# 运行 Cisco IOS 系统软件的 Catalyst 交换机上的 STP 故障排除

## 目录



## 简介

本文档提供了有关使用 Cisco IOS® 软件解决与生成树协议 (STP) 相关的问题的指南。其中有些特定命令仅适用于 Catalyst 6500/6000；然而，您可以将大多数原则应用于运行 Cisco IOS 软件的任何 Cisco Catalyst 交换机。

大多数 STP 故障排除主要针对以下三个问题：

- 转发环路
- 过度泛洪由于STP拓扑高速率更改(TC)
- 与收敛时间有关的问题

由于桥接没有任何机制可以跟踪某个数据包是否正在多次转发（例如，IP 存活时间 [TTL] 用于丢弃在网络中循环太久的流量），因此同一个第二层 (L2) 域中的两个设备之间只能存在一条路径。

STP 旨在根据 STP 算法阻塞冗余的端口，从而将冗余物理拓扑解析为树形拓扑。如果未阻塞冗余拓扑中的任何端口，并且在循环中无限地转发流量，则会出现转发环路（如 STP 环路）。

当转发环路启动后，可能会拥塞其路径上带宽最低的链路（如果所有链路的带宽相同，则所有链路都有可能被拥塞）。这种拥塞将造成数据包丢失，并导致受影响的 L2 域中出现网络中断情况。

对于过度泛洪，这些症状可能不会很明显。某些低速链路可能因泛洪流量而变得拥塞，因此使用这些拥塞链路的设备或用户可能会遇到连接缓慢或完全中断的情况。

## 先决条件

## 要求

Cisco 建议您了解以下主题：

- 各种生成树类型以及如何配置这些类型。有关详细信息，请参阅配置 STP 和 IEEE 802.1s MST。
- 各种生成树功能以及如何配置这些功能。有关详细信息，请参阅配置 STP 功能。

## 使用的组件

本文档中的信息基于以下软件和硬件版本：

- 带有 Supervisor 2 引擎的 Catalyst 6500
- Cisco IOS 软件版本 12.1(13)E

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

## 规则

有关文档规则的详细信息，请参阅 Cisco 技术提示规则。

# STP 为何发生故障

STP 对其运行环境进行了某些假定。以下假定与本文档的相关程度最高：

- 两个网桥之间的每条链路都是双向的。这意味着，如果 A 直接连接到 B，则 A 将接收 B 发送的内容，B 将接收 A 发送的内容，条件是两者之间的链路是连通的。
- 运行STP的每网桥能收到，有规律地处理和传送STP网桥协议数据单元(BPDU)，亦称STP数据包。

尽管这些假定似乎符合逻辑且很明显，但并不符合有些情况。其中多数情况都涉及某类硬件问题；然而，软件缺陷也可能导致 STP 故障。各种硬件故障、配置错误或布线错误可导致大多数 STP 故障，而只有少数 STP 故障是因为软件故障所致。由于交换机之间存在多余的额外连接，也可能出现 STP 故障。VLAN 将因这些额外的连接而进入关闭状态。要解决此问题，请删除交换机之间所有不需要的连接。

如果未满足其中一项假定，一个或多个网桥可能不会再接收或处理 BPDU。这意味着该网桥（或多个网桥）将无法发现网络拓扑。如果不了解正确的拓扑，交换机便不能阻塞环路。因此，泛洪流量将会基于循环拓扑进行循环、占用所有带宽并使网络瘫痪。

交换机无法接收 BPDU 的原因示例包括收发器或千兆位接口转换器 (GBIC) 故障、布线问题或者端口、板卡或 Supervisor 引擎出现硬件故障。导致 STP 故障的一个常见原因是网桥之间存在一条单向链路。在这种情况下，一个网桥发送 BPDU，但下游网桥从未收到它们。由于交换机无法处理收到的 BPDU，因此 STP 处理也可能因为 CPU 过载（99% 或更高）而中断。BPDU 可能在从一个网桥到另一个网桥的路径中损坏，这样也会阻止适当的 STP 行为。

除转发环路外，如果未阻塞任何端口，则还会出现只有某些数据包通过阻塞端口错误转发的情况。在大多数情况下，这是由软件问题造成的。此类行为可能会导致“慢速环路”。这意味着一些数据包已进入循环中，但多数流量仍在流经网络，因为链路很可能没有拥塞。

本文档的剩余部分将提供有关对大多数与 STP 相关的常见问题进行故障排除的指南。

## 转发循环故障排除

转发环路在其源（原因）和影响方面差别很大。由于可能影响 STP 的问题有多种，因此本文档只能提供有关如何排除转发环路故障的一般指南。

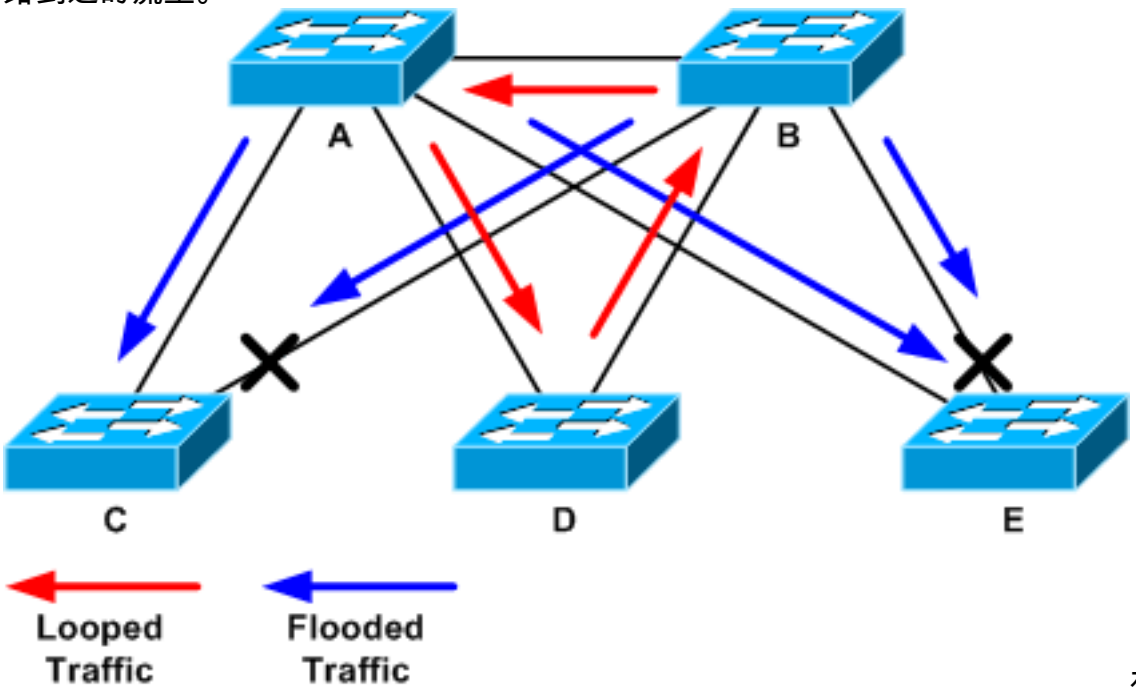
开始排除故障之前，您必须获取以下信息：

- 详述所有交换机和网桥的实际拓扑图
- 与这些交换机和网桥对应的（互连）端口号
- STP 配置详细信息，例如哪台交换机是根和备份根、哪些链路具有非默认成本或优先级，以及阻塞端口的位置

通常，故障排除涉及以下步骤（根据具体情况的不同，可能并不需要执行某些步骤）：

1. 确定环路。网络中形成转发环路时，通常会出现以下现象：来自、通往或经过受影响区域的连接损耗在路由器连接的高CPU利用率对可能导致多种症状，例如路由协议邻接飘荡或热备份路由协议(HSRP)活动路由器飘荡的受影响的分段或VLAN链路利用率极高（往往达到 100%）交换机背板利用率极高（与基线利用率相比）Syslog 消息指示网络中存在数据包循环（例如，HSRP IP 地址重复）Syslog 消息显示持续重复获知地址或 MAC 地址摆动消息许多接口上输出丢弃的数量不断增加**注意：** 其中任何一种原因单独可能指示不同的问题（或根本不指示问题）。然而，如果同时观察到其中多种原因，则表明网络中很可能形成了转发环路。**注意：** 验证这一点的最快方式是检查交换机背板流量利用率：`cat# show catalyst6000 traffic-meter traffic meter = 13% Never cleared peak = 14% reached at 12:08:57 CET Fri Oct 4 2002`
**注意：** 使用 Cisco IOS 软件的 Catalyst 4000 当前不支持此命令。如果当前流量级别远高于正常级别或者不清楚基准级别，请检查最近是否达到了峰值级别，以及峰值级别是否接近当前流量级别。例如，如果峰值流量级别为 15%，并且是在两分钟前刚刚达到的，而当前流量级别为 14%，则可能表明交换机正在非常高的负载下运行。如果流量负载处于正常级别，则可能表明不存在环路或环路中未涉及此设备。然而，此设备仍可能包含在一个慢速环路中。
2. 查明环路的拓扑（范围）。一旦确定网络中断的原因是转发环路，首先采取的操作就是停止该环路并恢复网络运行。要停止该环路，您必须知道该环路中涉及哪些端口：观察哪些端口的链路利用率（每秒数据包数）最高。**show interface** Cisco IOS 软件命令可以显示每个接口的利用率。要仅显示利用率信息和接口名称（用于快速分析），您可以使用 Cisco IOS 软件正则表达式输出过滤。请发出 **show interface|include line|\/sec** 命令，以便仅显示每秒数据包统计信息和接口名称：`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` 请特别注意链路利用率最高的那些接口。在本示例中，这些接口是 g2/3、g2/4 和 g2/8；它们可能是环路中涉及的端口。
3. 断开环路。要断开环路，您必须先关闭或断开所涉及的端口。务必不仅要停止环路，而且还要

查找并确定环路的根本原因。断开环路相对容易一些。**注意：** 为便于后续进行原因分析，您不需要立即关闭或断开所有端口，而是应每次关闭一个端口。通常最好关闭受环路影响的聚合点（例如分配或核心交换机）的端口。如果立即关闭所有端口并逐个启用或者重新连接它们，则可能不起作用；重新连接冲突端口后，环路将被停止并可能无法立即启动。因此，很难将故障与任何特定端口关联起来。**注意：** 建议您在重新启动交换机断开环路之前收集信息。否则，后续进行的根本原因分析将非常困难。禁用或断开每个端口后，必须检查交换机背板利用率是否恢复到正常级别。**注意：** 请记住，通常情况下，一些端口并不是在维持环路，而是在泛洪随环路到达的流量。关闭此类泛洪端口时，只会略微降低背板利用率，而不会使环路停止。在下面的示例拓扑中，环路位于交换机 A、B 和 D 之间。因此，正在维持链路 AB、AD 和 BD。如果关闭其中任意一条链路，将会停止环路。链路 AC、AE、BC 和 BE 只是在泛洪随环路到达的流量。




在维持端口关闭后，背板利用率将降至正常值。务必记下是哪个端口的关闭使背板利用率（和其他端口的利用率）恢复到正常级别。此时，环路将被停止并且网络运行情况应该有所改善；然而，由于可能未确定环路的原始原因，因此仍可能有一些问题没有解决。

4. 找到形成环路的原因并将其修正。停止环路后，您需要确定环路开始的原因。由于这些原因可能会变化，因此这通常是该过程的最困难部分。制定在任何情况下都适用的确切步骤也很困难。不过，以下提供了一些一般指南：检查拓扑图以找到冗余路径。这包括在上一步中找到的回到同一交换机的维持端口（在循环期间获取路径数据包）。在上一个示例拓扑中，此路径是 AD-DB-BA。请检查该冗余路径上的每台交换机有无下列问题：该交换机是否知道正确的 STP 根交换机？L2 网络中的所有交换机都应该通过协商确定常见的 STP 根。如果网桥对特定 VLAN 或 STP 实例中的 STP 根一致显示不同 ID，则明确表明这是一个问题症状。请发出 **show spanning-tree vlan vlan-id** 命令，以显示给定 VLAN 的根网桥 ID：`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` 由于在前面步骤中确定了环路中涉及的端口，因此可以从端口中找到 VLAN 编号。如果相关端口是中继端口，则通常会涉及中继上的所有 VLAN。如果实际情况不是这样（例如，环路可能发生在单个 VLAN 上），则您可以尝试发出 **show interfaces|包括L2|line|broadcast**命令(仅在Supervisor2及以后引擎在Catalyst 6500/6000系列交换机，因为Supervisor 1不提供每VLAN交换统计数据)。请仅查看 VLAN 接口。交换数据包数量最多的 VLAN 将往往是出现环路的 VLAN：`cat# show int | 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 在本示例中，VLAN 1 占有的广播和 L2 交换流量的数量最多。是否能正确识别根端口？根端口到根网桥的成本应该最低（有时一条路径按跳数考虑较短，但按成本考虑则较长，因为低速端口的成本较高）。要确定哪个端口被视为给定 VLAN 的根端口，请发出 **show spanning-tree vlan vlan** 命令：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是否在根端口和应该阻塞的端口上定期接收 BPDU？BPDU 由根网桥每隔 hello interval 发送一次（默认情况下为两秒）。非根网桥将接收、处理、修改和传播从根网桥接收的 BPDU。请发出 **show spanning-tree interface interface detail** 命令，以查看是否正在接收 BPDU：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：** 已在该命令的两次输出之间收到一个 BPDU（计数器从 53 到 54）。显示的计数器实际上是 STP 进程本身维护的计数器。这意味着，如果接收计数器增加，则表示不仅物理端口收到了 BPDU，STP 进程也收到了 BPDU。如果 received BPDU 计数器在应该是根备用或备份端口的端口上没有增加，则检查该端口是否完全接收任何多播（BPDU 作为多播发送）。请发出 **show interface interface counters** 命令：cat# **show interface g3/2 counters** Port InOctets InUcastPkts **InMcastPkts** InBcastPkts Gi3/2 14873036 2 **89387** 0 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 端口角色的概要可在使用环路防护和 BPDU 迟滞检测功能的生成树协议增强功能的 STP 端口角色概要部分中找到。）如果未收到任何 BPDU，请检查端口是否在对错误进行计数。请发出 **show interface interface counters errors** 命令：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 Runs Giants Gi4/3 0 0 0 0 0 0 0 有可能物理端口收到了 BPDU，但这些 BPDU 仍未到达 STP 进程。如果前两个示例中使用的命令显示收到了一些多播，但错误数没有增加，则检查在 STP 进程级别是否正在丢弃 BPDU。请在 Catalyst 6500 上发出 **remote command switch test spanning-tree process-stats** 命令：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 在本示例中使用的命令显示了 STP 进程统计信息。务必验证丢弃计数器是否没有增加，而收到的数据包是否正在增加。如果收到的数据包没有增加，但物理端口正在接收多播，请验证数据包是否正由交换机带内接口（CPU 的接口）接收。请在 Catalyst 6500/6000 上发出 **remote command switch show ibc|i rx_input** 命令：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 本示例显示带内端口在两次输出之间收到了 23 个数据包。**注意：** 这 23 个数据包不只是 BPDU 数据包；这是带内端口收到的所有数据包的全局计数器。如果没有任何迹象表明正在本地交换机或端口上丢弃 BPDU，则您必须移至链路另一侧的交换机并验证该交换机是否在发送 BPDU。指定的非根端口是否经常发送 BPDU？如果根据端口角色，端口正在发送 BPDU，但是邻居没有收到它们，请检查是否在实际发送 BPDU。请发出 **show spanning-tree interface interface detail** 命令：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 **BPDU: 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 **BPDU: sent 1776**, received 1 在本示例中，已在输出之间发出两个 BPDU。**注意：** STP 进程维护 BPDU:sent 计数器。这意味着该计数器可指示已向物理端口发送要最终发出的 BPDU。请检查端口计数器是否会因传输的多播数据包而增加。请发出 **show interface interface counters** 命令。这可帮助确定是否会发出 BPDU：cat# **show interface g3/1 counters** Port InOctets InUcastPkts InMcastPkts InBcastPkts Gi3/1 127985312 83776 812319 19 Port OutOctets OutUcastPkts **OutMcastPkts** OutBcastPkts 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** OutBcastPkts Gi3/1 131826447 3442 **872346** 386 对于所有这些步骤，其思路都是查找没有接收、发送或处理 BPDU 的交换机或链路。有可能（但可能性不大）STP 计算了端口的正确状态，但是由于控制层面问题，无法在转发硬件上设置此状态。如果在硬件级别未阻塞推测的阻塞端口，则可能会创建环路。如果您怀疑网络中存在此类问题，请与 Cisco 技术支持联系以获取进一步帮助。

5. 恢复冗余能力。一旦发现有造成环路的设备或链路，则必须将此设备与网络隔离，或者采取相应措施来解决问题（例如更换光纤或 GBIC）。必须恢复在步骤 3 中断开的冗余链路。务必对造成环路的设备或链路进行尽可能少的处理，这是因为导致环路的许多情况可能非常短暂、断断续续而且不稳定。这意味着，如果在故障排除期间或之后清除该情况，则在这种情况再次出现之前可能需要一段时间。该情况还有可能根本不会再出现。应该尽力保留该情况，以便由 Cisco 技术支持进一步调查。务必在重置交换机之前收集有关该情况的信息。如果某个情况消失，通常便无法确定环路的根本原因。查找触发环路的设备或链路是一项重要工作，但是您需要确保同一类型的其他故障不会再引起环路。有关详细信息，请参阅本文档的防止网络发生转发环路部分。

## 排除过多的拓扑结构更改导致泛洪故障

TC 机制的作用是，在转发拓扑发生更改后更正 L2 转发表。这对于避免连接中断很有必要，因为在

TC 后，之前可以通过特定端口访问的一些 MAC 地址可能可通过其他端口访问。TC 可缩短发生 TC 的 VLAN 中所有交换机上的转发表老化时间；因此，如果没有重新识别该地址，它将会老化并且会发生泛洪以确保数据包到达目标 MAC 地址。

TC 通过在端口的 STP 状态和 STP 转发状态之间切换来触发。在 TC 后，即使特定目标 MAC 地址已老化，泛洪也不应持续很长时间。该地址将由来自 MAC 地址已老化的主机的第一个数据包重新识别。当 TC 在很短的间隔内重复发生时，可能会出现此问题。交换机会经常使其转发表快速老化，因此泛洪也几乎经常发生。

**注意：** 对于快速 STP 或多重 STP（IEEE 802.1w 和 IEEE 802.1s），TC 由端口状态更改为 forwarding，以及从已指定到根的角色更改触发。对于快速 STP，L2 转发表会立即刷新（与 802.1d 相对），从而缩短老化时间。立即刷新转发表可更快地恢复连接，但会导致更多泛洪。

如果网络配置良好，则应很少发生 TC 事件。当交换机端口上的某条链路接通或断开时，一旦该端口的 STP 状态在 forwarding 间切换，便会最终发生 TC。当端口抖动时，将导致重复的 TC 和泛洪。

启用了 STP portfast 功能的端口在转为/转出 forwarding 状态时，将不会导致 TC。portfast 在所有终端设备端口（例如打印机、PC 和服务器）上的配置应该将 TC 数限制为一个较低数量，强烈建议您这样做。有关 TC 的详细信息，请参阅了解生成树协议拓扑结构变化。

如果网络上有重复 TC，则您必须标识这些 TC 的来源并采取相应操作来减少它们，从而将泛洪降到最低。

对于 802.1d，有关 TC 事件的 STP 信息在网桥中通过 TC 通知 (TCN) 传播，TCN 是一种特殊类型的 BPDU。如果跟踪接收 TCN BPDU 的端口，则可以查找产生 TC 的设备。

### 确定泛洪是否由 STP TC 引起

通常，您可以基于以下情况确定存在泛洪：性能降低、不应该拥塞的链路上出现数据包丢弃，以及数据包分析程序显示多个单播数据包发送到不在本地网段上的同一目标）。

有关单播泛洪的详细信息，请参阅交换式园区网络中的单播泛洪。

在运行 Cisco IOS 软件的 Catalyst 6500/6000 上，您可以检查转发引擎计数器（仅针对 Supervisor 2 引擎）以估计泛洪数量。请发出 **remote command switch show earl statistics|i MISS_DA|ST_FR** 命令：

```
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
```

在本示例中，第一列显示自上次执行此命令以来的更改，第二列显示自上次重新启动以来的累计值。第一行显示泛洪帧的数量，第二行显示已处理的帧的数量。如果这两个值非常接近，或者第一个值在高速增加，则表明交换机可能正在泛洪流量。不过，由于计数器并不精细，因此这只能与验证泛洪的其他方式一起使用。每台交换机（非每个端口或 VLAN）有一个计数器。发现一些泛洪的数据包很正常，这是因为如果目标 MAC 地址不在转发表中，交换机将始终会泛洪。当交换机接收目标地址尚未识别的数据包时，就会出现这种情况。

### 跟踪 TC 的源

如果发生过度泛洪的 VLAN 的 VLAN 编号是已知的，请定期检查 STP 计数器以查看 TC 的数目是否很大或在增加。请发出 **show spanning-tree vlan vlan-id detail** 命令（在本示例中，使用的是

VLAN 1）：

```
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
```

如果不知道 VLAN 编号，您可以使用数据包分析程序或检查所有 VLAN 的 TC 计数器。

**采取步骤来防止 TC 过多**

您可以定期监控 number of topology changes 计数器以查看它是否在增加。然后，移到连接到所示端口的网桥，接收最后一个 TC（在前一个示例中为端口 GigabitEthernet1/1）并查看 TC 从该网桥的何处发出。必须重复此过程，直到找到未启用 STP portfast 的终端站端口，或者直到发现需要修复的抖动链路。如果 TC 仍然来自其他源，则需要重复整个过程。如果链路属于终端主机，您应该配置 portfast 功能以防止生成 TC。

**注意：** 在 Cisco IOS 软件 STP 实施中，如果 TCN BPDU 是由 VLAN 中的端口接收的，TC 的计数器将只会增加。如果收到带有 set TC 标志的正常配置 BPDU，则 TC 计数器不会增加。这意味着，如果您怀疑 TC 是造成泛洪的原因，则最好从该 VLAN 的 STP 根网桥开始跟踪 TC 的来源。这样可以获得有关 TC 的数量和来源的最准确信息。

# 排除收敛时间相关故障

有时会出现 STP 的实际操作与预期行为不匹配的情况。以下是两个最常见的问题：

- STP 收敛或再收敛比预期所需时间长。
- 生成的拓扑与预期的不同。

通常，此行为的原因如下：

- 实际拓扑和记录的拓扑之间不匹配
- 配置错误，例如 STP 计时器的配置不一致、超出 STP 直径或者 portfast 配置错误
- 收敛或再收敛期间交换机 CPU 过载
- 软件缺陷

如前文所述，由于可能影响 STP 的问题有多种，因此本文档只能提供用于故障排除的一般指南。

要了解收敛所需时间比预期长的原因，请查看 STP 事件序列以查明所发生的情况以及发生的顺序。由于 Cisco IOS 软件中的 STP 实施不具有特殊的日志记录（端口不一致等特定事件除外），因此您可以使用 Cisco IOS 软件 STP 调试功能来了解所发生的情况。

对于STP，用Catalyst 6500/6000，运行Cisco IOS软件的处理在交换机处理器(SP) (或Supervisor)执行，因此调试在SP需要启用。对于Cisco IOS软件网桥组，处理在路由处理器(RP)执行，因此调试在RP (MSFC)需要启用。

## STP 调试命令

许多 STP **debug** 命令适用于开发工程。对于没有详细了解 Cisco IOS 软件中的 STP 实施的某些用户，这些命令不会提供任何有意义的输出。部分调试命令可提供立即可读的输出，例如端口状态更改、角色更改、事件（例如 TC）以及已接收和传输的 BPDU 的转储。本部分没有完整介绍所有调

试命令，而是简单介绍了最常使用的调试命令。

**注意：** 使用 **debug** 命令时，请启用最低限度的必要调试操作。如果不需要实时调试，请将输出记录到日志中而不是将其打印到控制台。过多的调试可能会使 CPU 过载并中断交换机操作。要将调试输出定向到日志而非控制台或 Telnet 会话，请在全局配置模式下发出 **logging console informational** 和 no logging monitor 命令。

要查看一般事件日志，请对每 VLAN 生成树 (PVST) 和快速 PVST 发出 **debug spanning-tree event** 命令。这是可提供有关 STP 所发生的情况的一般信息的第一个调试。

在多生成树 (MST) 模式下，发出 **debug spanning-tree event** 命令不起作用。因此，请发出 **debug spanning-tree mstp roles** 命令以查看端口角色更改。

要查看端口 STP 状态更改，请将 **debug spanning-tree switch state** 命令与 debug pm vp 命令一起发出：

```
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
```

要了解 STP 以某种方式运行的原因，查看通过交换机接收和发送的 BPDU 通常很有用：

```
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 00000000000000074F1CE847000000138048000652 5F0E4 08010010014000 2000F00 Nov 6 11:44:27: SP: STP: VLAN1 Gi2/1:0000 00 00 00 000000074F1CE847 00000013 80480006525F0E40 8010 0100 1400 0200 0F00
```

此调试适用于 PVST、快速 PVST 和 MST 模式；但它不会对 BPDU 的内容进行解码。然而，您可以使用它来确保收到 BPDU。

要查看 BPDU 的内容，请对 PVST 和快速 PVST 将 **debug spanning-tree switch rx decode** 命令与 debug spanning-tree switch rx process 命令一起发出。发出 **debug spanning-tree mstp bpdu-rx** 命令可以查看 MST 的 BPDU 的内容：

```
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
```

对于 MST 模式，您可以使用此 **debug** 命令启用详细的 BPDU 解码：

```
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：** 对于 Cisco IOS 软件版本 12.1.13E 和更高版本，支持用于 STP 的条件调试。这意味着，您可以基于每个端口或每个 VLAN 调试所接收或传输的 BPDU。

发出 **debug condition vlan vlan_num** 或 debug condition interface interface 命令，可将调试输出的范围限制为每个接口或每个 VLAN。

## 防止网络发生转发循环

为应对 STP 无法正确处理某些故障，Cisco 开发了许多功能和增强功能来防止网络出现转发环路。

排除 STP 故障可帮助隔离特定故障并可能找到其原因，而实施这些增强功能是确保网络不会出现转发环路的唯一方法。

以下方法可防止网络出现转发环路：

1. 在所有交换机对交换机链路上启用单向链路检测 (UDLD)。有关 UDLD 的详细信息，请参阅了解和配置单向链路检测协议功能。
2. 在所有交换机上启用环路防护。有关环路防护的详细信息，请参阅使用环路防护和 BPDU 迟滞检测功能的生成树协议增强功能。启用 UDLD 和环路防护后，可消除引起转发环路的大多数可能原因。冲突链路（或所有与出现故障的硬件相关的链路）将被关闭或阻塞，而不会创建转发环路。**注意：** 尽管这两个功能似乎有些冗余，但两者都有其独特的功能。因此，同时使用这两个功能可提供最高级别的保护。有关 UDLD 和环路防护的详细比较，请参阅环路防护与单向链路检测。关于是必须使用主动 UDLD 还是正常 UDLD 有不同的观点。值得注意的是，与正常模式 UDLD 相比，主动 UDLD 不会针对环路提供更多防护。主动 UDLD 可检测端口卡住情形（如果链路接通，但没有任何关联的流量黑洞）。增加的这一功能的不利方面是，主动 UDLD 可能会在不存在一致故障时禁用链路。通常，用户会混淆 UDLD hello interval 修改与主动 UDLD 功能。这是不正确的。可以在这两种 UDLD 模式下修改计时器。**注意：** 在极少

数情况下，主动 UDLD 可以关闭所有上行链路端口，这会从根本上将交换机与网络的其余部分隔离。例如，当两台上游交换机的 CPU 利用率非常高，并且使用主动模式 UDLD 时，可能发生这种情况。因此，如果没有对交换机进行带外管理，建议您配置 errordisable 超时。

3. 在所有终端站端口上启用 portfast。您必须启用 portfast 来限制 TC 和随后的泛洪的数量，这可能会影响网络的性能。请将此命令仅用于连接到终端站的端口。否则，意外出现的拓扑环路可能会导致数据包环路并中断交换机和网络运行。**警告：** 在使用 **no spanning-tree portfast** 命令时，应特别谨慎。此命令只会删除任何端口特定的 portfast 命令。如果您在全局配置模式下定义 **spanning-tree portfast default** 命令，并且端口不是中继端口，此命令会隐式启用 portfast。如果不全局配置 portfast，**no spanning-tree portfast** 命令与 spanning-tree portfast disable 命令等效。
4. 在两侧（如果支持）和 non-silent 选项上将 EtherChannel 设置为 desirable 模式。使端口聚合协议(PAgP)保证在信道对等体之间的运行时一致性。这样可针对环路提供额外的防护，尤其是在信道重新配置期间（例如，链路加入或离开信道以及链路故障检测）更是如此。系统内置了一种在默认情况下已启用的信道配置错误防护功能，可防止由于信道配置错误或其他情况而引起的转发环路。有关此功能的详细信息，请参阅了解 EtherChannel 不一致检测。
5. 不要在交换机对交换机链路上禁用自动协商（如果支持）。自动协商机制可以传达远程故障信息，这是在远程端检测故障的最快方法。如果在远程端检测到故障，本地端将关闭链路，即使该链路仍在接收脉冲也是如此。与高级检测机制（如 UDLD）相比，自动协商的速度非常快（以微秒计），但缺少 UDLD 端到端覆盖（例如整个数据路径：CPU - 转发逻辑 - 端口 1 - 端口 2 - 转发逻辑 - CPU 与端口 1 - 端口 2）。在故障检测方面，主动 UDLD 模式可提供与自动协商类似的功能。当协商在链路的两端受支持时，不需要启用主动模式 UDLD。
6. 在调整 STP 计时器时应谨慎。STP 计时器彼此之间以及在网络拓扑上相互依赖。STP 可能无法正确处理对计时器所做的任意修改。有关 STP 计时器的详细信息，请参阅了解和调整生成树协议计时器。
7. 如果可能存在拒绝服务攻击，请使用根防护来保护网络 STP 外围安全。通过根防护和 BPDU 防护，可以防止 STP 受到外界影响。如果可能存在此类攻击，则必须使用根防护和 BPDU 防护来保护网络。有关根防护和 BPDU 防护的详细信息，请参阅以下文档：生成树协议根防护增强生成树 PortFast BPDU 防护增强功能
8. 在支持 portfast 的端口上启用 BPDU 防护，以防止 STP 受到连接到这些端口的未授权网络设备（例如集线器、交换机和桥接路由器）的影响。如果已正确配置根防护，则它已经可以防止 STP 受到外界的影响。如果启用了 BPDU 防护，它将关闭正在接收任何 BPDU（不仅是高级 BPDU）的端口。如果需要调查此类事件，则这可能很有用，因为 BPDU 防护会生成 syslog 消息并关闭端口。值得注意的是，如果两个启用了 portfast 的端口直接或通过集线器相连，则根防护或 BPDU 防护不会防止短期环路。
9. 在管理 VLAN 上避免用户流量。管理 VLAN 包含在构造块（而非整个网络）中。交换机管理接口在管理 VLAN 上接收广播数据包。如果出现过多的广播（例如广播风暴或应用程序发生故障），则交换机 CPU 可能会超载，这可能使 STP 的运行失真。
10. 可预测（硬编码）的 STP 根和备份 STP 根放置。必须配置 STP 根和备份 STP 根，以便在发生故障时以可预测的方式进行收敛，并在每种情形下构建最佳拓扑。请不要将 STP 优先级置于默认值，以防止执行无法预测的根交换机选择。

## 相关信息

- LAN 产品支持
- LAN 交换技术支持
- 技术支持和文档 - Cisco Systems