

# 查看可恢复以太网协议

## 目录

[简介](#)

[支持的平台](#)

[背景信息](#)

[为什么选择 REP ?](#)

[好处](#)

[限制](#)

[协议运行](#)

[分段](#)

[链路状态层](#)

[责任](#)

[端口状态](#)

[数据包详细信息](#)

[硬件泛洪层 \(HFL\)](#)

[BPA](#)

[考虑事项](#)

[BPA 行为](#)

[硬件协助](#)

[EPA](#)

[分段统计信息](#)

[检测分段完整情况](#)

[启动 VLAN 负载均衡](#)

[PDU格式](#)

[故障排除](#)

[中断的链路调查](#)

[备用 \(ALT\) 端口](#)

[排除 REP 邻接故障](#)

[调试](#)

[有用的调试](#)

[不太有用的调试](#)

[相关信息](#)

## 简介

本文档概述弹性以太网协议(REP)。

## 支持的平台

- 桌面交换业务部门 (DSBU) 城域交换机 ( 3750ME 和 ME3400 ) 版本 12.2(40)SE 及更高版本
- Cisco Catalyst 4500 系列交换机 12.2(44)SG 及更高版本
- Whitney2 (12.2SXI) 以后的 Cisco Catalyst 6500 系列交换机

- Cobra (12.2SRC) 以后的 Cisco Catalyst 7600 系列路由器

## 背景信息

### 为什么选择 REP ?

REP 是用于在某些特定第 2 层网络设计中替换生成树协议 (STP) 的协议。最新的 STP 规范是多生成树 (MST) 协议，该协议已在 802.1Q-2005 中定义。希望使用 MST 替代方案的用户应适当关注以下几点：

- STP 将桥接域视为一个整体。因此，如果更改任意一条远程链路的状态，则系统会从本地故障状态中恢复正常。如果将桥接域分段为较小的独立部分，则只能缓解 STP 的明显的不可预测性。遗憾的是，如果不从生成树中删除一些关键功能（如在所有情况下都防止环路），实现这一点就很复杂，甚至是不可能的。
- STP 收敛对于期望恢复时间为 50 毫秒 (ms) 的服务提供商来说似乎很慢，这在电路交换技术中很常见。这种缓慢并非由协议本身引起的；平台需要优化才能以更有效的方式运行 STP。同时，还需要采用可解决平台限制问题的新解决方案。
- MST 负载均衡配置不具备灵活性。为了让 MST 实现实例负载均衡，所有网桥都必须属于同一个区域。区域由用户配置定义，如果不引入某些网络重新收敛，就无法修改交换机上的 MST 配置。可以通过精细预配置来解决此问题，或者通过使用其他协议（例如，VLAN 中继协议 [VTP] v3）将其限制在有限范围内。

### 好处

下面是 REP 带来的一些好处：

- REP 提供以下收敛时间：
  - 3750ME 的收敛时间介于 20ms 至 79ms 之间
  - ME3400 的收敛时间介于 40ms 和 70ms 之间
- 可在当前硬件上运行
- 可预测，且端口已阻止
- 配置简单

### 限制

下面是 REP 的一些局限性：

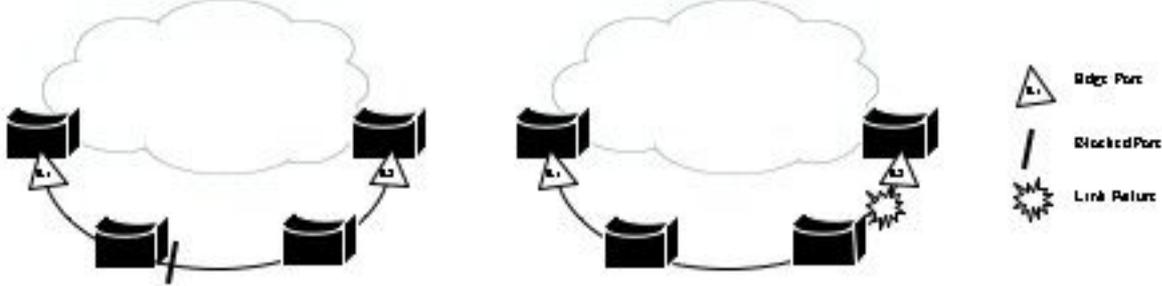
- 不支持即插即用
- 无法防止错误配置（容易形成环路）
- 冗余数量有限（仅能承受一个链路故障）
- 无法发现全局拓扑（仅支持分段拓扑）
- Cisco 专有

## 协议运行

## 分段

REP 使用分段作为最小网络构件。分段就是链接在一起的一些端口。网桥上的一个给定分段只能有两个端口，并且每个分段端口最多可以有一个外部邻居。分段的定义完全由用户配置来实现。分段由用户确定的两个**边缘端口**来终止。在分段上运行的 REP 协议将尽可能实现最小化运行，且仅保证以下属性：

- 如果分段中的所有端口均在线且可供使用，则其中一个会在逻辑上阻止每个 VLAN 的流量。
- 如果分段中的至少一个端口因任何原因而不可使用，则所有其他可使用端口会转发所有 VLAN。
- 在链路故障情况下，应尽快取消阻止所有剩余的可使用端口。同样地，当最后一个故障端口再次恢复正常运行时，当选择每个 VLAN 一个逻辑阻塞的端口时，必须尽可能减少网络中断。



简单构件

图 1.分段作为

图 1 显示的示例分段包括分布在四个网桥间的六个端口。已配置的边缘端口 E1 和 E2 在图中用三角形来表示，而在逻辑上阻止的端口用线条来表示。当所有端口均可供使用时，如左图所示，单个端口被阻止。当网络中出现故障时，如右图所示，在逻辑上阻止的端口回到转发状态。

如图 1 所示，当分段开放时，它绝不会在其两个边缘端口之间提供连接。REP 边缘交换机之间的连接假设存在于分段的外部（通过 STP）。使用可选配置时，如果 REP 分段中出现故障，则会生成 STP 拓扑变更通告 (TCN)，以便加快收敛。

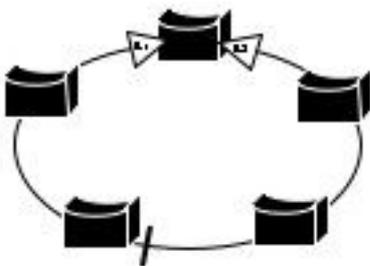


图 2.分段可以围成一个环

如图 2 所示，当两个边缘端口位于同一个交换机上时，分段围成一个环。在这种情况下，分段的边缘端口之间可以连接。事实上，此配置允许您在分段中的任意两个交换机之间创建一个冗余连接。

如图 1 和图 2 所示，如果使用开放和闭合分段的组合，即可实现各种不同的网络设计。

## 链路状态层

### 责任

- 与唯一邻居建立连接。
- 定期检查与邻居连接的完整性。
- 发送和接收更高层状态机的消息。
- 确认从邻居接收的数据。

- 限制协议数据单元 (PDU) 的速率。

## 端口状态

为 REP 进行端口配置时，它会出现以下状态：

- 故障状态 ( 阻止 )
- 形成的邻接关系：
- 备用端口 ( 阻止，但可使用 )
- 丢失无线接入点 (AP) 选择：
- 开放端口 ( 如果另一个端口选择了该“AP” )

在这些条件下端口状态未变为可供使用：

- 在端口上未检测到邻居
- 在端口上检测到多个邻居
- 邻居未确认 (ACK) 消息

## 数据包详细信息

默认情况下，REP 向本地 VLAN ( 未标记 ) 上的网桥协议数据单元 (BPDU) 类 MAC 地址发送 Hello 数据包，以便不运行该功能的设备将这些数据包丢弃。每个链路状态层 (LSL) PDU 包括发送的 PDU 的序列号和收到的最后一个 PDU 的远程序列号。这样可确保端口之间实现可靠的传输。每个邻居会保留发送的每个 PDU 的副本，直至收到 ACK。如果未收到任何 ACK，则系统会在计时器过期后重新发送。

实际 LSL PDU 包含：

- ProtocolVersion ( 当前为 0 )
- SegmentID
- RemotePortID
- LocalPortID
- LocalSeqNumber
- RemoteSeqNumber
- 更高层 TLV

系统会在每个询问间隔或当更高层协议需要 LSL 数据包时发送该数据包。当构建 LSL PDU 时，先填充其自身字段，例如 SegmentID 和 LocalPortID。接下来，它会在更高层协议队列 ( 例如阻止端口通告 [BPA] 或结束端口通告 [EPA] ) 中查找，以便查看是否有任何其他数据需要排队。

## 硬件泛洪层 (HFL)

HFL 是在链路发生故障后帮助快速收敛的 REP 模块。它不会像 LSL 一样将 PDU 发送到 BPDU MAC 地址，而是将组播 PDU 发送到 REP 管理 VLAN 上的特定 MAC 地址 (0100.0ccc.ccce)。这样，它会在硬件中泛洪到分段中的所有交换机。

HFL 数据包格式很简单：

- 协议版本 ( 仍然是 0 )
- SegmentID
- 更高层类型长度值 (TLV)

此时，通过 HFL 发送的唯一 TLV 是 BPA。

## BPA

BPA 由 AP 发送，以便通告它们依照其端口优先级阻止的 VLAN。这有助于通知网段链路故障，并确保每个 VLAN 的每个网段只有一个 AP。这不容易实现。

### 考虑事项

在稳定拓扑中，AP 选择很简单。端口上线时会先作为所有 VLAN 的 AP (阻止)。当端口从另一个具有更高优先级的端口接收 BPA 时，它知道自己可安全地取消阻止。当分段上的端口发生故障时，可使用同一过程来取消阻止其他端口。所有发生故障的端口都会生成比当前 AP 更高的端口优先级 (优先级中包含 failed 位)，这将导致当前 AP 取消阻塞。

但是，当此链路恢复正常运行时，会出现一些问题。当发生这种情况时，优先级的失效位会清除，且优先级恢复正常。即使此端口知道其新优先级，网段的其他部分也可能具有来自此端口的过时 BPA 信息。此图说明了此场景：

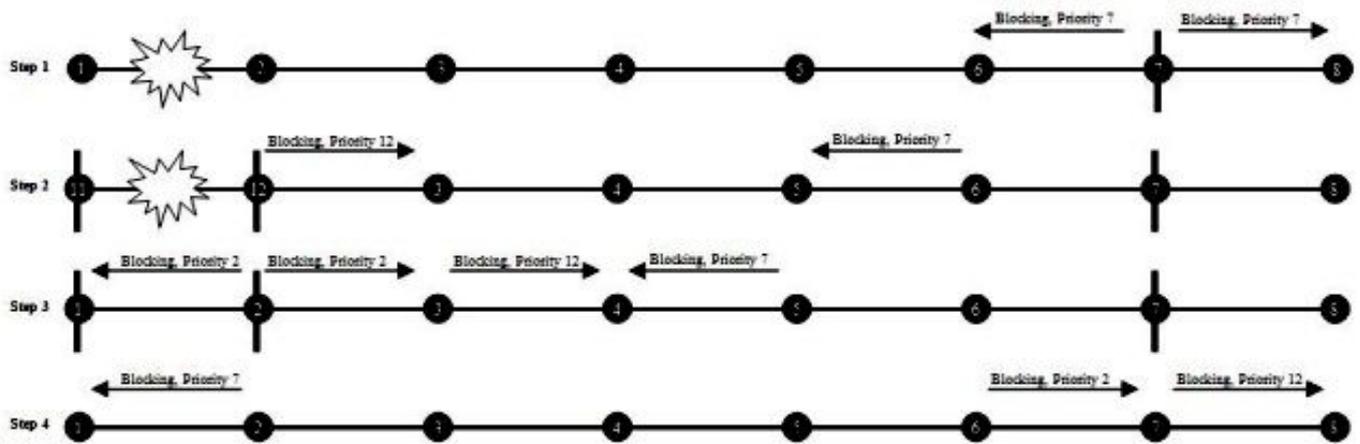


图 3. 打开数据段的陈旧信息

在本场景开始时，端口 7 会阻塞并通告其优先级为 7。接下来，11 和 12 之间的链路中断，导致 12 发送一个 BPA，指明它使用优先级 12 进行阻止。在这些阻塞端口接收另一个端口的 BPA 之前，端口 12 会恢复正常并可操作。不久之后，端口 12 会收到端口 7 的 BPA (优先级为 7)，因此它会取消阻止。然后，端口 7 会从端口 12 (优先级为 12) 获得陈旧的 BPA，因此它会取消阻止。这样就形成一个环路。由于出现这种争用条件，所以 BPA 会使用密钥。

### BPA 行为

每个端口使用以下信息计算端口优先级：

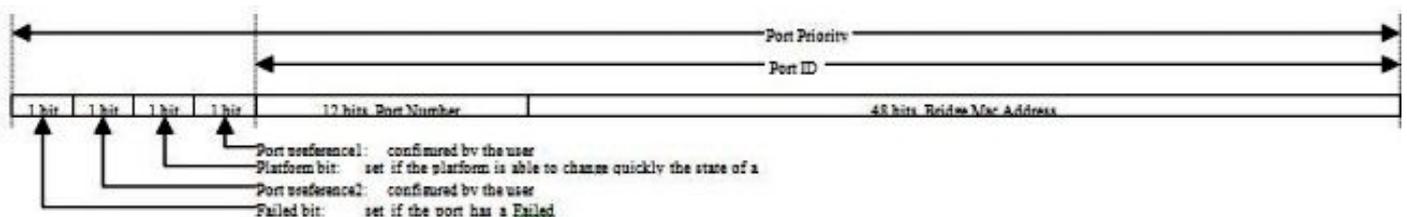


图 4. 端口优先级

为什么故障端口始终会选为分段上的 AP，现在已经非常清楚。当端口从故障变为备用时，它会基于其端口 ID 和一个随机数生成唯一的密钥，并与其端口 ID 一起通告。仅当 AP 从包含其本地密钥

的阻止端口收到消息时才会取消阻止。此机制可帮助防止出现前面部分所述的争用条件场景。下图显示了当端口接通和断开时会发生的情况：

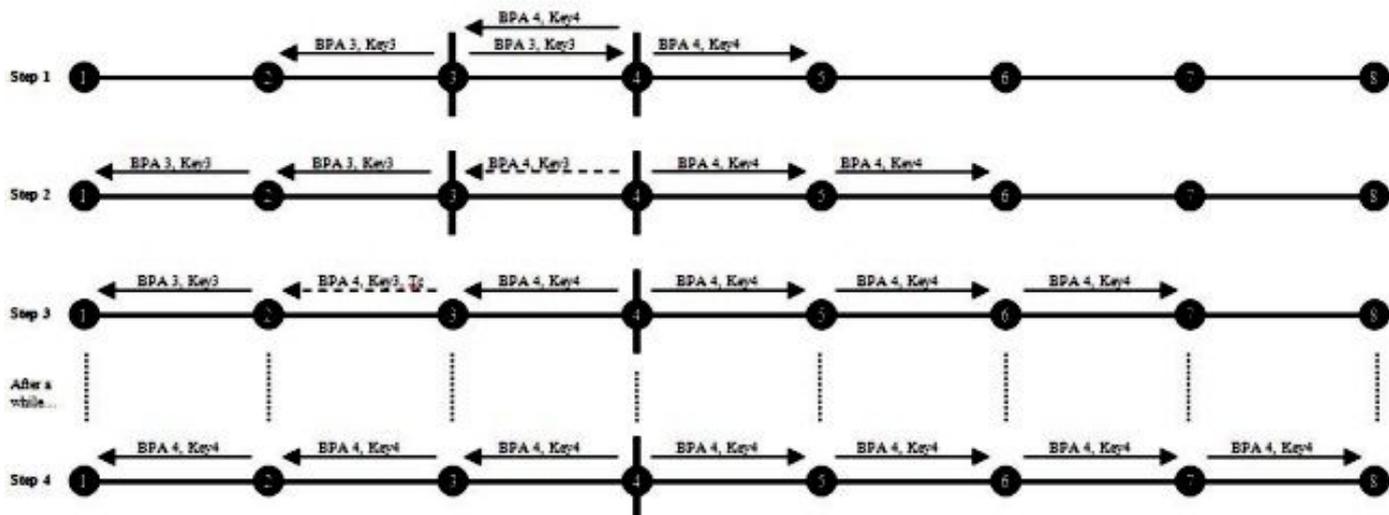


图 5.链路正常运行时的 BPA 运行

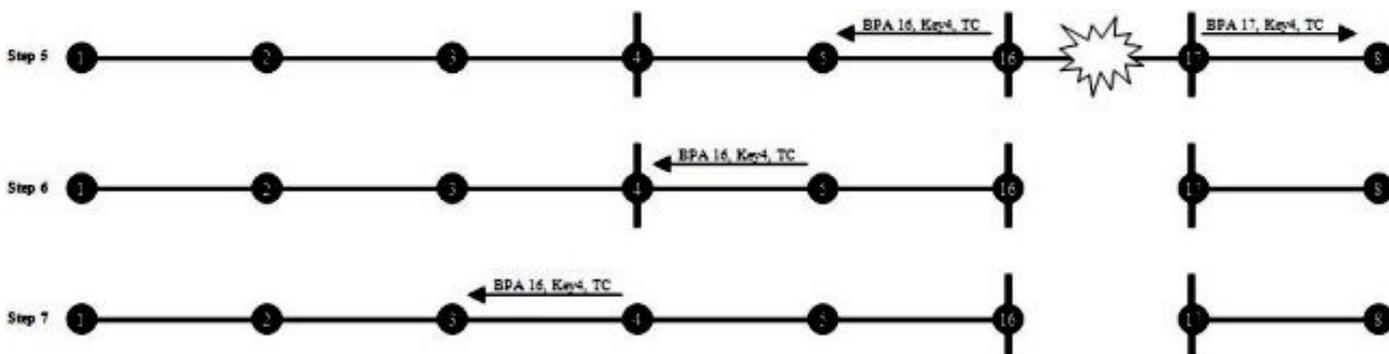


图 6.发生链路故障之后的 BPA 运行

### 硬件协助

当分段上发生链路故障时，BPA 会通过 HFL 泛洪到分段的其余部分。为了让泛洪完全有效，无论是在所有分段端口上，还是在分段外部的边缘端口之间，都必须承载管理 VLAN。由于 HFL 不能保证可靠的传输，BPA 也通过 LSL 发送此信息。如果 HFL 传送出现任何问题，LSL 可确保进行重新收敛。

### EPA

结束端口是边缘端口或故障端口。当分段在两边通过边缘端口来终止时，它会被视为是完整的，且可实现 VLAN 负载均衡。当分段由故障端口来终止时，由于所有端口都是开放的，因此无法实现负载均衡。

结束端口定期发送 EPA，经由 LSL 进行中继。这些消息：

- 传播有关分段的统计信息
- 检测分段完整情况
- 启动 VLAN 负载均衡

### 分段统计信息

每个结束端口通过 LSL 发送周期性 EPA，其中包含关于其自身的消息。每个中间端口添加自己的

信息并中继EPA。因为这些消息会朝两个方向移动，每个 REP 参与交换机均已了解整个 REP 分段。EPA 中包含的信息包括：

- 网桥 ID
- 两个 REP 参与端口的端口 ID 和状态

## 检测分段完整情况

每个边缘端口会发送具有其自己的边缘优先级和特殊密钥（与 BPA 密钥无关）的特殊选择 EPA 消息。接收此消息的第一个端口会在此消息中放入其自己的端口优先级并将其中继到下一个交换机。路径上的每台交换机将自己的端口优先级与EPA中的端口优先级进行比较，如果优先级更高，则用自己的端口优先级替换。当边缘端口收到 EPA 时，会将边缘优先级与其自己的优先级进行比较。如果收到的EPA具有更高的优先级，边缘端口会将其带有密钥的下一个EPA消息发送到主边缘。此机制可帮助实现以下两个目标：

- 确保分段完整
- 为两个边缘端口提供具有最高优先级的中间端口的信息

## 启动 VLAN 负载均衡

VLAN负载均衡通过两个不同AP实现，这些AP会阻塞不同的VLAN。主边缘负责至少一个VLAN子集上的AP，并发送一条EPA消息，告知最高优先级端口阻止其余端口。系统已利用 EPA 选择消息来获取有关具有最高优先级的中间端口的信息。为此生成的消息类型是包含最高优先级端口需要阻止的 VLAN 位图的 EPA 命令 TLV。

## PDU格式

EPA 报头：

- 类型=EPA
- 实例号
- 可选 TLV

选择 TLV：

- edgePriority
- edgeKey
- BestPortPriority

命令 TLV：

- SelectedPortPriority
- SelectedVLANs

信息 TLV：

- 网桥 ID
- 两个端口 ID
- 端口角色

## 故障排除

## 中断的链路调查

以下是正常拓扑的示例：

```
SwitchA#show rep topology
REP Segment 1
BridgeName PortName edge Role
-----
SwitchA Fa0/2 Pri Alt
SwitchC Fa1/0/23 Open
SwitchC Fa1/0/2 Open
SwitchD Fa0/23 Open
SwitchD Fa0/2 Open
SwitchB Fa1/0/23 Sec Open
```

以下是出现中断的示例：

```
SwitchA#show rep topology
REP Segment 1
Warning: REP detects a segment failure, topology may be incomplete
BridgeName PortName edge Role
-----
SwitchA Fa0/2 Sec Open
SwitchC Fa1/0/23 Open
SwitchC Fa1/0/2 Fail
```

以前的链路如下所示：

```
SwitchA#show rep topology archive
REP Segment 1
BridgeName PortName edge Role
-----
SwitchA Fa0/2 Pri Open
SwitchC Fa1/0/23 Open
SwitchC Fa1/0/2 Open
SwitchD Fa0/23 Open
SwitchD Fa0/2 Open
SwitchB Fa1/0/23 Sec Alt
```

输入此命令，以便获得 SwitchC 和 SwitchD 之间链路故障的更多详情：

```
SwitchA#show rep topology archive detail
REP Segment 1
<snip>
SwitchC, Fa1/0/2 (Intermediate)
Open Port, all vlans forwarding
Bridge MAC: 0017.5959.c680
Port Number: 004
Port Priority: 010
Neighbor Number: 3 / [-4]
SwitchD, Fa0/23 (Intermediate)
Open Port, all vlans forwarding
Bridge MAC: 0019.e73c.6f00
Port Number: 019
Port Priority: 000
Neighbor Number: 4 / [-3]
<snip>
```

恢复的链路如下所示：

```
SwitchA#show rep topology
REP Segment 1
BridgeName PortName edge Role
-----
SwitchA Fa0/2 Pri Open
SwitchC Fa1/0/23 Open
SwitchC Fa1/0/2 Alt
SwitchD Fa0/23 Open
SwitchD Fa0/2 Open
SwitchB Fa1/0/23 Sec Open
```

请注意，以前发生故障的端口仍保留为AP并继续阻塞。这是因为 AP 选择仅在阻止端口之间发生。当此链路发生故障时，拓扑中的所有其他端口均开放。当链路接通时，SwitchC 和 SwitchD 均发送带有其优先级的 BPA。SwitchC Fa1/0/2 拥有更高的优先级，因此它成为 AP。这种情况一直保持到拓扑中的另一个端口发生故障，或者一直保持到执行 **preempt**。

## 备用 (ALT) 端口

ALT 端口阻止部分或全部 VLAN。如果 REP 网段发生故障，则没有 ALT 端口；所有端口都是打开的。这就是在发生故障时 REP 能够为数据流量提供活动路径的方式。

在完整 REP 分段（没有故障）中，存在一个或两个 ALT 端口。如果已启用 VLAN 负载均衡，则分段中有两个 ALT 端口（其中一个 ALT 端口阻止一组指定的 VLAN，而另一个 ALT 端口始终在主边缘端口，阻止一组补充 VLAN）。如果未启用 VLAN 负载均衡，则分段中只有一个 ALT 端口，阻止所有 VLAN。

端口联机的顺序，以及内置端口优先级确定网段中的哪个端口成为 ALT 端口。如果希望特定端口成为 ALT 端口，请使用 **preferred** 关键字进行配置。示例如下：

```
interface gig3/10
rep segment 3 edge preferred
```

假设 **gig3/1** 是主边缘端口，且您要配置 VLAN 负载均衡：

```
interface gig3/1
rep segment 3 edge primary
rep block port preferred vlan 1-150
```

使用此配置，在完成抢占之后，端口 **gig3/10** 是阻止 VLAN 1 至 150 的 ALT 端口，且端口 **gig3/1** 是阻止 VLAN 151 至 4094 的 ALT 端口。

抢占可使用 **rep preempt segment 3** 命令手动完成，如果在主边缘端口下配置了 **rep preempt delay <seconds>**，也可自动完成。

当分段在链路故障后恢复正常时，与故障端口邻接的两个端口之一会作为 ALT 端口开始运行。然后，在完成抢占之后，ALT 端口的位置会变为由配置来指定。

## 排除 REP 邻接故障

输入此命令以查看是否存在邻接关系：

```
SwitchC#show interface fa1/0/23 rep
Interface Seg-id Type LinkOp Role
-----
```

FastEthernet1/0/23 1 TWO\_WAY Open

输入此命令以获取详细信息：

```
SwitchC#show interface fa1/0/23 rep detail
FastEthernet1/0/23 REP enabled
Segment-id: 1 (Segment)
PortID: 001900175959C680
Preferred flag: No
Operational Link Status: TWO_WAY
Current Key: 000400175959C6808335
Port Role: Open
Blocked VLAN: <empty>
Admin-vlan: 1
Preempt Delay Timer: disabled
Configured Load-balancing Block Port: none
Configured Load-balancing Block VLAN: none
STCN Propagate to: none
LSL PDU rx: 255547, tx: 184557
HFL PDU rx: 3, tx: 2
BPA TLV rx: 176096, tx: 2649
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 870, tx: 109
EPA-COMMAND TLV rx: 2, tx: 2
EPA-INFO TLV rx: 45732, tx: 45733
```

## 调试

大多数调试会打印过多输出信息，作用并不是很大。以下是完整列表（某些仅用于内部服务）：

```
SwitchB#debug rep ?
all all debug options
bpa-event bpa events
bpasm BPA state machine
database protocol database
epasm EPA state machine
error protocol errors
failure-recovery switchover events
lslsm LSL state machine
misc miscellaneous
packet protocol PDU
prsm Port Role state machine
showcli show debug info
```

## 有用的调试

以下是一些有用的调试：

- **debug rep showcli**(needs service internal) — 当您输入常规show rep命令时，此debug会打印许多额外信息。
- **debug rep error** — 此调试可能非常有用。
- **debug rep failure-recovery** — 此调试输出在链路中断时经过的消息。

```
*Mar 5 05:01:11.530: REP LSL-OP Rx EXT Local (Fa0/23 seg:1, tc:1, frs:0) prio:
*Mar 5 05:01:11.530: 0x80 0x00 0x19 0x00 0x17 0x59 0x59 0xC6
*Mar 5 05:01:11.530: 0x80
*Mar 5 05:01:11.530: REP Flush from Fa0/23 to REP, sending msg
```

```
*Mar 5 05:01:11.530: REP LSL-OP Rx INT Local (Fa0/2 seg:1, tc:1, frs:0) prio:
*Mar 5 05:01:11.530: 0x80 0x00 0x19 0x00 0x17 0x59 0x59 0xC6
*Mar 5 05:01:11.530: 0x80
*Mar 5 05:01:11.530: REP Flush from Fa0/2 to REP, sending msg
```

- **debug rep prsm** — 此调试可用于对不形成的邻接关系进行故障排除。它会为您提供在链路接通/中断时所发生情况的详细信息。

```
4d05h: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
4d05h: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2,
changed state to up
```

```
*Mar 5 05:06:19.098: rep_pr Fa0/2 - pr: during state FAILED_PORT,
got event 5(no_ext_neighbor)
*Mar 5 05:06:19.098: @@@ rep_pr Fa0/2 - pr: FAILED_PORT ->
FAILED_PORT_NO_EXT_NEIGHBOR[Fa0/2]rep_pr_act_no_ext_neighbor@272:
PRSM->fp_no_ext_neighbor state
[Fa0/2]rep_pr_lsl_event_handler@448: REP_MSG_EXT_PEER_GONE rcvd
```

```
4d05h: %REP-4-LINKSTATUS: FastEthernet0/2 (segment 1) is operational
*Mar 5 05:06:22.236: rep_pr Fa0/2 - pr: during state FAILED_PORT_NO_EXT_
NEIGHBOR, got event 0(link_op)
*Mar 5 05:06:22.236: @@@ rep_pr Fa0/2 - pr:
FAILED_PORT_NO_EXT_NEIGHBOR ->
ALTERNATE_PORT[Fa0/2]rep_pr_act_ap@162: PRSM->alternate state
[Fa0/2]rep_pr_lsl_event_handler@431: REP_MSG_LINKOP_TRUE rcvd
```

```
*Mar 5 05:06:23.125: rep_pr Fa0/2 - pr: during state ALTERNATE_PORT,
got event 2(pre_empt_ind)
*Mar 5 05:06:23.133: @@@ rep_pr Fa0/2 - pr: ALTERNATE_PORT -> UNBLOCK_VLANS_ACT
*Mar 5 05:06:23.133: rep_pr Fa0/2 - pr: during state UNBLOCK_VLANS_ACT,
got event 3(no_local_block_vlan)
*Mar 5 05:06:23.133: @@@ rep_pr Fa0/2 - pr: UNBLOCK_VLANS_ACT ->
OPEN_PORT[Fa0/2]rep_pr_act_op@252: PRSM->active state
[Fa0/2]rep_pr_act_uva@222: PRSM unblock vlans
[Fa0/2]rep_pr_sm_preempt_ind@374: Posting pre empt indication
```

- **debug rep epasm** — 此调试在拓扑更改期间提供有用的信息。如果分段保持稳定，则不会打印任何内容。

以下是端口离线时的输出：

```
*Mar 5 04:48:31.463: rep_epa_non_edge Fa0/2 - epa-non-edge: during state
INTERMEDIATE_PORT, got event 1(lr_eq_fp)*Mar 5 04:48:31.463: @@@ rep_epa_non_
edge Fa0/2 - epa-non-edge: INTERMEDIATE_PORT -> FAILED_PORT[Fa0/2]rep_epa_non_
edge_act_failed_port@164: Trigger archiving
[Fa0/23]rep_epa_set_peer_archive_flag@1084: set arch flag
[Fa0/2]rep_epa_non_edge_act_failed_port@171: no edge, failed port
*Mar 5 04:48:35.473: rep_epa_non_edge Fa0/2 - epa-non-edge: during state
FAILED_PORT, got event 0(epa_hello_tmo)
*Mar 5 04:48:35.473: @@@ rep_epa_non_edge Fa0/2 - epa-non-edge: FAILED_PORT ->
FAILED_PORT[Fa0/2]rep_epa_non_edge_act_periodic_tx@90: archiving on port down
[Fa0/2]rep_epa_copy_topology@913: deip=0x3396F18,pe=0,se=1,fp=0,ap=0,op=2
[Fa0/23]rep_epa_non_edge_handle_info_tlv@1560: archiving on internal flag
[Fa0/23]rep_epa_copy_topology@913: deip=0x33961F0,pe=1,se=0,fp=0,ap=1,op=3
[Fa0/2]rep_epa_non_edge_act_periodic_tx@102: epa non edge, send info tlv
[Fa0/23]rep_epa_set_peer_archive_flag@1084: set arch flag
[Fa0/2]rep_epa_non_edge_handle_election_tlv@325: archiving on seg cfg change
[Fa0/2]rep_epa_copy_topology@913: deip=0x3396F18,pe=0,se=1,fp=0,ap=0,op=2
[Fa0/2]rep_epa_set_peer_archive_flag@1084: set arch flag
[Fa0/23]rep_epa_non_edge_handle_election_tlv@325: archiving on seg cfg change
[Fa0/23]rep_epa_copy_topology@913: deip=0x33961F0,pe=1,se=0,fp=0,ap=1,op=3
```

```
[Fa0/2]rep_epa_non_edge_handle_info_tlv@1560: archiving on internal flag
[Fa0/2]rep_epa_copy_topology@913: deip=0x3396F18,pe=0,se=1,fp=0,ap=0,op=2
```

以下是端口联机时的输出：

```
*Mar 5 04:49:39.982:      rep_epa_non_edge Fa0/2 - epa-non-edge: during state FAILED_PORT,
got event 2(lr_neq_fp)
*Mar 5 04:49:39.982: @@@ rep_epa_non_edge Fa0/2 - epa-non-edge: FAILED_PORT ->
INTERMEDIATE_PORT[Fa0/2]rep_epa_non_edge_stop_timer@123: epa non edge, stop the timer
[Fa0/2]rep_epa_copy_topology@913: deip=0x32E2FA4,pe=0,se=1,fp=0,ap=1,op=1
[Fa0/2]rep_epa_copy_to_stable_topology@1040: copy to stbl
[Fa0/23]rep_epa_copy_topology@913: deip=0x3ACFFB8,pe=1,se=0,fp=0,ap=0,op=4
[Fa0/23]rep_epa_copy_to_stable_topology@1040: copy to stbl
```

## 不太有用的调试

- **debug rep bpa-event** — 此调试告知您何时接收BPA，以及您对其执行的操作。它每秒显示四行。

```
[Fa0/23]: BPA: Sending ext pak to bparx
[Fa0/2]: BPA: Enqueued internal pak
[Fa0/2]: BPA: Sending int msg to bparx
[Fa0/2]: BPA: Relay pak
[Fa0/2]: BPA: Enqueue ext pak
```

- **debug rep bpsm** — 此调试告诉您每收到BPA时的BPA状态机执行的操作。它每秒显示三行。

```
*Mar 5 04:44:23.857:      rep_bpa_rx BPA RX sm: during state BPA_RX_IDLE,
got event 0(bpa_rx_bpa_msg)
*Mar 5 04:44:23.857: @@@ rep_bpa_rx BPA RX sm: BPA_RX_IDLE -> BPA_RX_IDLE
[Fa0/23]: BPA Rx sm: Received bpa: <internal> 0, <vlan_detail> 0
[Fa0/23]: BPA Rx sm: Role 2: TC 0; Internal 0; Frm Remote Segment 0
```

```
*Mar 5 04:44:23.857:      rep_bpa_rx BPA RX sm: during state BPA_RX_IDLE,
got event 0 (bpa_rx_bpa_msg)
*Mar 5 04:44:23.857: @@@ rep_bpa_rx BPA RX sm: BPA_RX_IDLE -> BPA_RX_IDLE
[Fa0/2]: BPA Rx sm: Received bpa: <internal> 1, <vlan_detail> 0
[Fa0/2]: BPA Rx sm: Role 2: TC 0; Internal 1; Frm Remote Segment 0
```

- **debug rep lsism** — 此调试会转储低级LSL消息处理。

```
*Mar 5 05:03:10.564: REP Fa0/23 seq:4411 ACK'ed (ref: 1)
*Mar 5 05:03:10.564: REP Fa0/23 seq:4412 ACK'ed (ref: 1)
*Mar 5 05:03:10.564: REP LSL: Fa0/23 rx expected seq# (4744),
process it (TLV: 0).
*Mar 5 05:03:10.782: REP Fa0/2 seq:440 ACK'ed (ref: 1)
```

## 相关信息

- [思科技术支持和下载](#)

## 关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。