1104890 broadcasts, 0 runts, 0 giants, 0 throttles
```

在本例中，VLAN 1佔用的廣播和L2交換流量最多。確保根埠標識正確。

根埠到根網橋的成本最低（有時，一條路徑的跳數較短，但成本較長，因為低速埠成本較高）。要確定哪個埠被視為給定VLAN的根埠，請發出show spanning-tree vlan 命令：

```
<#root>
```

```
cat#
```

```
show spanning-tree vlan 333
```

```
MST03
```

```
Spanning tree enabled protocol mstp
Root ID    Priority    32771
           Address    0050.14bb.6000
           Cost      20000
```

```
Port      136 (GigabitEthernet3/8)
```

```
        Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
Bridge ID Priority    32771 (priority 32768 sys-id-ext 3)
Address    00d0.003f.8800
Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

Interface	Role	Sts	Cost	Prio.Nbr	Status
Gi3/8	Root	FWD	20000	128.136	P2p
Po1	Desg	FWD	20000	128.833	P2p

確保在根埠和應該阻塞的埠上定期接收BPDU。

BPDU由根網橋以每個hello間隔（預設情況下為兩秒）傳送。非根網橋接收、處理、修改和傳播從根接收的BPDU。請發出show spanning-tree interface interface detail 命令，以檢視是否收到了BPDU：

```
<#root>
```

```
cat#
```

```
show spanning-tree interface g3/2 detail
```

```
Port 130 (GigabitEthernet3/2) of MST00 is backup blocking
Port path cost 20000, Port priority 128, Port Identifier 128.130.
Designated root has priority 0, address 0007.4f1c.e847
Designated bridge has priority 32768, address 00d0.003f.8800
Designated port id is 128.129, designated path cost 2000019
```


Timers: message age 4, forward delay 0, hold 0

Number of transitions to forwarding state: 0

Link type is point-to-point by default, Internal
Loop guard is enabled by default on the port
BPDU: sent 3,


received 53

cat#

show spanning-tree interface g3/2 detail

Port 130 (GigabitEthernet3/2) of MST00 is backup blocking
Port path cost 20000, Port priority 128, Port Identifier 128.130.
Designated root has priority 0, address 0007.4f1c.e847
Designated bridge has priority 32768, address 00d0.003f.8800
Designated port id is 128.129, designated path cost 2000019
Timers: message age 5, forward delay 0, hold 0
Number of transitions to forwarding state: 0
Link type is point-to-point by default, Internal
Loop guard is enabled by default on the port
BPDU: sent 3,

received 54

 注意：在該命令的兩個輸出之間收到一個BPDU（計數器從53變為54）。

顯示的計數器實際上是STP進程本身維護的計數器。這意味著，如果接收計數器遞增，則不僅物理埠接收了BPDU，STP進程也接收了BPDU。如果received BPDU計數器在應該是根備用或備份埠的埠上沒有增加，則檢查該埠是否完全接收多播（BPDU作為多播傳送）。發出show interface interface counters command：

<#root>

cat#

show interface g3/2 counters

Port	InOctets	InUcastPkts	InMcastPkts	InBcastPkts
Gi3/2	14873036	2		

Port	OutOctets	OutUcastPkts	OutMcastPkts	OutBcastPkts
Gi3/2	114365997	83776	732086	19

cat#

```
show interface g3/2 counters
```

```
Port          InOctets   InUcastPkts
InMcastPkts
  InBcastPkts
Gi3/2         14873677   2
89391
              0

Port          OutOctets  OutUcastPkts  OutMcastPkts  OutBcastPkts
Gi3/2         114366106   83776         732087        19
```

STP埠角色的概要可在[使用環路防護和BPDU遲滯檢測功能的生成樹協定增強功能的使用環路防護和BPDU遲滯檢測增強STP](#) 部分中找到。如果未收到BPDU，請檢查連線埠是否計算錯誤數量。請發出show interface interface counters errorscommand：

```
<#root>
```

```
cat#
```

```
show interface g4/3 counters errors
```

```
Port   Align-Err  FCS-Err  Xmit-Err  Rcv-Err  UnderSize  OutDiscards
Gi4/3   0          0        0         0        0          0

Port   Single-Col  Multi-Col  Late-Col  Excess-Col  Carri-Sen  Runts  Giants
Gi4/3   0          0         0         0          0         0      0
```

BPDU可能由物理埠接收，但仍無法到達STP進程。如果上兩個示例中使用的命令顯示收到某些組播且未計算錯誤，則檢查是否在STP進程級別丟棄BPDU。在Catalyst 6500上發出remote command switch test spanning-tree process-statscommand：

```
<#root>
```

```
cat#
```

```
remote command switch test spanning-tree process-stats
```

```
-----TX STATS-----
transmission rate/sec      = 2
paks transmitted           = 5011226
paks transmitted (opt)     = 0
opt chunk alloc failures  = 0
max opt chunk allocated    = 0
-----RX STATS-----

receive rate/sec          = 1
```

```

paks received at stp isr = 3947627
paks queued at stp isr   = 3947627

paks dropped at stp isr   = 0
drop rate/sec            = 0

paks dequeued at stp proc = 3947627
paks waiting in queue    = 0
queue depth              = 7(max) 12288(total)
-----PROCESSING STATS-----
queue wait time (in ms)  = 0(avg) 540(max)
processing time (in ms)  = 0(avg) 4(max)
proc switch count       = 100
add vlan ports          = 20
time since last clearing = 2087269 sec

```

本示例中使用的命令顯示了STP進程的統計資訊。請務必確認丟棄計數器不會增加，而接收的資料包確實會增加。如果未增加接收的資料包，但物理埠確實接收了多播，請驗證交換機帶內介面（CPU介面）是否接收了資料包。發出remote command switch show ibc | i Catalyst 6500/6000上的rx_inputcommand：

```


<#root>
cat#
remote command switch show ibc | i rx_input

rx_inputs=
5626468
, rx_cumbytes=859971138
cat#
remote command switch show ibc | i rx_input

rx_inputs=
5626471
, rx_cumbytes=859971539

```

此範例顯示，在兩次輸出之間，頻內連線埠已接收23個封包。

 注意：這23個資料包不只是BPDU資料包；它是帶內埠收到的所有資料包的全局計數器。

如果沒有指示本地交換機或埠上丟棄了BPDU，您必須移至鏈路另一端的交換機，並驗證該交換機是否傳送了BPDU。檢查BPDU是否定期在非根指定埠上傳送。如果埠角色同意，則埠傳送BPDU，但鄰居不接收BPDU。檢查BPDU是否已傳送。發出show spanning-tree interface interface detailcommand：

<#root>

cat#

```
show spanning-tree interface g3/1 detail
```

Port 129 (GigabitEthernet3/1) of MST00 is

designated

forwarding

Port path cost 20000, Port priority 128, Port Identifier 128.129.
Designated root has priority 0, address 0007.4f1c.e847
Designated bridge has priority 32768, address 00d0.003f.8800
Designated port id is 128.129, designated path cost 2000019
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 0
Link type is point-to-point by default, Internal
Loop guard is enabled by default on the port

BPDUs: sent 1774

, received 1

cat#

```
show spanning-tree interface g3/1 detail
```

Port 129 (GigabitEthernet3/1) of MST00 is

designated


forwarding

Port path cost 20000, Port priority 128, Port Identifier 128.129.
Designated root has priority 0, address 0007.4f1c.e847
Designated bridge has priority 32768, address 00d0.003f.8800
Designated port id is 128.129, designated path cost 2000019
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 0
Link type is point-to-point by default, Internal
Loop guard is enabled by default on the port

BPDUs: sent 1776

, received 1

在本示例中，在輸出之間傳送兩個BPDU。

 **注意：**STP進程維護BPDU: sentcounter。這表示計數器表示BPDU已傳送到物理埠並已傳送。檢查傳輸的多播資料包的埠計數器是否增加。發出show interface interface counterscommand。這有助於確定BPDU的流量。

<#root>

cat#

```
show interface g3/1 counters
```

Port	InOctets	InUcastPkts	InMcastPkts	InBcastPkts
Gi3/1	127985312	83776	812319	19

Port	OutOctets	OutUcastPkts
------	-----------	--------------

OutMcastPkts

Port	OutBcastPkts	OutMcastPkts
Gi3/1	131825915	3442

872342

386

cat#

```
show interface g3/1 counters
```

Port	InOctets	InUcastPkts	InMcastPkts	InBcastPkts
Gi3/1	127985312	83776	812319	19

Port	OutOctets	OutUcastPkts
------	-----------	--------------

OutMcastPkts

Port	OutBcastPkts	OutMcastPkts
Gi3/1	131826447	3442

872346

386

透過所有這些步驟，我們可以找到未接收、傳送或處理BPDU的交換機或鏈路。STP可能已計算埠的正確狀態，但由於控制平面問題，它無法在轉發硬體上設定此狀態。如果埠在硬體級別沒有被阻塞，則會建立環路。如果您認為在網路中存在此問題，請[聯絡Cisco技術支援](#)以獲取進一步幫助。

5. 恢復冗餘

找到導致環路的裝置或鏈路後，必須將該裝置與網路隔離，否則必須解決問題。（例如更換光纖或GBIC）。必須恢復步驟3中斷開的冗餘鏈路。


請勿操作導致環路的裝置或鏈路，這一點很重要，因為導致環路的許多情況都是暫時的、間歇性的、不穩定的。這意味著，如果在調查中或調查後清除該情況，則此情況暫時不會發生，或者根本不會發生。必須記錄該情況，以便[Cisco技術支援](#)可以進一步調查。在重設交換器之前，請務必收集有關情況的資訊。如果條件消失，則無法確定環路的根本原因。如果收集資訊，請確保此問題不會再次導致環路。有關詳細資訊，請參閱[防止網路發生轉發環路](#)。

調查拓撲更改

拓撲更改(TC)機制的作用是在拓撲更改後糾正L2轉發表。這是避免連線中斷所必需的，因為以前可以透過特定埠訪問的MAC地址可能會更改並且可以透過不同的埠訪問。TC縮短了TC發生VLAN中所有交換機的轉發表時間。因此，如果未重新獲知地址，該地址將老化，並發生泛洪以確保資料包到

達目的MAC地址。

埠的STP狀態與STPforwardingstate之間發生更改時將觸發TC。在TC之後，即使特定目的MAC地址已過期，泛洪也不會持續很長時間。該地址由來自MAC地址已老化主機的第一個資料包重新獲取。當TC在較短的間隔內重複發生時，可能會出現此問題。交換機不斷快速老化其轉發表，因此泛洪幾乎可以保持恆定。

 **注意：**使用快速STP或多STP (IEEE 802.1w和IEEE 802.1s)，TC由埠狀態更改為 forwarding以及角色更改(從designatedtoroot)觸發。使用快速STP時，L2轉發表會立即刷新，而802.1d則會縮短老化時間。轉發表立即刷新可更快地恢復連線，但可能會導致更多泛洪

在配置良好的網路中，TC是一個罕見的事件。當交換機埠上的某條鏈路接通或斷開時，一旦該埠的STP狀態更改為forwarding，便會最終發生TC。當埠擺動時，這將導致重複的TC和泛洪。

啟用了STP portfast功能的埠在TC轉為/轉出theforwarding狀態時不會導致TC。在所有終端裝置埠 (如印表機、PC和伺服器) 上配置portfast可以將TC限制在低數量，強烈建議這樣做。

如果網路上存在重複的TC，您必須確定這些TC的來源並採取措施減少這些TC，以最大限度地減少泛洪。

使用802.1d時，有關TC事件的STP資訊透過TC通知(TCN)在網橋之間傳播，TCN是一種特殊型別的BPDU。如果您按照接收TCN BPDU的埠進行操作，則可以找到產生TC的裝置。

找出泛洪的原因

您可以確定存在由低效能、鏈路上不應擁塞的資料包丟棄，以及資料包分析器向不在本地網段中的同一目標顯示多個單播資料包。有關單播泛洪的詳細資訊，請參閱[交換式園區網路中的單播泛洪](#)。

在執行Cisco IOS軟體的Catalyst 6500/6000上，您可以檢查轉送引擎計數器 (僅在Supervisor 2引擎上) 來估計泛洪量。發出remote command switch show earl statistics | i MISS_DA|ST_FR命令：

```
<#root>
```

```
cat#
```

```
remote command switch show earl statistics | i MISS_DA|ST_FR
```

```
ST_MISS_DA      =      18          530308834
ST_FRMS         =      97          969084354
```

```
cat#
```

```
remote command switch show earl statistics | i MISS_DA|ST_FR
```

```
ST_MISS_DA      =         4          530308838
ST_FRMS         =      23          969084377
```

在此範例中，第一欄顯示自上次執行此命令以來的變更，第二欄顯示自上次重新開機之後的累計值

。第一行顯示泛洪幀的數量，第二行顯示處理的幀的數量。如果兩個值相近，或者第一個值以較高的速率增加，則可能是交換機正在泛洪流量。但是，這只能與其他驗證泛洪的方法結合使用，因為計數器不是精細的。每台交換機有一個計數器，而不是每個埠或VLAN。看到一些泛洪資料包是正常的，因為如果目標MAC地址不在轉發表中，交換機總是可能會泛洪。當交換機接收到目的地址尚未獲知的資料包時，可能出現這種情況。

尋找TC的來源

如果已知發生過度泛洪的VLAN的VLAN編號，請檢查STP計數器以檢視TC的數量是否高或定期增加。發出show spanning-tree vlan vlan-id detail command (在本例中使用的是VLAN 1) ：

```
<#root>
```

```
cat#
```

```
show spanning-tree vlan 1 detail
```

```
VLAN0001 is executing the ieee compatible Spanning Tree protocol  
Bridge Identifier has priority 32768, sysid 1, address 0007.0e8f.04c0  
Configured hello time 2, max age 20, forward delay 15  
Current root has priority 0, address 0007.4f1c.e847  
Root port is 65 (GigabitEthernet2/1), cost of root path is 119  
Topology change flag not set, detected flag not set
```


```
Number of topology changes 1 last change occurred 00:00:35 ago  
from GigabitEthernet1/1
```

```
Times: hold 1, topology change 35, notification 2  
hello 2, max age 20, forward delay 15  
Timers: hello 0, topology change 0, notification 0, aging 300
```

如果VLAN編號未知，您可以使用資料包分析器或檢查所有VLAN的TC計數器。

採取措施防止過度的TC

您可以監控topology changes counter的數量以檢視它是否定期增加。然後，移至連線到所示埠的網橋，接收最後一個TC (在前面的示例中，為GigabitEthernet1/1埠)，檢視該網橋的TC來自何處。必須重複此過程，直至找到未啟用STP portfast的終端站埠，或者直至找到需要修復的抖動鏈路。如果TC來自其他來源，則需要重複整個過程。如果鏈路屬於終端主機，則可以配置portfast功能以防止生成TC。

 注意：在Cisco IOS軟體STP實施中，只有當VLAN中的埠接收到TCN BPDU時，TC的計數器才能增加。如果接收到具有設定TC標誌的正常配置BPDU，則TC計數器不會遞增。這意味著，如果懷疑某個TC是泛洪的原因，請開始從該VLAN中的STP根網橋跟蹤TC的源。它可以獲得有關TC的數量和來源的最準確資訊。

解決收斂時間相關問題

在某些情況下，STP的實際操作與預期行為不匹配。以下是兩個最常見的問題：

- STP收斂或重新收斂所需的時間比預期長。
- 拓撲結果不同於預期。


通常，以下是該行為的原因：

- 實際拓撲與記錄的拓撲不匹配。
- 配置錯誤，如STP計時器配置不一致、STP直徑增加或portfast配置錯誤。
- 在收斂或重新收斂期間交換機CPU過載。
- 軟體缺陷。

如前所述，由於可能會影響STP的各種問題，本文檔只能提供故障排除的一般指導原則。要瞭解收斂時間為什麼比預期長，請檢視STP事件序列，瞭解發生的情況和順序。由於Cisco IOS軟體中的STP實施不會記錄結果(除特定事件(如埠不一致))，因此您可以使用Cisco IOS軟體對STP進行調試以便更清楚地檢視。對於STP，使用運行Cisco IOS軟體的Catalyst 6500/6000，處理在交換機處理器(SP)(或Supervisor)上完成，因此需要在SP上啟用調試。對於Cisco IOS軟體網橋組，處理在路由處理器(RP)上完成，因此需要在RP(MSFC)上啟用調試。

使用STP Debug命令

許多STPdebug命令適用於開發工程。如果沒有詳細瞭解Cisco IOS軟體中的STP實施，則他們無法提供任何對他人有意義的輸出。某些調試可以提供可立即讀取的輸出，例如埠狀態更改、角色更改、事件(如TC)以及已接收和已傳輸BPDU的轉儲。本部分並不提供所有調試的完整說明，而是簡要介紹最常用的調試。

 **注意：**使用debug命令時，請啟用最低限度的必要調試操作。如果不需要進行即時調試，請將輸出記錄到日誌中，而不是列印到控制檯。過度調試會使CPU過載並中斷交換機運行。

要將調試輸出定向到日誌而非控制檯或Telnet會話，請在全局配置模式下發出logging console informational and no logging monitor commands。要檢視一般事件日誌，請對每VLAN生成樹(PVST)和快速PVST發出debug spanning-tree event command。這是提供有關STP發生情況的第一個調試。在多生成樹(MST)模式下，發出debug spanning-tree event command不起作用。因此，請發出debug spanning-tree mstp roles命令以檢視埠角色更改。要檢視埠STP狀態更改，請將debug spanning-tree switch state command與the debug pm vpc command一起發出：

```
<#root>
```

```
cat-sp#
```

```
debug spanning-tree switch state
```

```
Spanning Tree Port state changes debugging is on
```


cat-sp#

debug pm vp

Virtual port events debugging is on

Nov 19 14:03:37: SP: pm_vp 3/1(333): during state forwarding, got event 4(remove)

Nov 19 14:03:37: SP:

@@@

pm_vp 3/1(333):

forwarding -> notforwarding

port 3/1 (was forwarding) goes down in vlan 333

Nov 19 14:03:37: SP: *** vp_fwdchange: single: notfwd: 3/1(333)

Nov 19 14:03:37: SP: @@@ pm_vp 3/1(333): notforwarding -> present

Nov 19 14:03:37: SP: *** vp_linkchange: single: down: 3/1(333)

Nov 19 14:03:37: SP: @@@ pm_vp 3/1(333): present -> not_present

Nov 19 14:03:37: SP: *** vp_statechange: single: remove: 3/1(333)

Nov 19 14:03:37: SP: pm_vp 3/2(333): during state notforwarding,
got event 4(remove)

Nov 19 14:03:37: SP:

@@@

pm_vp 3/2(333): notforwarding -> present

Nov 19 14:03:37: SP: *** vp_linkchange: single: down: 3/2(333)

Port 3/2 (was not forwarding) in vlan 333 goes down

Nov 19 14:03:37: SP: @@@ pm_vp 3/2(333): present -> not_present

Nov 19 14:03:37: SP: *** vp_statechange: single: remove: 3/2(333)

Nov 19 14:03:53: SP: pm_vp 3/1(333): during state not_present,
got event 0(add)

Nov 19 14:03:53: SP: @@@ pm_vp 3/1(333): not_present -> present

Nov 19 14:03:53: SP: *** vp_statechange: single: added: 3/1(333)

Nov 19 14:03:53: SP: pm_vp 3/1(333): during state present,
got event 8(linkup)

Nov 19 14:03:53: SP:

@@@

pm_vp 3/1(333): present ->
notforwarding

Nov 19 14:03:53: SP: STP SW: Gi3/1 new blocking req for 0 vlans

Nov 19 14:03:53: SP: *** vp_linkchange: single: up: 3/1(333)

Port 3/1 link goes up and blocking in vlan 333

Nov 19 14:03:53: SP: pm_vp 3/2(333): during state not_present,
got event 0(add)

Nov 19 14:03:53: SP: @@@ pm_vp 3/2(333): not_present -> present

Nov 19 14:03:53: SP: *** vp_statechange: single: added: 3/2(333)

Nov 19 14:03:53: SP: pm_vp 3/2(333): during state present,
got event 8(linkup)

```
Nov 19 14:03:53: SP:
```

```
@@@
```

```
pm_vp 3/2(333): present ->
```

```
notforwarding
```

```
Nov 19 14:03:53: SP: STP SW: Gi3/2 new blocking req for 0 vlans
```

```
Nov 19 14:03:53: SP: *** vp_linkchange: single: up: 3/2(333)
```

```
Port 3/2 goes up and blocking in vlan 333
```

```
Nov 19 14:04:08: SP: STP SW: Gi3/1 new learning req for 1 vlans
```

```
Nov 19 14:04:23: SP: STP SW: Gi3/1 new forwarding req for 0 vlans
```

```
Nov 19 14:04:23: SP: STP SW: Gi3/1 new forwarding req for 1 vlans
```

```
Nov 19 14:04:23: SP: pm_vp 3/1(333): during state notforwarding,  
got event 14(forward_notnotify)
```

```
Nov 19 14:04:23: SP:
```

```
@@@ pm_vp 3/1(333): notforwarding ->
```

```
forwarding
```

```
Nov 19 14:04:23: SP: *** vp_list_fwdchange: forward: 3/1(333)
```

```
Port 3/1 goes via learning to forwarding in vlan 333
```

要瞭解STP以某種方式運行的原因，檢視交換機接收和傳送的BPDU通常很有用：

```
<#root>
```

```
cat-sp#
```

```
debug spanning-tree bpdu receive
```

```
Spanning Tree BPDU Received debugging is on
```

```
Nov 6 11:44:27: SP: STP: VLAN1 rx BPDU: config protocol = ieee,  
packet from GigabitEthernet2/1 , linktype IEEE_SPANNING ,  
enctype 2, encsize 17
```

```
Nov 6 11:44:27: SP: STP: enc 01 80 C2 00 00 00 00 06 52 5F 0E 50 00 26 42 42 03
```

```
Nov 6 11:44:27: SP: STP: Data 000000000000000074F1CE8470000001380480006525F0E4  
080100100140002000F00
```

```
Nov 6 11:44:27: SP: STP: VLAN1 Gi2/1:0000 00 00 00 000000074F1CE847 00000013  
80480006525F0E40 8010 0100 1400 0200 0F00
```

此調試適用於PVST、快速PVST和MST模式；但它不會解碼BPDU的內容。但是，您可以使用它確保收到BPDU。要檢視BPDU的內容，請對PVST和快速PVST將debug spanning-tree switch rx decodecommand與debug spanning-tree switch rx processcommand一起發出。發出debug spanning-tree mstp bpdu-rxcommand檢視MST的BPDU的內容：

```
<#root>
```

```
cat-sp#
```

```
debug spanning-tree switch rx decode
```

Spanning Tree Switch Shim decode received packets debugging is on

cat-sp#

```
debug spanning-tree switch rx process
```

Spanning Tree Switch Shim process receive bpdu debugging is on

```
Nov 6 12:23:20: SP: STP SW: PROC RX: 0180.c200.0000<-0006.525f.0e50 type/len 0026
Nov 6 12:23:20: SP:      encap SAP linktype ieee-st vlan 1 len 52 on v1 Gi2/1
Nov 6 12:23:20: SP:      42 42 03 SPAN
Nov 6 12:23:20: SP:      CFG P:0000 V:00 T:00 F:00 R:0000 0007.4f1c.e847 00000013
Nov 6 12:23:20: SP:      B:8048 0006.525f.0e40 80.10 A:0100 M:1400 H:0200 F:0F00

Nov 6 12:23:22: SP: STP SW: PROC RX: 0180.c200.0000<-0006.525f.0e50 type/len 0026
Nov 6 12:23:22: SP:      encap SAP linktype ieee-st vlan 1 len 52 on v1 Gi2/1
Nov 6 12:23:22: SP:      42 42 03 SPAN
Nov 6 12:23:22: SP:      CFG P:0000 V:00 T:00 F:00 R:0000 0007.4f1c.e847 00000013
Nov 6 12:23:22: SP:      B:8048 0006.525f.0e40 80.10 A:0100 M:1400 H:0200 F:0F00
```

對於MST模式，您可以使用thisdebugcommand：

<#root>

cat-sp#

```
debug spanning-tree mstp bpdu-rx
```

Multiple Spanning Tree Received BPDUs debugging is on

```
Nov 19 14:37:43: SP: MST:BPDU DUMP [
```

```
rcvd_bpdu Gi3/2
```

```
Repeated]
```


```
Nov 19 14:37:43: SP: MST:  Proto:0 Version:3 Type:2 Role: DesgFlags[ F ]
Nov 19 14:37:43: SP: MST:  Port_id:32897 cost:2000019
Nov 19 14:37:43: SP: MST:  root_id :0007.4f1c.e847 Prio:0
Nov 19 14:37:43: SP: MST:  br_id   :00d0.003f.8800 Prio:32768
Nov 19 14:37:43: SP: MST:  age:2 max_age:20 hello:2 fwdelay:15
Nov 19 14:37:43: SP: MST:  V3_len:90 PathCost:30000 region:STATIC rev:1
Nov 19 14:37:43: SP: MST:  ist_m_id :0005.74
Nov 19 14:37:43: SP: MST:BPDU DUMP [
```

```
rcvd_bpdu Gi3/2
```

```
Repeated]
```

```
Nov 19 14:37:43: SP: MST:  Proto:0 Version:3 Type:2 Role: DesgFlags[ F ]
Nov 19 14:37:43: SP: MST:  Port_id:32897 cost:2000019
Nov 19 14:37:43: SP: MST:  root_id :0007.4f1c.e847 Prio:0
Nov 19 14:37:43: SP: MST:  br_id   :00d0.003f.8800 Prio:32768
Nov 19 14:37:43: SP: MST:  age:2 max_age:20 hello:2 fwdelay:15
Nov 19 14:37:43: SP: MST:  V3_len:90 PathCost:30000 region:STATIC rev:1
Nov 19 14:37:43: SP: MST:  ist_m_id :0005.7428.1440 Prio:32768 Hops:18
Num Mrec: 1
Nov 19 14:37:43: SP: MST: stci=3 Flags[ F ] Hop:19 Role:Desg [Repeated]
Nov 19 14:37:43: SP: MST:      br_id:00d0.003f.8800 Prio:32771 Port_id:32897
Cost:2000028.1440 Prio:32768 Hops:18 Num Mrec: 1
```

Nov 19 14:37:43: SP: MST: stci=3 Flags[F] Hop:19 Role:Desg [Repeated]
Nov 19 14:37:43: SP: MST: br_id:00d0.003f.8800 Prio:32771 Port_id:32897
Cost:20000

 注意：對於Cisco IOS軟體版本12.1.13E及更高版本，支援STP的條件調試。這意味著您可以調試每個埠或每個VLAN上接收或傳輸的BPDU。

發出debug condition vlan vlan_num ordebug condition interface interface 命令，可將調試輸出的範圍限制為每個介面或每個VLAN。

保護網路免受轉發環路的影響

Cisco開發了許多功能和增強功能，以在STP無法管理某些故障時保護網路免受轉發環路的影響。

當您排除STP故障時，它有助於隔離特定故障並可能找到其原因，而實施這些增強功能是保護網路免受轉發環路影響的唯一方法。

以下是保護網路免受轉發環路影響的方法：


1. 在所有交換器到交換器連結上啟用單向連結偵測(UDLD)

有關UDLD的詳細資訊，請參閱[瞭解和配置單向鏈路檢測協定功能](#)。


2. 在所有交換機上啟用環路防護

有關環路防護的詳細資訊，請參閱[使用環路防護和BPDU遲滯檢測功能的生成樹協定增強功能](#)。

啟用後，UDLD和環路防護可消除導致轉發環路的大部分原因。出現故障的鏈路（或依賴於故障硬體的所有鏈路）關閉或被阻塞，而不是建立轉發環路。


 注意：雖然這兩個功能看起來有些冗餘，但每個功能都有其獨特的功能。因此，同時使用這兩個功能可提供最高級別的保護。有關UDLD和環路防護的詳細比較，請參閱[環路防護與單向鏈路檢測](#)。

對於必須使用主動還是常規UDLD，有不同的意見。與普通模式UDLD相比，主動UDLD無法提供更強的環路保護。主動UDLD會檢測埠停滯情況（當鏈路處於工作狀態但沒有相關的流量黑洞）。此新增功能的缺點是，如果沒有一致的失敗，主動UDLD可能會停用連結。通常，使用者會混淆UDLDhellointerval的修改與主動UDLD功能。這是不正確的。計時器可以在兩種UDLD模式下修改。

 注意：在極少數情況下，主動UDLD會關閉所有上行鏈路埠，這實際上會將交換機與網路的其餘部分隔離。例如，當兩台上游交換機的CPU使用率都極高，並且都使用主動模式UDLD時，就可能會出現這種情況。因此，建議您配置交換機不具有帶外管理功能的超時，即無法消除。

3. 在所有終端站連線埠上啟用Portfast

您必須啟用portfast來限制TC和後續泛洪的數量，這可能會影響網路效能。請僅對連線到終端站的埠使用此命令。否則，意外拓撲環路可能導致資料包環路，並中斷交換機和網路運行。

 **注意：**使用no spanning-tree portfast命令時要小心。此命令僅刪除任何埠特定portfast命令。如果您在全局配置模式下定義spanning-tree portfast default命令，並且埠不是中繼埠，此命令會隱式啟用portfast。如果不全局配置portfast，no spanning-tree portfast 命令與spanning-tree portfast disable 命令等效。

4. 在兩側（如果支援）和Non-silent 選項上將EtherChannel設定為Desirable 模式

Desirablemode可以啟用埠聚合協定(PAgP)，以確保通道對等體之間的運行時一致性。這提供了額外的環路保護級別，特別是在通道重新配置期間（例如鏈路加入或離開通道時，以及鏈路故障檢測）。內建的通道錯誤配置防護預設啟用，可防止由於通道錯誤配置或其他情況而導致的轉發環路。有關此功能的詳細資訊，請參閱[瞭解EtherChannel不一致檢測](#)。

5. 不要停用交換器對交換器連結上的自動交涉（如果支援）

自動交涉機制可以傳遞遠端錯誤資訊，這是偵測遠端故障的最快方式。如果在遠端端檢測到故障，即使鏈路收到脈衝，本地端也會關閉鏈路。相較於UDLD等高階偵測機制，自動交涉速度極快（在微秒內），但缺少UDLD的端對端涵蓋範圍（例如整個資料路徑：CPU—轉送邏輯—連線埠1—連線埠2—轉送邏輯—CPU與連線埠1—連線埠2）。主動UDLD模式在故障檢測方面提供與自動協商類似的功能。當連結的兩端都支援交涉時，不需要啟用主動模式UDLD。

6. 調整STP計時器時要格外小心

STP計時器相互依賴並且取決於網路拓撲。對計時器進行任意修改時，STP無法正常工作。有關STP計時器的詳細資訊，請參閱[瞭解和調整生成樹協定計時器](#)。

7. 如果可能發生拒絕服務攻擊，請透過根防護保護網路STP周邊

根防護和BPDU防護允許您保護STP免受外部影響。如果有可能發生此類攻擊，則必須使用根防護和BPDU防護來保護網路。有關根防護和BPDU防護的詳細資訊，請參閱以下文檔：

- [生成樹通訊協定根防護增強功能](#)
- [跨距樹狀目錄 PortFast BPDU 防護增強功能](#)

8. 在啟用了Portfast的埠上啟用BPDU防護，以防止STP受到連線到埠的未授權網路裝置（如集線器、交換機和橋接路由器）的影響

如果正確配置根防護，它將防止STP受到外界的影響。如果啟用了BPDU防護，它會關閉接收任何BPDU的埠。這在調查事件時非常有用，因為BPDU防護會生成系統日誌消息並關閉埠。如果根或BPDU防護不能防止短週期環路，則兩個快速啟用的埠直接連線或透過集線器連線。

9. 避免管理VLAN上的使用者流量

管理VLAN包含在一個構建塊中，而不是整個網路。

交換機管理介面接收管理VLAN上的廣播資料包。如果出現過多的廣播（例如廣播風暴或應用程式出現故障），交換機CPU可能會超載，從而可能扭曲STP操作。

10. 可預測（硬編碼）的STP根和備份STP根放置

必須配置STP根和備份STP根，以便在出現故障時以可預測的方式進行收斂，並在每個場景中構建最佳拓撲。請勿將STP優先順序保留為預設值，以防止不可預知的根交換機選擇。

相關資訊

- [LAN 產品支援](#)
- [LAN 交換技術支援](#)

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。