ACI SPAN指南

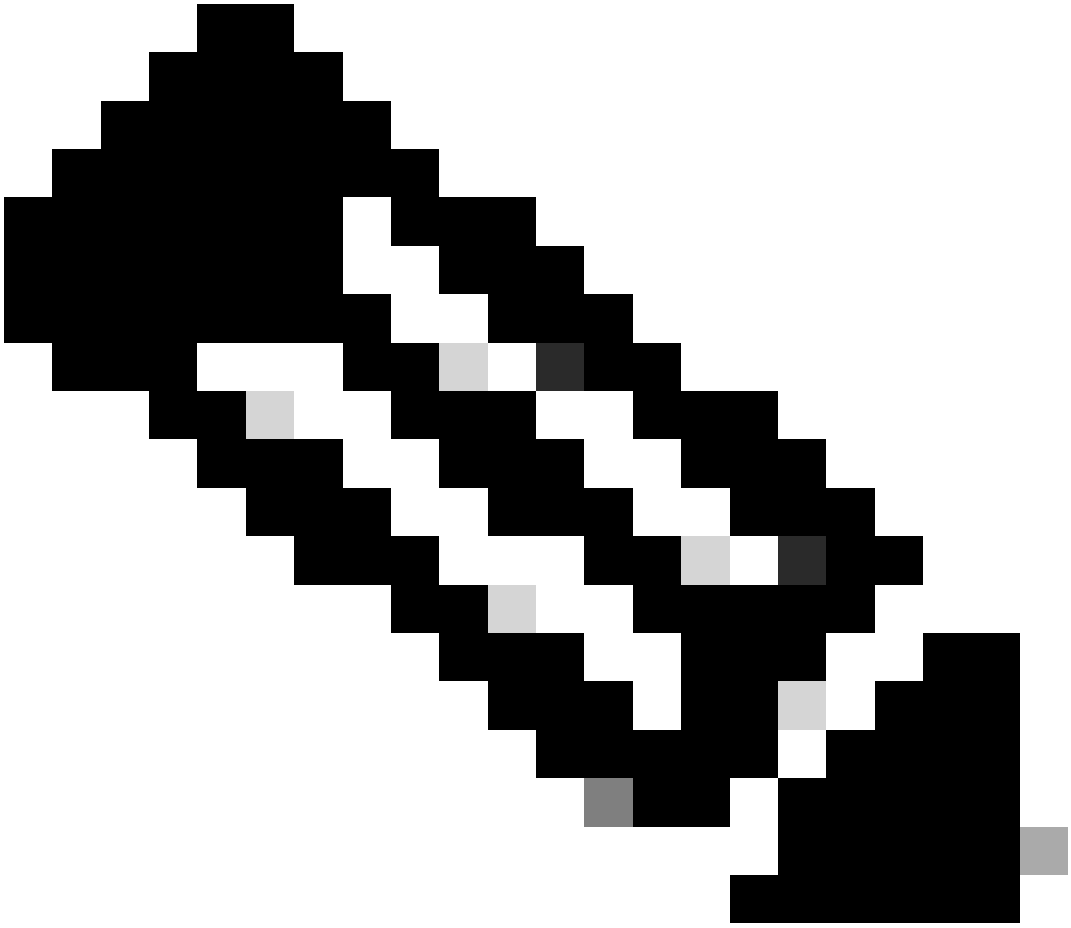
目录

简介

本文档介绍如何在思科以应用为中心的基础设施(ACI)上配置交换端口分析器(SPAN)。

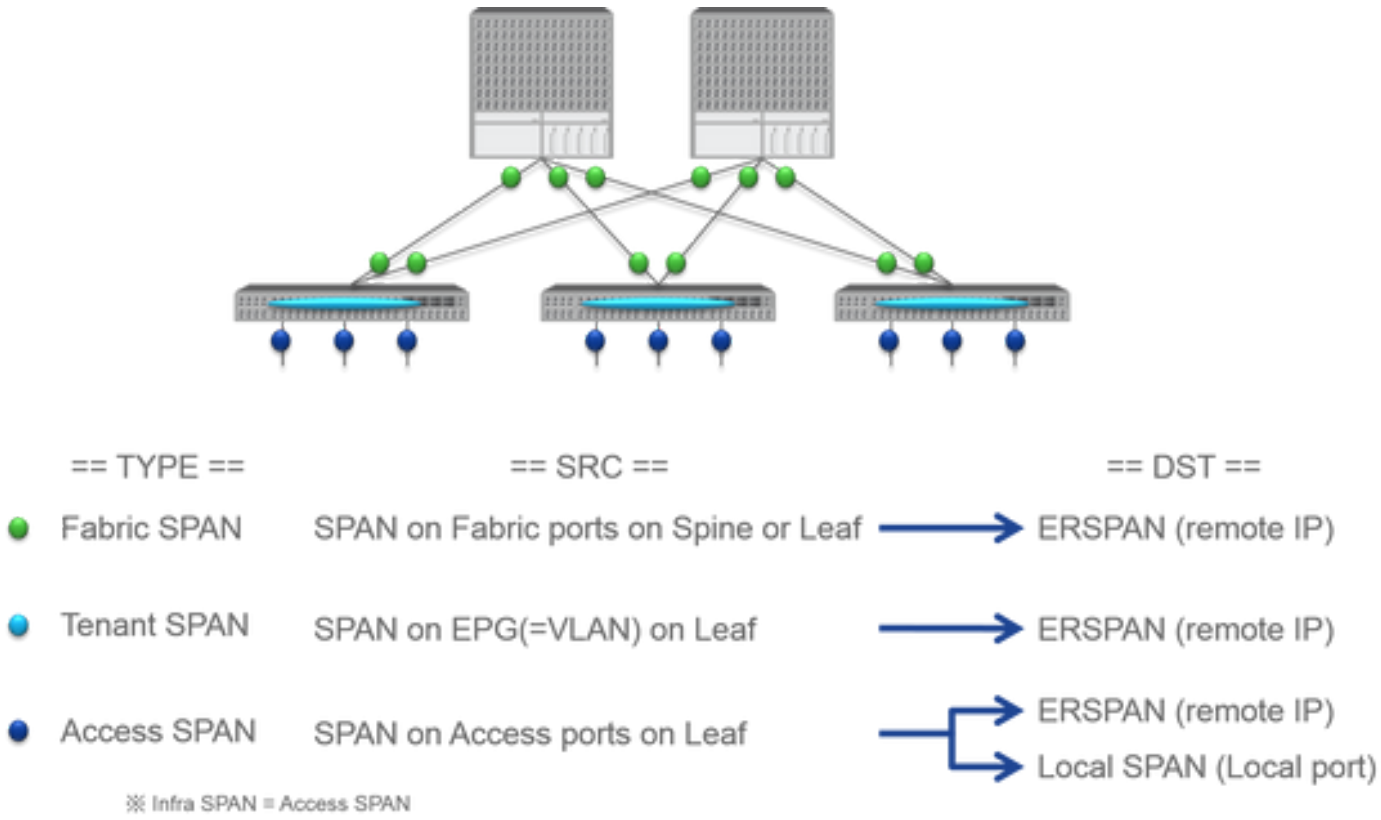
背景信息

一般来说，SPAN有三种类型。本地SPAN、远程SPAN (RSPAN)和封装远程SPAN (ERSPAN)。这些SPAN之间的差异主要在于复制数据包的目的地。Cisco ACI支持本地SPAN和ERSPAN。



注意：本文档假设读者已经对SPAN大致熟悉，例如本地SPAN和ERSPAN差异。

思科ACI中的SPAN类型



思科ACI有三种类型的SPAN；Fabric SPAN、Tenant SPAN和Access SPAN。每个SPAN之间的差异是复制数据包的来源。

如前所述

- **Fabric SPAN** 是捕获从 **interfaces between Leaf and Spine switches**传入和传出的数据包。
    - Access SPAN 捕获进出的interfaces between Leaf switches and external devices数据包。
    - Tenant SPAN 捕获进出的EndPoint Group (EPG) on ACI Leaf switches数据包。

此SPAN名称对应于要在思科ACI GUI上配置的位置。

- 交换矩阵SPAN配置在 Fabric > Fabric Policies
    - 访问SPAN配置在 Fabric > Access Policies

- 租户SPAN配置在 Tenants > {each tenant}

至于每个SPAN的目标，只有Access SPANLocal SPAN能和ERSPAN。另外两个SPAN(Fabric和Tenant)只能使用ERSPAN。

限制和指导原则

请查看思科APIC故障排除指南的限制和指导原则。在Troubleshooting Tools and Methodology > Using SPAN中提到它。

配置

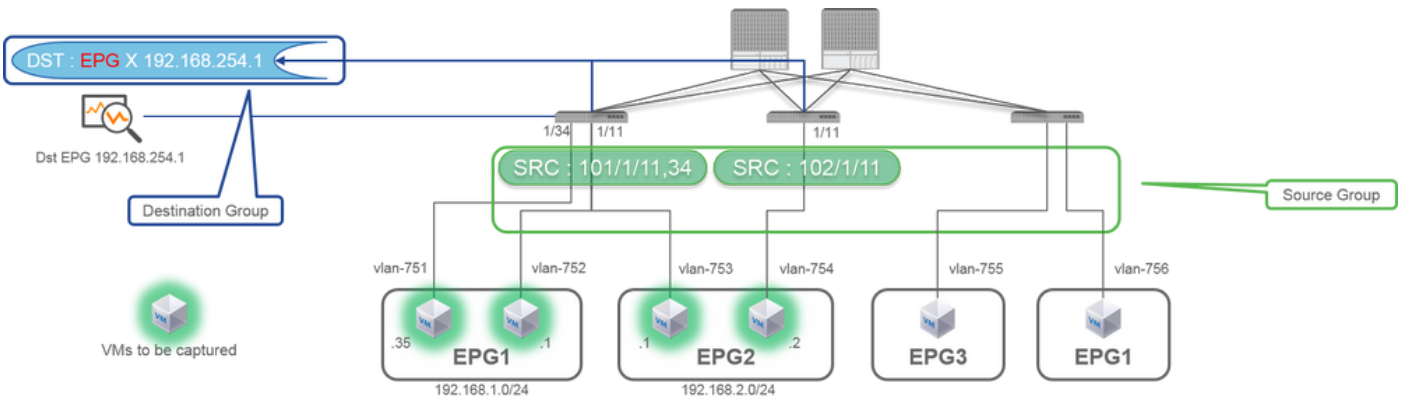
本节介绍与每个SPAN类型的配置相关的简要示例。在后面的章节中，有关于如何选择范围类型的特定示例案例。

思科APIC故障排除指南：故障排除工具和方法>使用SPAN中还对SPAN配置进行了说明。

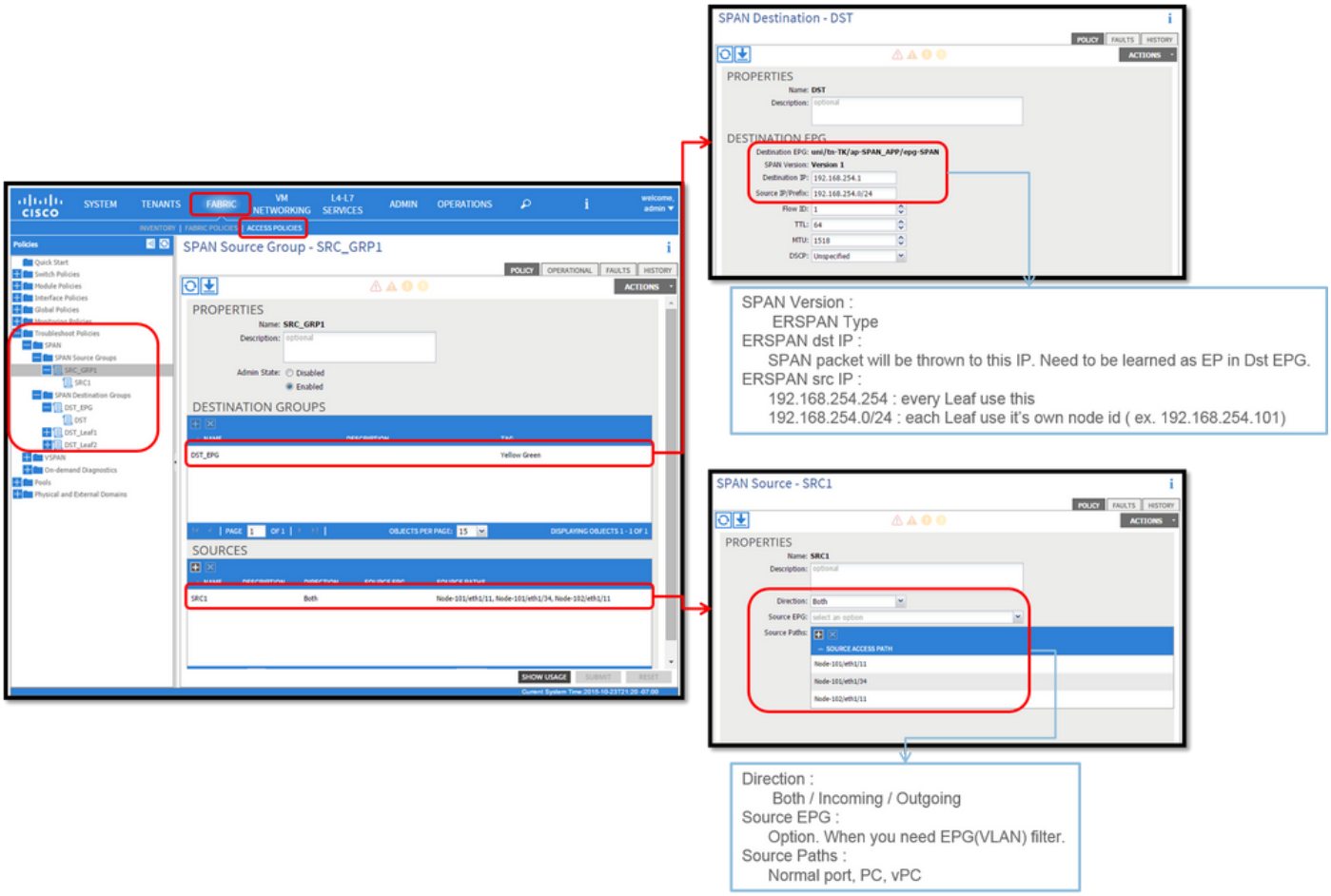
UI可能与当前版本不同，但配置方法相同。

接入SPAN (ERSPAN)

拓扑示例



配置示例



其中：

导航到FABRIC > ACCESS POLICIES > Troubleshoot Policies > SPAN。

- SPAN Source Groups
    - SPAN Destination Groups

SPAN Source Group 领带Destination和Sources。

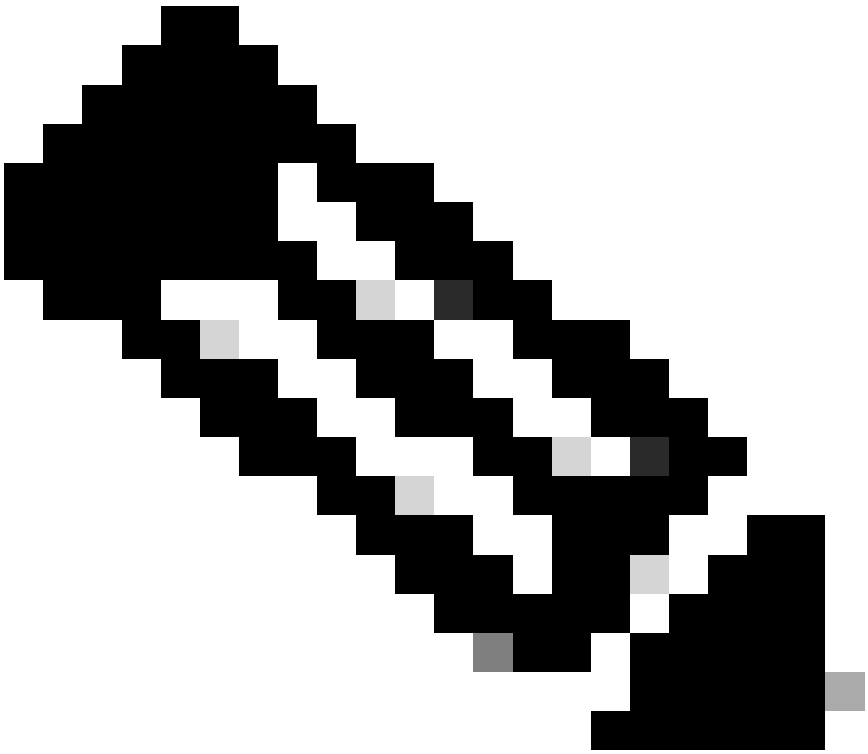
方式：

1. 创建SPAN Source Group(SRC_GRP1)。

- 在SPAN Source Group (SRC_GRP1)下创建SPAN Source (SRC1)。

- 为SPAN Source (SRC1)配置这些参数。

- 方向-源EPG（选项）
- 源路径（可以是多个接口）



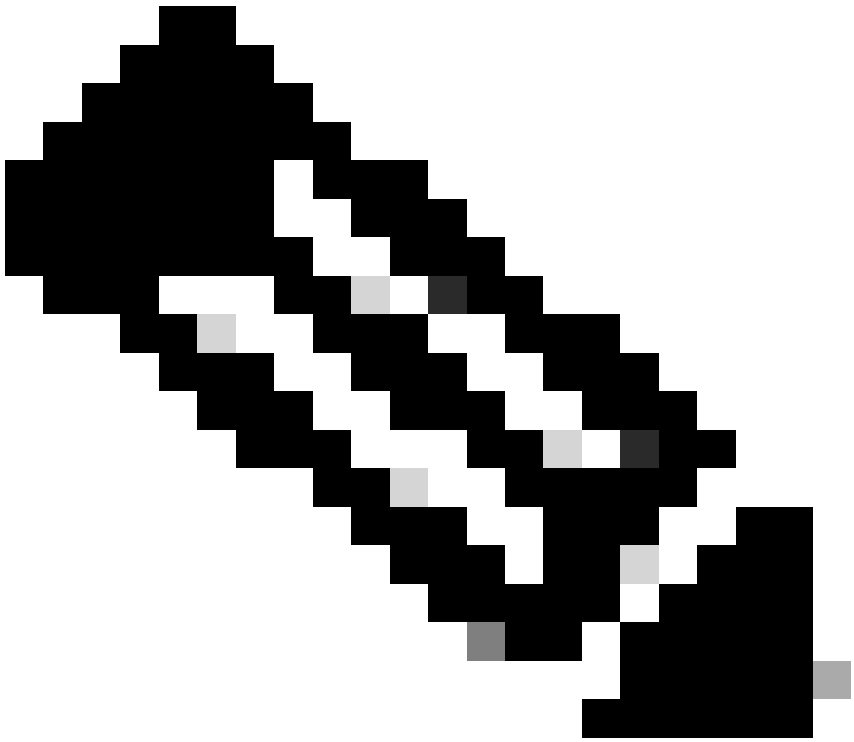
注意：请参阅图片了解每个参数的详细信息。

- 创建SPAN Destination Group(DST_EPG)。

- 创建SPAN Destination(DST)。

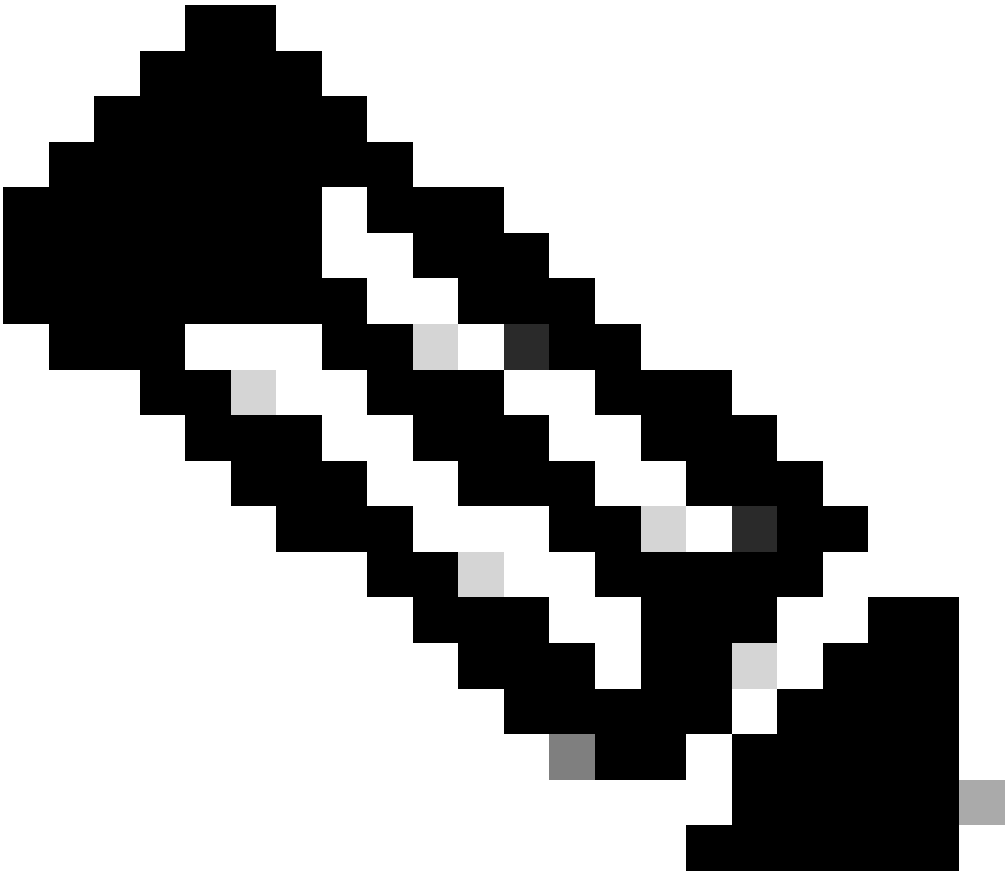
- 为SPAN Destination (DST)配置这些参数

- 目标EPG
-目的 IP
- 源IP/前缀(可以是任何IP。如果使用前缀，则源节点的节点ID用于未定义的位。例如，前缀：1.0.0.0/8 on node-101 => src IP 1.0.0.101)
 - 其他参数可以保留为默认值



注意：请参阅图片了解每个参数的详细信息。

- 确保SPAN Destination Group与相应的SPAN Source Group关联。
- 确保Admin State已启用。

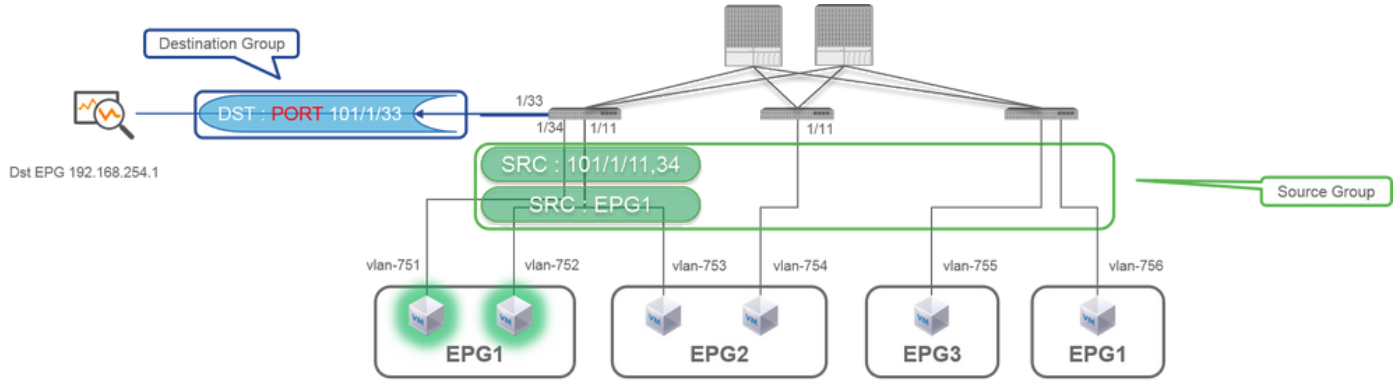


注意：当您在此管理状态下选择已禁用时，SPAN将停止。如果稍后重复使用所有策略，则无需删除它们。

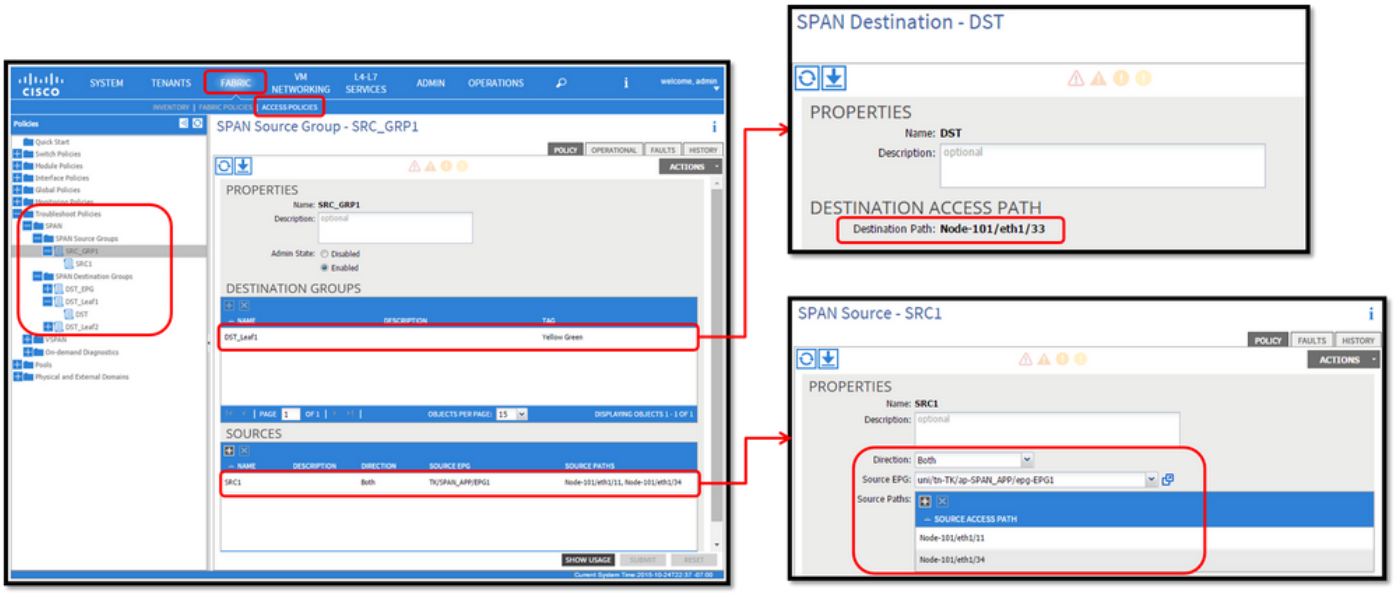
另外，请确保将ERSPAN的目标IP学习为指定目标EPG下的终端。在前面的示例中，必须在Tenant TK > Application profile SPAN_APP > EPG SPAN下获取192.168.254.1。或者，如果目标设备是静默主机，则可以在此EPG下将目标IP配置为静态终端。

接入SPAN（本地）

拓扑示例



配置示例



- 其中：

Fabric > ACCESS POLICIES > Troubleshoot Policies > SPAN

- SPAN Source Groups

- SPAN Destination Groups

SPAN Source Group 领带Destination 和 Sources.

- 方式：

1. 创建SPAN Source Group(SRC_GRP1)

SPAN Source

- 在SPAN Source Group (SRC_GRP1)下创建(SRC1)

- 为SPAN Source (SRC1)配置这些参数

- 方向
- 源EPG（选项）
- 源路径（可以是多个接口）

※请参阅图片了解每个参数的详细信息。

- 创建SPAN Destination Group(DST_Leaf1)

- 创建SPAN Destination(DST)

- 为SPAN Destination (DST)配置这些参数

- 目标接口和节点。

- 确保SPAN Destination Group与相应的SPAN Source Group关联。

-

确保Admin State处于启用状态。
当您在此管理状态上选择Disabled时，※ SPAN停止。如果稍后重复使用所有策略，则无需删除它们。

目标接口不需要按接口策略组进行任何配置。将电缆插入ACI枝叶交换机上的接口时，该功能会发挥作用。

限制:

- 对于本地SPAN，目标接口和源接口必须在同一枝叶上配置。

- 只要目标接口处于UP状态，它就不需要在EPG上。

- 当虚拟端口通道(vPC)接口指定为源端口时，无法使用本地SPAN
  但是，有一个应急方案。在第一代枝叶交换机上，作为vPC或PC成员的单个物理端口可配置为SPAN源。使用此本地SPAN可用于vPC端口上的流量。
  但是，此选项在第二代枝叶交换机上不可用(CSCvc11053)。相反，在2.1(2e)、2.2(2e)及后续版本中，通过CSCvc44643添加了“VPC组件PC”对SPAN的支持。这样，任何层代枝叶都可以将作为vPC成员的端口通道配置为SPAN源。这允许任何一代枝叶交换机对vPC端口上的流量使用本地SPAN。

- 指定第二代枝叶上端口信道的各个端口只会生成数据包的子集(也由于CSCvc11053)。

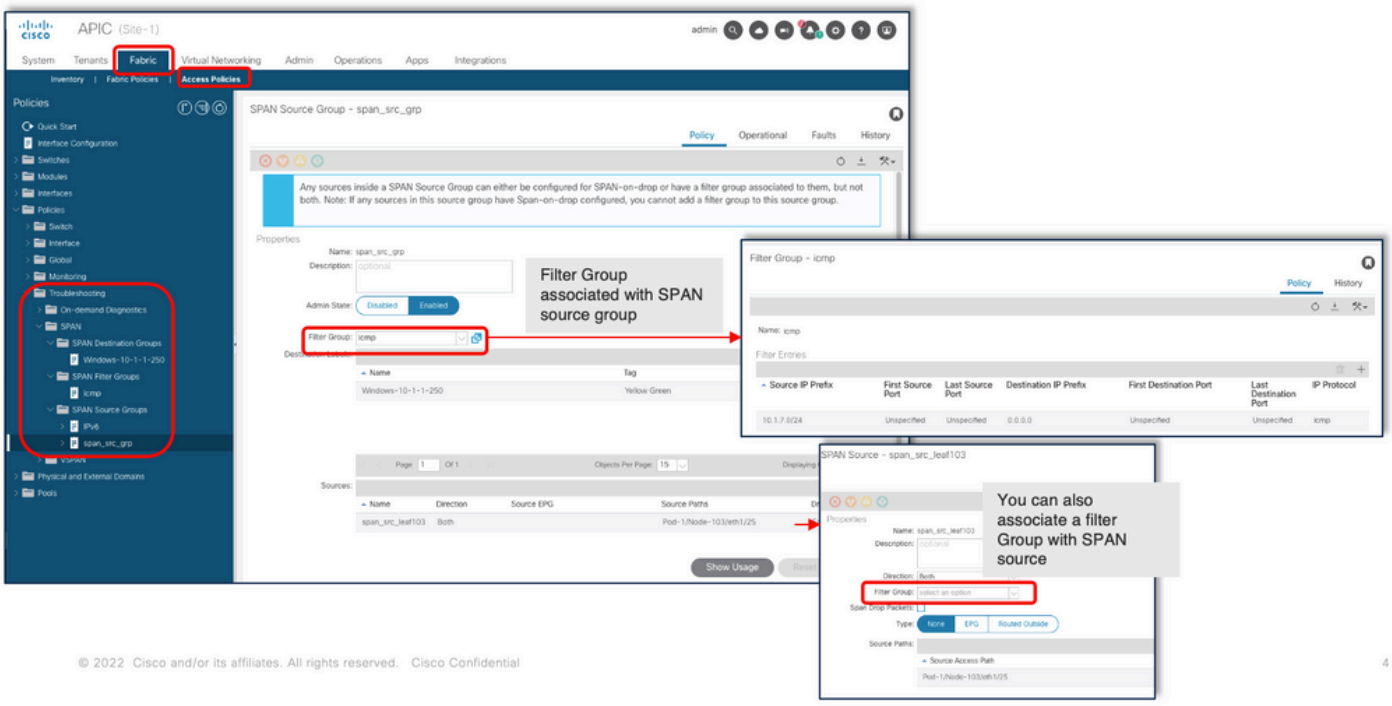
- PC和vPC不能用作本地SPAN的目的端口。从4.1(1)开始，PC可用作本地SPAN的目的端口。

访问SPAN -使用ACL过滤器

可以对访问SPAN源使用ACL过滤器。此功能提供跨特定流量或流入/流出SPAN源的能力。
当需要SPAN流量特定流量时，用户可以将SPAN Acl应用到源。
交换矩阵SPAN和租户SPAN源组/源中不支持它。
在过滤器组中添加过滤器条目时必须小心，因为它可以为当前使用过滤器组的每个源添加tcam条目。

过滤器组可以关联到：
-Span源：过滤器组用于过滤在此Span源下定义的所有接口上的流量。
-Span源组：过滤器组（例如x）用于过滤在该Span源组的每个Span源下定义的所有接口上的流量。

在此配置快照中，过滤器组应用于Span源组。

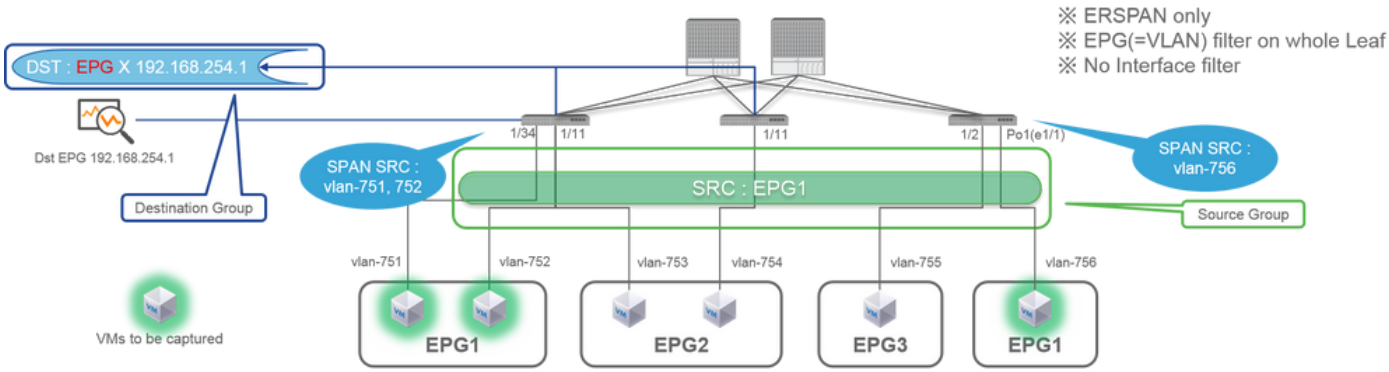


如果特定Span源已经与过滤器组（例如y）关联，则使用该过滤器组(y)来过滤该特定Span源下所有接口上的组

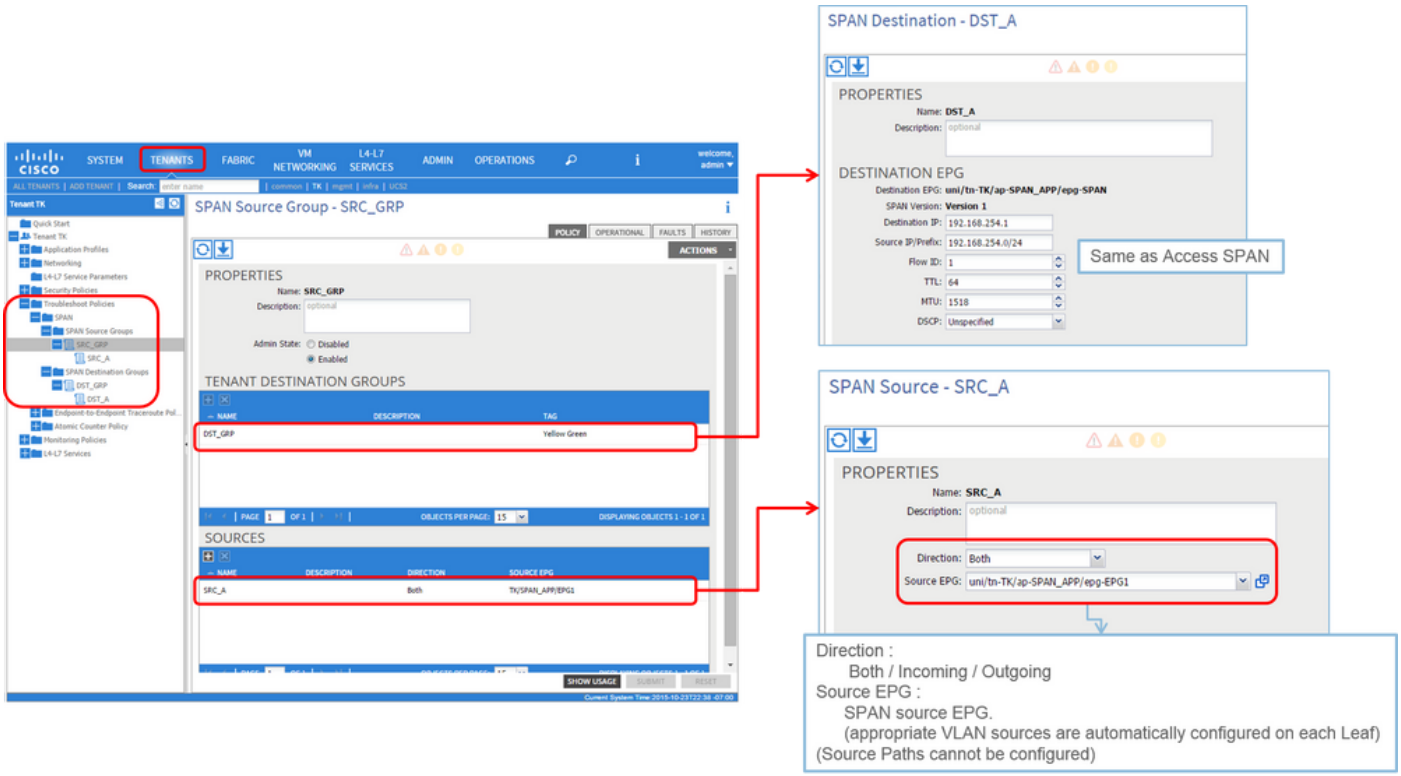
- 在源组应用的过滤器组自动应用于该源组中的所有源。
- 在源应用的过滤器组仅适用于该源。
- 过滤器组同时应用于源组以及该源组中的源，在源应用的过滤器组优先。
- 删除应用于源的过滤器组，自动应用应用于父源组的过滤器组。
- 应用于源组的过滤器组将被删除，它将从该源组中当前继承的所有源中删除。

租户SPAN (ERSPAN)

拓扑示例



配置示例



- 其中：

Tenants > {tenant name} > Troubleshoot Policies > SPAN

- SPAN Source Groups

- SPAN Destination Groups

※ SPAN源组关联Destination和 Sources.

- 方式：

1. 创建SPAN Source Group(SRC_GRP)

- 在SPAN Source Group (SRC_GRP)下创建SPAN Source(SRC_A)

- 为SPAN Source (SRC_A)配置这些参数
  - 方向
  - 源EPG

※请参阅图片了解每个参数的详细信息。

- 创建SPAN Destination Group(DST_GRP)

- 创建SPAN Destination(DST_A)

- 为SPAN Destination(DST_A)配置这些参数
  - 目标EPG
  -目的 IP
  - 源IP/前缀
  - 其他参数可以保留为默认值

※请参阅图片了解每个参数的详细信息。

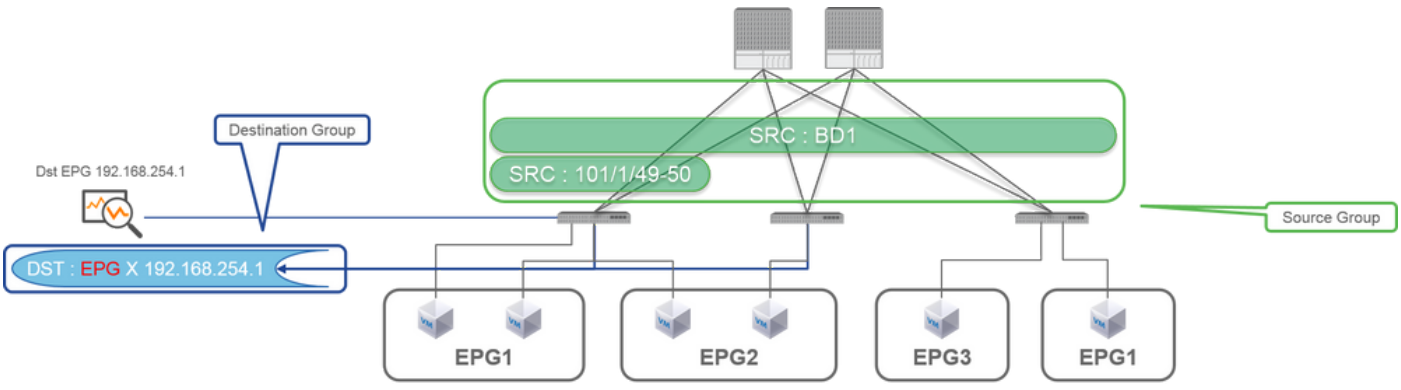
- 确保SPAN Destination Group与相应的SPAN Source Group关联。

- 确保Admin State处于启用状态。

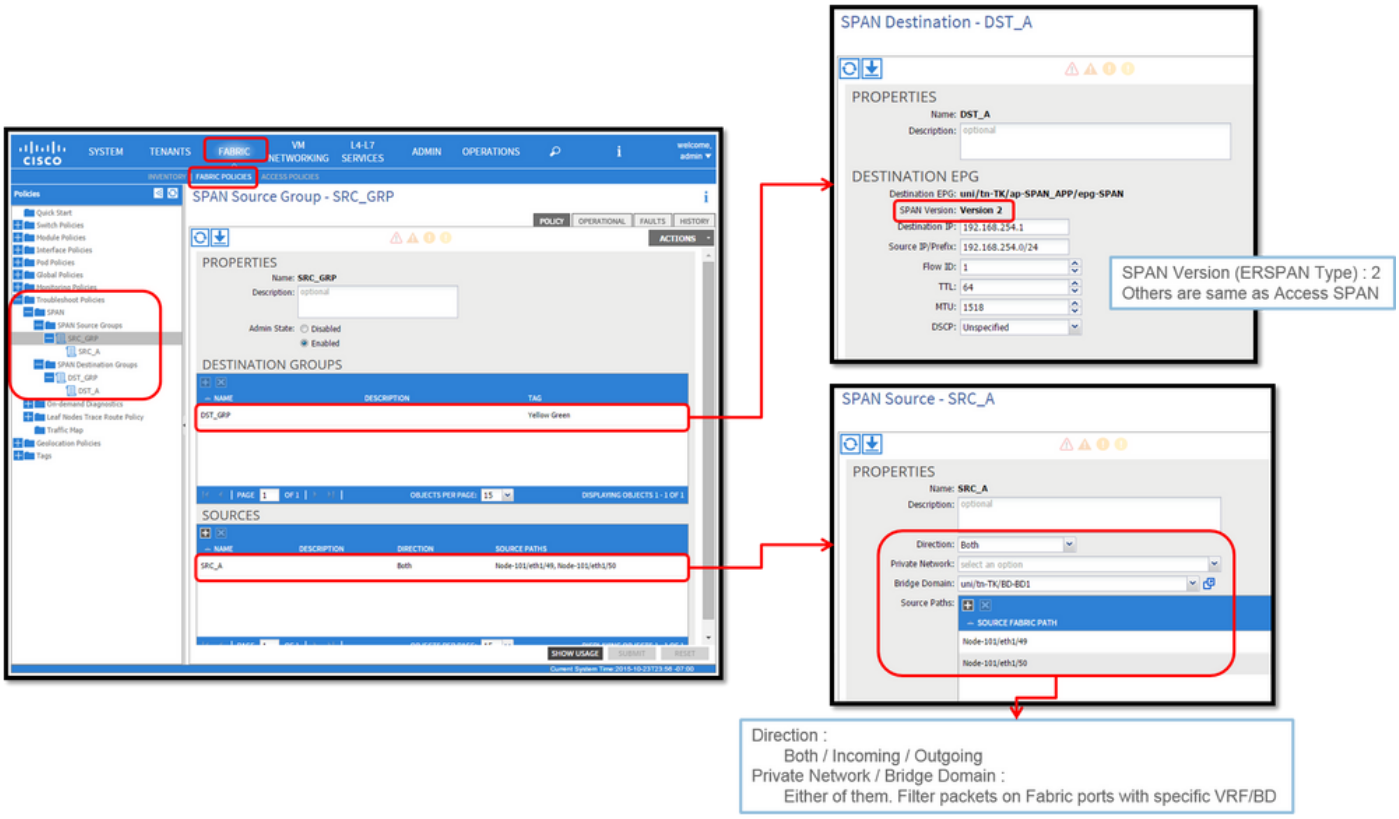
当您在此管理状态上选择Disabled时，※ SPAN停止。如果稍后重复使用所有策略，则无需删除它们。

交换矩阵SPAN (ERSPAN)

拓扑示例



配置示例



- 其中：

Fabric > FABRIC POLICIES > Troubleshoot Policies > SPAN

- Fabric

- SPAN Destination Groups

※SPAN Source GroupDestination和 Sources

- 方式：

1. 创建SPAN Source Group(SRC_GRP)

- 在SPAN Source Group (SRC_GRP)下创建SPAN Source(SRC_A)

- 为SPAN Source (SRC_A)配置这些参数
  - 方向
  - 专用网络（选项）
  - 桥接域（选项）
  - 源路径（可以是多个接口）

※请参阅图片了解每个参数的详细信息。

- 创建SPAN Destination Group(DST_GRP)

- 创建SPAN Destination(DST_A)

- 为SPAN Destination (DST_A)配置这些参数
  - 目标EPG
  -目的 IP
  - 源IP/前缀
  - 其他参数可以保留为默认值

※请参阅图片了解每个参数的详细信息。

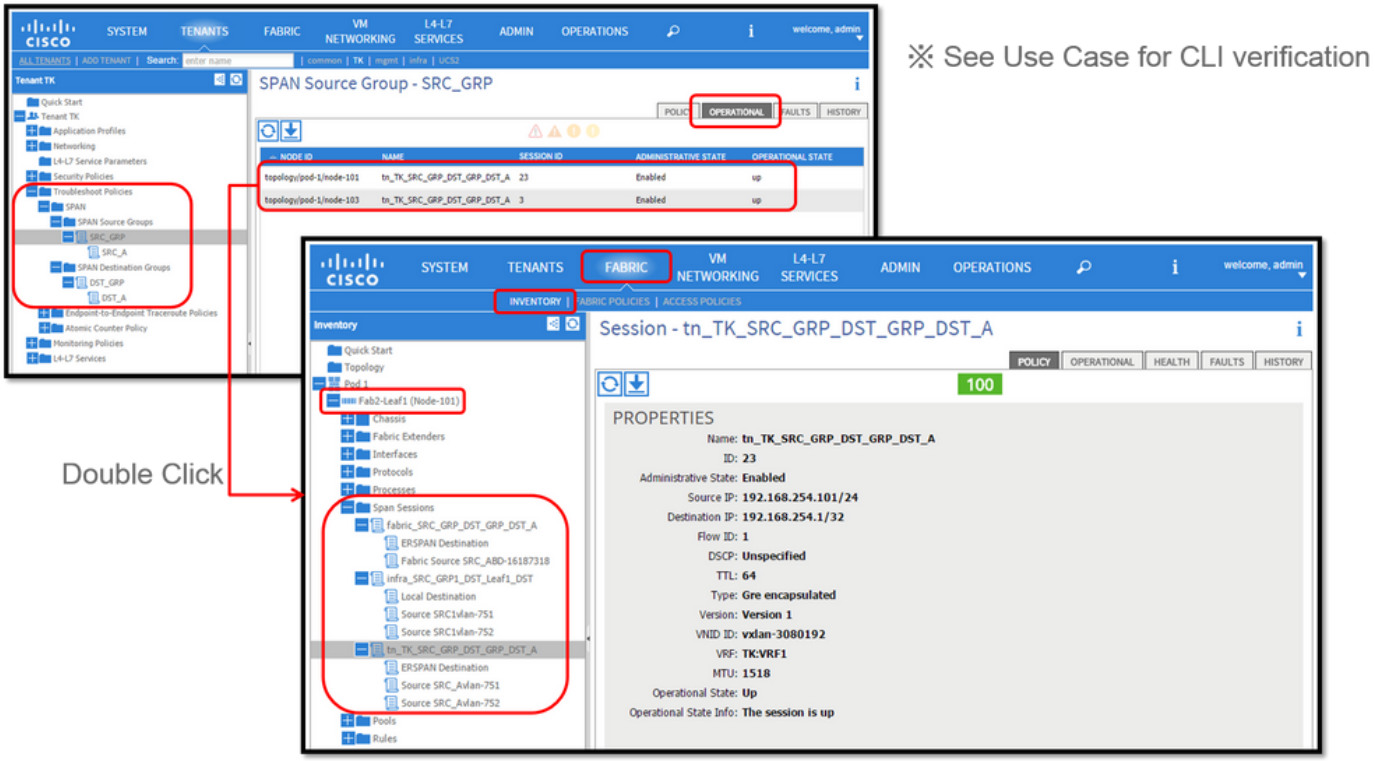
- 确保SPAN Destination Group与相应的SPAN Source Group关联。

- 确保Admin State 已启用。

※此Admin State上选择“已禁用”时，SPAN将停止。如果稍后重复使用所有策略，则无需删除它们。

尽管后面的“ERSPAN版本（类型）”部分对此进行了说明，但您可以看到ERSPAN版本II用于交换矩阵SPAN，版本I用于租户和接入SPAN。

GUI验证



- 验证SPAN配置策略

1. Fabric > ACCESS POLICIES > Troubleshoot Policies > SPAN > SPAN Source Groups > Operational tab

   - Fabric > FABRIC POLICIES > Troubleshoot Policies > SPAN > SPAN Source Groups > Operational tab
   - Tenants > {tenant name} > Troubleshoot Policies > SPAN > SPAN Source Groups > Operational tab

请确保运行状态为up。

- 对节点自身的SPAN会话进行验证

1. 从SPAN Configuration Policy或双击每个会话 Fabric > INVENTORY > Node > Span Sessions > { SPAN session name }

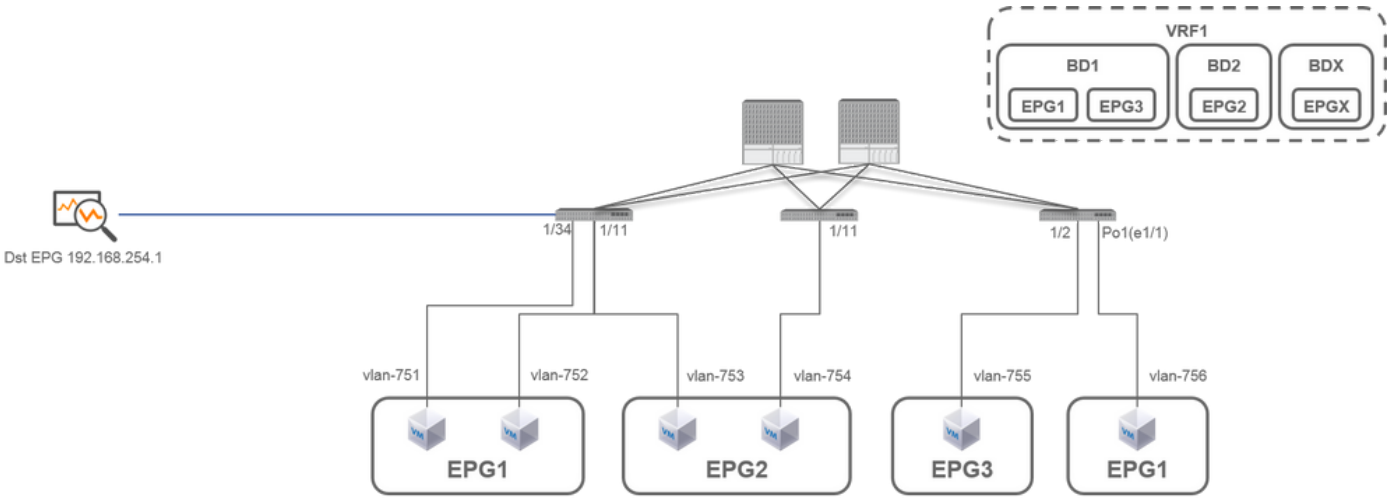
请确保运行状态为up。

SPAN会话命名约定：

- 交换矩阵SPAN：fabric_xxxx

- 访问SPAN：infra_xxxx
- 租户SPAN：tn_xxxx

选择ACI SPAN类型

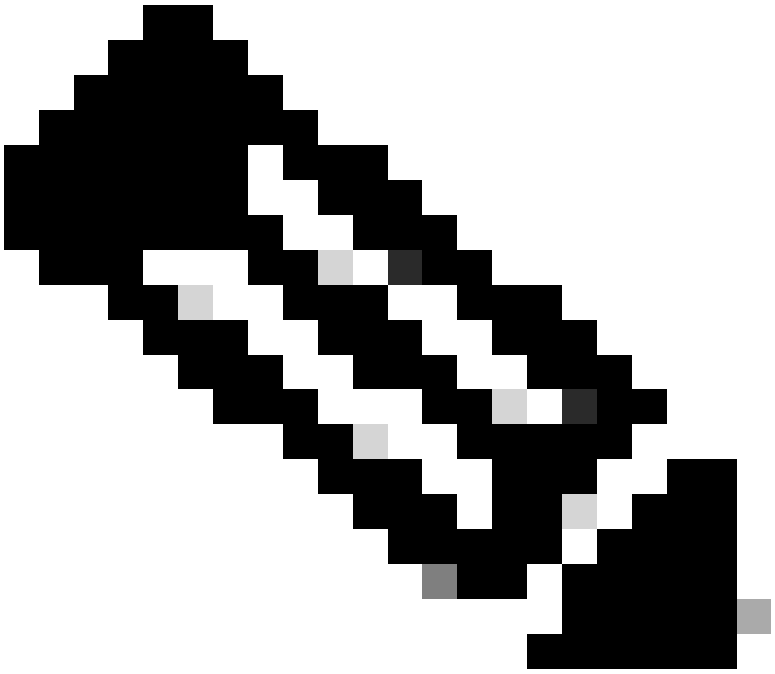


本部分描述了每个ACI SPAN类型的详细方案(Access, Tenant, Fabric)。每个方案的基本拓扑在上一部分中提到。

如果您了解这些场景，则可以根据需要选择适当的ACI SPAN类型，例如必须捕获仅特定接口上的数据包，或者必须捕获特定EPG上的所有数据包，而不考虑必须捕获的接口等等。

在思科ACI中，SPAN使用source group和destination group进行配置。源组包含多个源因素，例如接口或EPG。目标组包含目标信息，例如本地SPAN的目标接口或ESPAN的目标IP。

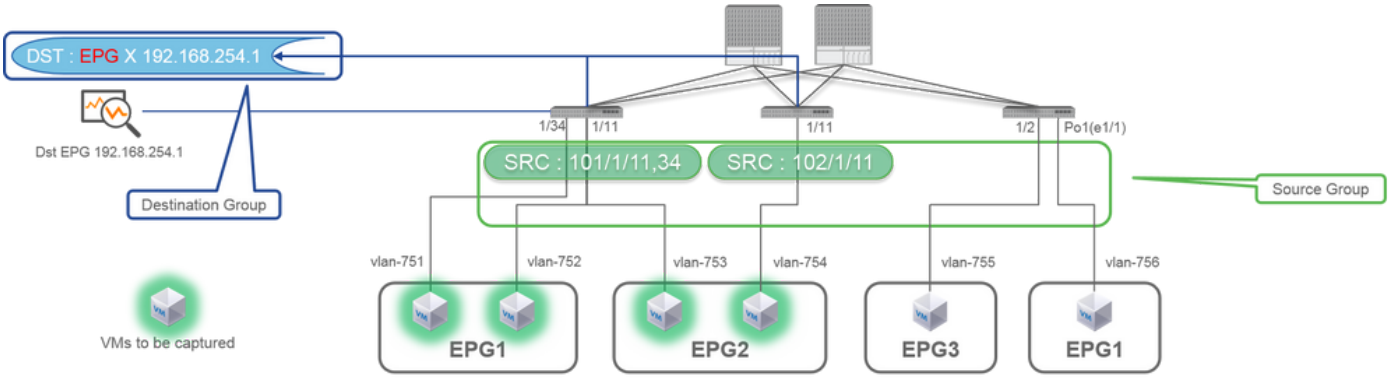
在捕获数据包后，请参阅“如何读取SPAN数据”部分以解码捕获的数据包。



注意：请重点关注每个拓扑中以绿灯突出显示的VM。每种场景都是从这些突出显示的VM捕获数据包。

接入SPAN (ERSPAN)

例 1.源“Leaf1 e1/11 e1/34和Leaf2 e1/11” | 目标“192.168.254.1”




```
Fab2-Leaf1# show monitor session all
   session 13
---------------
description       : Span session 13
type              : erspan
version           : version not specified
state             : up (active)
erspan-id         : 1
granularity       :
vrf-name          : TK:VRF1
acl-name          :
ip-ttl            : 64
ip-dscp           : ip-dscp not specified
destination-ip    : 192.168.254.1/32
origin-ip         : 192.168.254.101/24
mode              : access
source intf       :
    rx            : Eth1/11       Eth1/34
    tx            : Eth1/11       Eth1/34
    both          : Eth1/11       Eth1/34
source VLANs      :
    rx            :
    tx            :
    both          :
filter VLANs      : filter not specified
```

```
Fab2-Leaf2# show monitor session all
   session 12
---------------
description       : Span session 12
type              : erspan
version           : version not specified
state             : up (active)
erspan-id         : 1
granularity       :
vrf-name          : TK:VRF1
acl-name          :
ip-ttl            : 64
ip-dscp           : ip-dscp not specified
destination-ip    : 192.168.254.1/32
origin-ip         : 192.168.254.102/24
mode              : access
source intf       :
    rx            : Eth1/11
    tx            : Eth1/11
    both          : Eth1/11
source VLANs      :
    rx            :
    tx            :
    both          :
filter VLANs      : filter not specified
```

```
Fab2-Leaf3# show monitor session all
Note: No sessions configured
```

- Source Group
  - 枝叶1 e1/11
  - 枝叶1 e1/34
  - 枝叶2 e1/11
- Destination Group
  - EPG X上的192.168.254.1

访问SPAN可以为一个SPAN会话指定多个接口。它可以捕获从指定接口传入或传出的所有数据包，而不管它们的EPG如何。

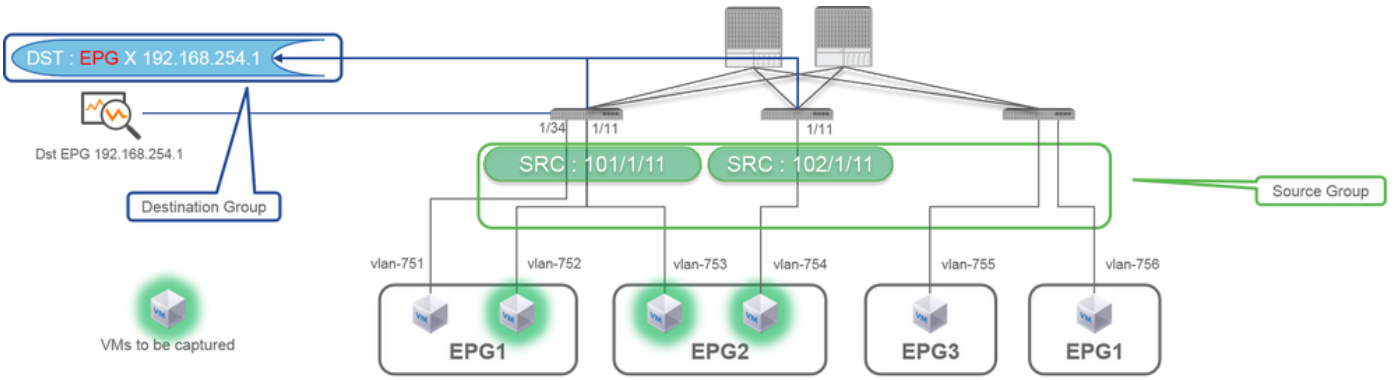
将多个接口指定为来自多个枝叶交换机的源组时，目标组必须是ERSPAN，而不是本地SPAN。

在本示例中，它会从EPG1和EPG2上的所有VM复制数据包。

CLI检查点

- 请确保状态为“up (active)”
- “destination-ip”是ERSPAN的目标IP
    - “origin-ip”是ERSPAN的源IP

案例 2.源“Leaf1 e1/11和Leaf2 e1/11”｜目标“192.168.254.1”




```
Fab2-Leaf1# show monitor session all
   session 2
---------------
description       : Span session 2
type              : erspan
version           : version not specified
state             : up (active)
erspan-id         : 1
granularity       :
vrf-name          : TK:VRF1
acl-name          :
ip-ttl            : 64
ip-dscp           : ip-dscp not specified
destination-ip    : 192.168.254.1/32
origin-ip         : 192.168.254.101/24
mode              : access
source intf       :
    rx            : Eth1/11
    tx            : Eth1/11
    both          : Eth1/11
source VLANs      :
    rx            :
    tx            :
    both          :
filter VLANs      : filter not specified
```

```
Fab2-Leaf2# show monitor session all
   session 3
---------------
description       : Span session 3
type              : erspan
version           : version not specified
state             : up (active)
erspan-id         : 1
granularity       :
vrf-name          : TK:VRF1
acl-name          :
ip-ttl            : 64
ip-dscp           : ip-dscp not specified
destination-ip    : 192.168.254.1/32
origin-ip         : 192.168.254.102/24
mode              : access
source intf       :
    rx            : Eth1/11
    tx            : Eth1/11
    both          : Eth1/11
source VLANs      :
    rx            :
    tx            :
    both          :
filter VLANs      : filter not specified
```

```
Fab2-Leaf3# show monitor session all
Note: No sessions configured
```

- **源组**
  - 枝叶1 e1/11
  - 枝叶2 e1/11
- **目标组**
  - EPG X上的192.168.254.1

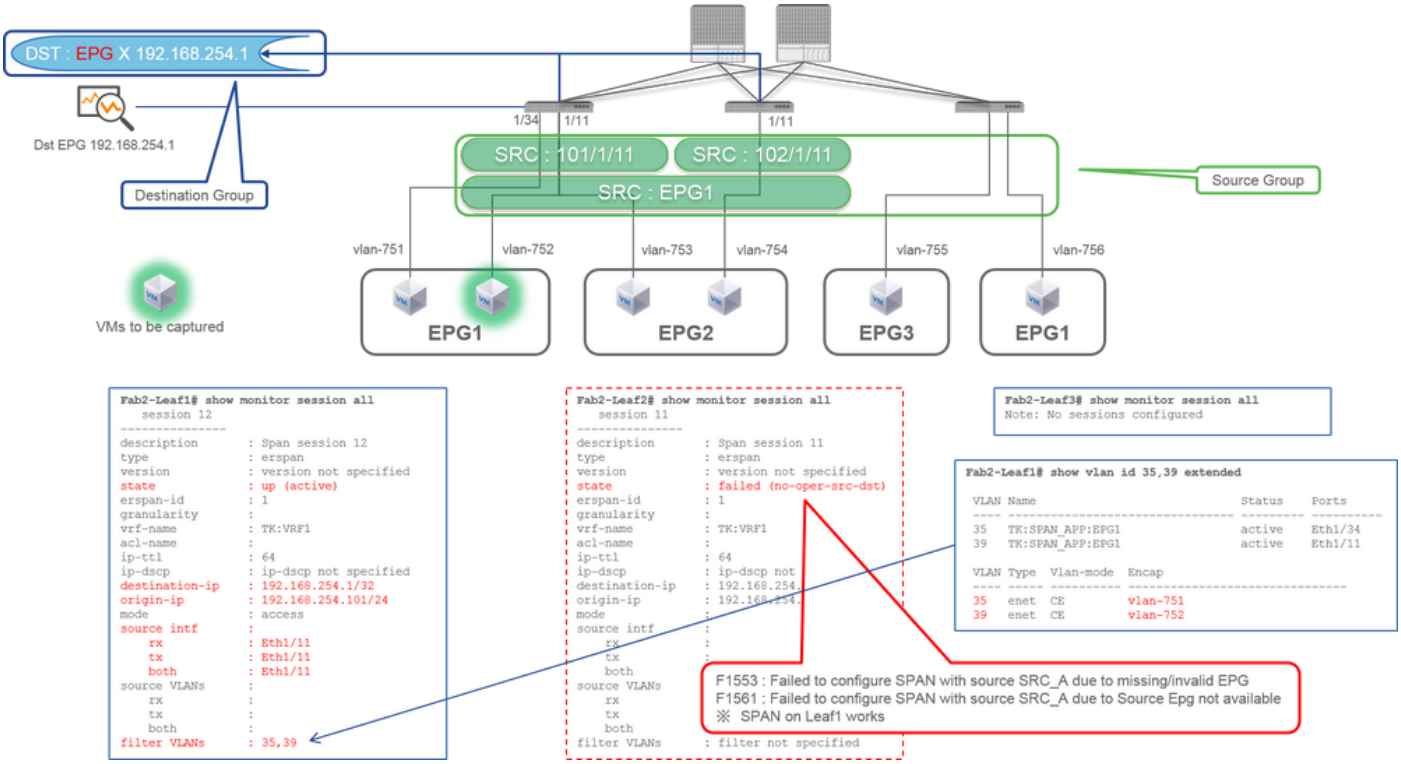
在本示例中，从之前案例1中配置的SPAN源组中删除枝叶1 e1/34。

本示例中的关键点是无论使用EPG如何，接入SPAN都可以指定源接口。

CLI检查点

- 枝叶1上的源接口已从“Eth1/11 Eth1/34”更改为“Eth1/11”

案例 3.源“Leaf1 e1/11 & Leaf2 e1/11 & EPG1过滤器” ｜目标“192.168.254.1”



- **源组**
  - 枝叶1 e1/11
  - 枝叶2 e1/11
  - 过滤EPG1
- **目标组**
  - EPG X上的192.168.254.1

本示例显示接入SPAN还可以在源端口上指定特定EPG。当多个EPG在单个接口上流动且仅需要捕获该接口上EPG1的流量时，这非常有用。

由于EPG1未部署在Leaf2上，Leaf2的SPAN发生故障，故障为F1553和F1561。但是，Leaf1上的SPAN仍然有效。

此外，由于EPG1在Leaf1上使用两个VLAN (VLAN-751,752)，因此会为SPAN会话自动添加两个VLAN过滤器。

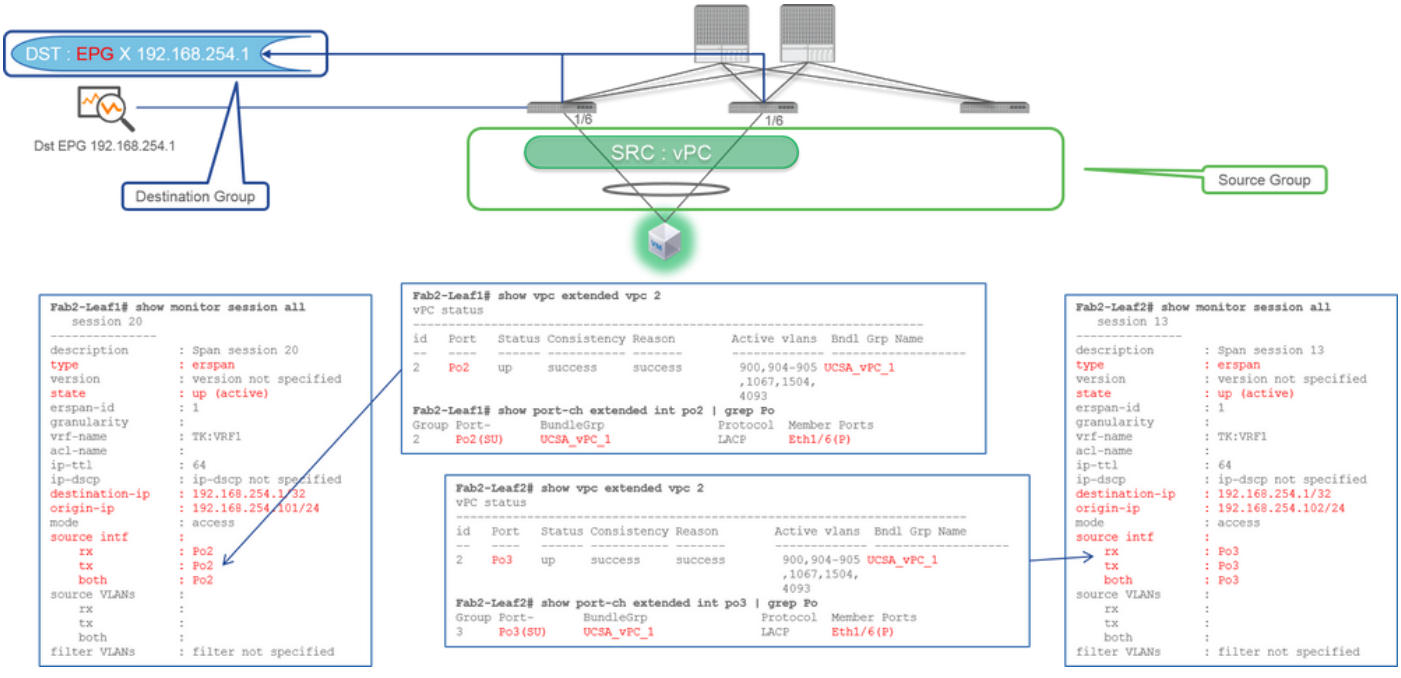
请注意，CLI上的VLAN ID (35， 39)是内部VLAN，即所谓的PI-VLAN（平台独立VLAN），它不是线路上的实际ID。如图所示，**show vlan extended** 命令显示了实际封装VLAN ID和PI-VLAN的映射。

通过此SPAN会话，我们可以只捕获枝叶1 e1/11上EPG1 (VLAN-752)的数据包，即使EPG2 (VLAN-753)在同一个接口上传输。

CLI检查点

- 根据用于过滤器的EPG添加过滤器VLAN。
- 如果枝叶交换机上没有对应的EPG，该枝叶交换机上的SPAN会话将失败。

案例 4.源“Leaf1-Leaf2 vPC”｜目标“192.168.254.1”

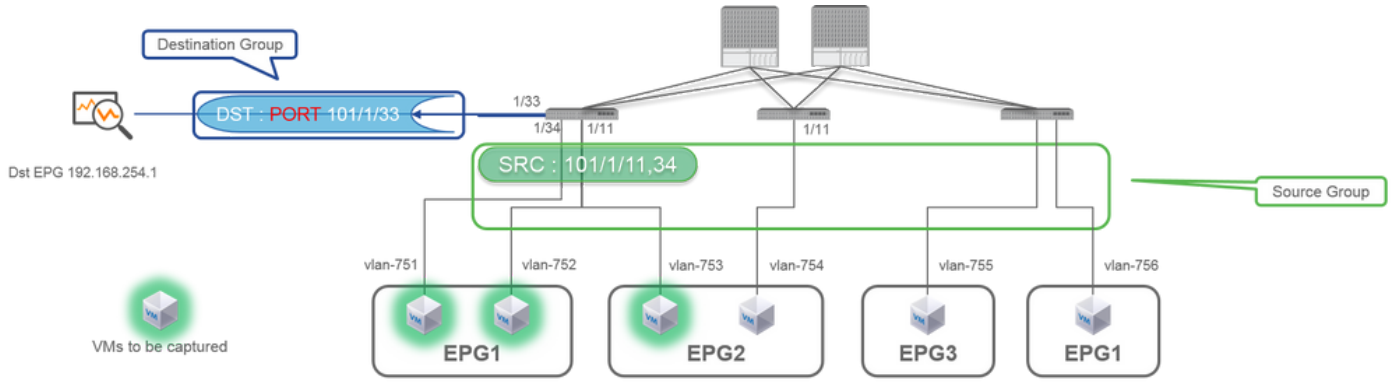


- **源组**
  - 枝叶1 – 2e1/11
- **目标组**
  - EPG X上的192.168.254.1

当vPC接口配置为源时，目标必须是远程IP (ERSPAN)，而不是接口（本地SPAN）

接入SPAN（本地SPAN）

例 1.源“Leaf1 e1/11 e1/34” | 目标“Leaf1 e1/33”




```
Fab2-Leaf1# show monitor session all
   session 14
---------------
description       : Span session 14
type              : local
state             : up (active)
mode              : access
source intf       :
    rx            : Eth1/11       Eth1/34
    tx            : Eth1/11       Eth1/34
    both          : Eth1/11       Eth1/34
source VLANs      :
    rx            :
    tx            :
    both          :
filter VLANs      : filter not specified
destination ports : Eth1/33
```

```
Fab2-Leaf2# show monitor session all
Note: No sessions configured
```

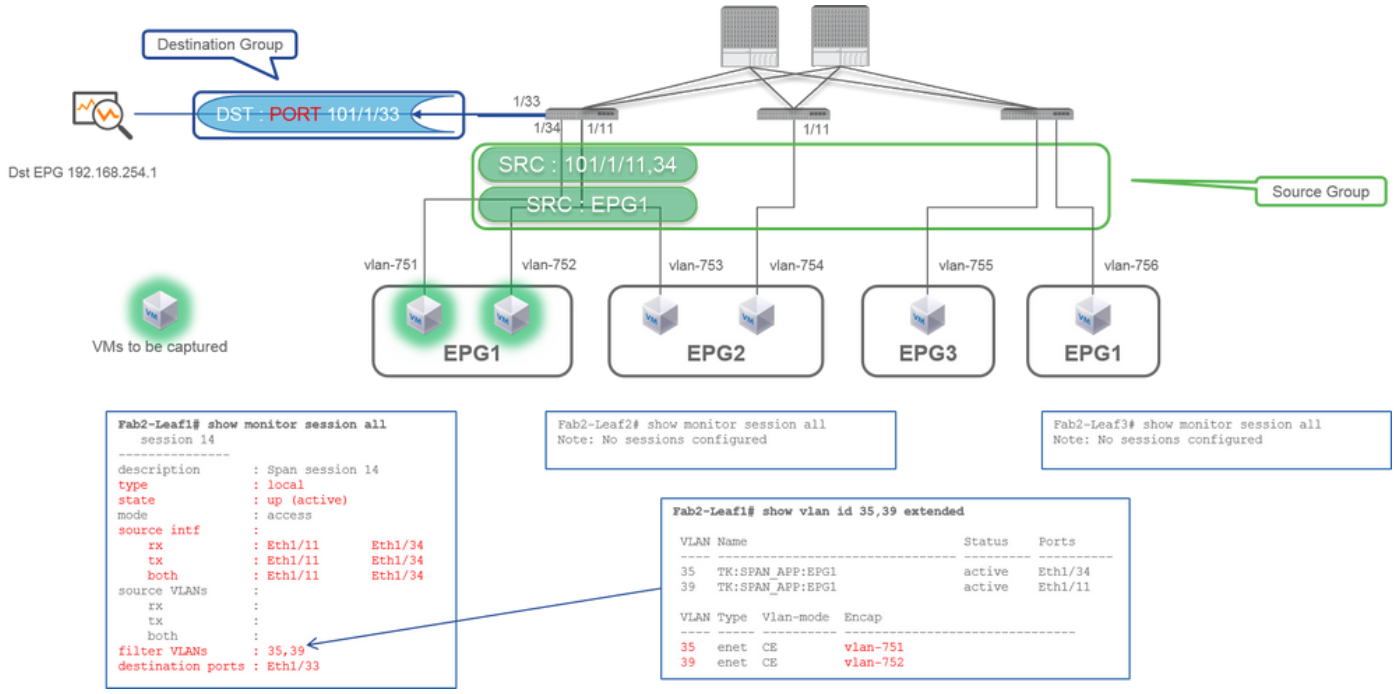
```
Fab2-Leaf3# show monitor session all
Note: No sessions configured
```

- **源组**
  - 枝叶1 e1/11
  - 枝叶1 e1/34
- **目标组**
  - 枝叶1 e1/33

接入SPAN也可以使用本地SPAN（即特定接口作为目标）

但是，在这种情况下，源接口必须与目标接口位于同一枝叶上。

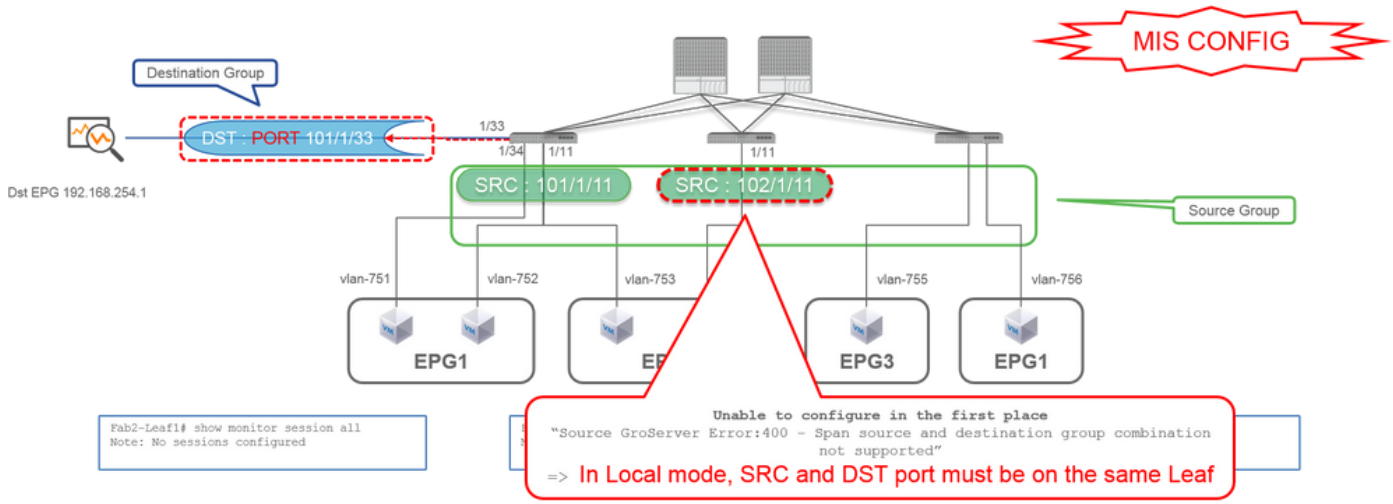
案例 2.Src“Leaf1 e1/11 e1/34和EPG1过滤器 | Dst "Leaf1 e1/33"



- **源组**
  - 枝叶1 e1/11
  - 枝叶1 e1/34
  - EPG1过滤器
- **目标组**
  - 枝叶1 e1/33

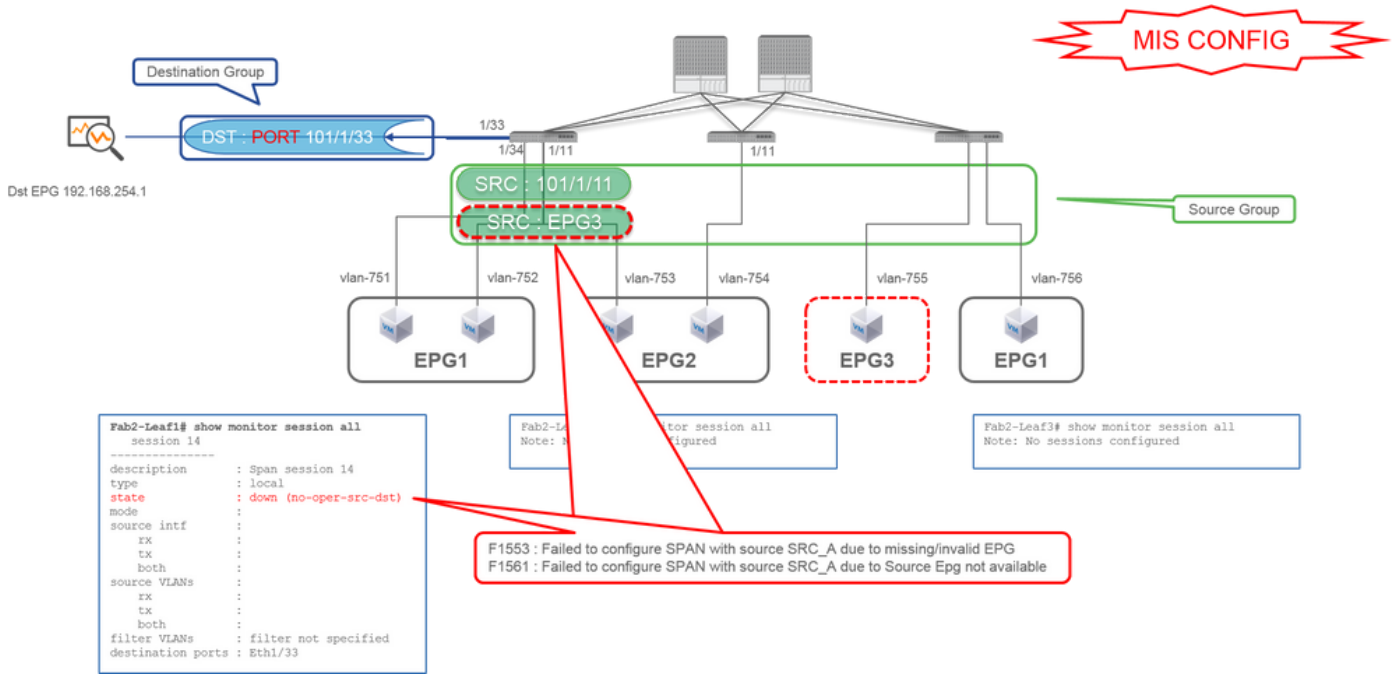
具有本地SPAN的接入SPAN也可以使用EPG过滤器和ERSPAN。

案例 3.源“Leaf1 e1/11和Leaf2 e/11”｜Dst“Leaf1 e1/33”（错误案例）



- **源组**
  - 枝叶1 e1/11
  - 枝叶2 e1/11
- **目标组**
  - 枝叶1 e1/33

案例 4.源“Leaf1 e1/11和EPG3过滤器” ｜ Dst“Leaf1 e1/33”（错误案例）



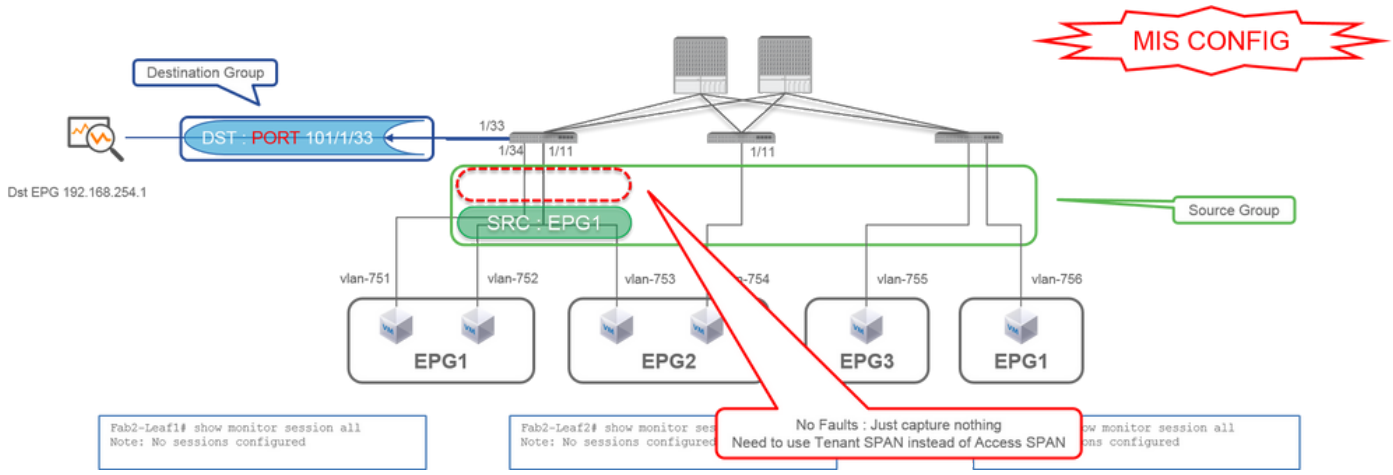
- **源组**

- 枝叶1 e1/11
- EPG3过滤器

- **目标组**
  - 枝叶1 e1/33

这与接入SPAN (ERSPAN)上的情况3相似，但在本示例中，由于EPG3在枝叶1上不存在，枝叶1上仅有一个SPAN会话发生故障。因此，SPAN根本不起作用。

案例5：源“EPG1过滤器” ｜ Dst“Leaf1 e1/33”（错误案例）

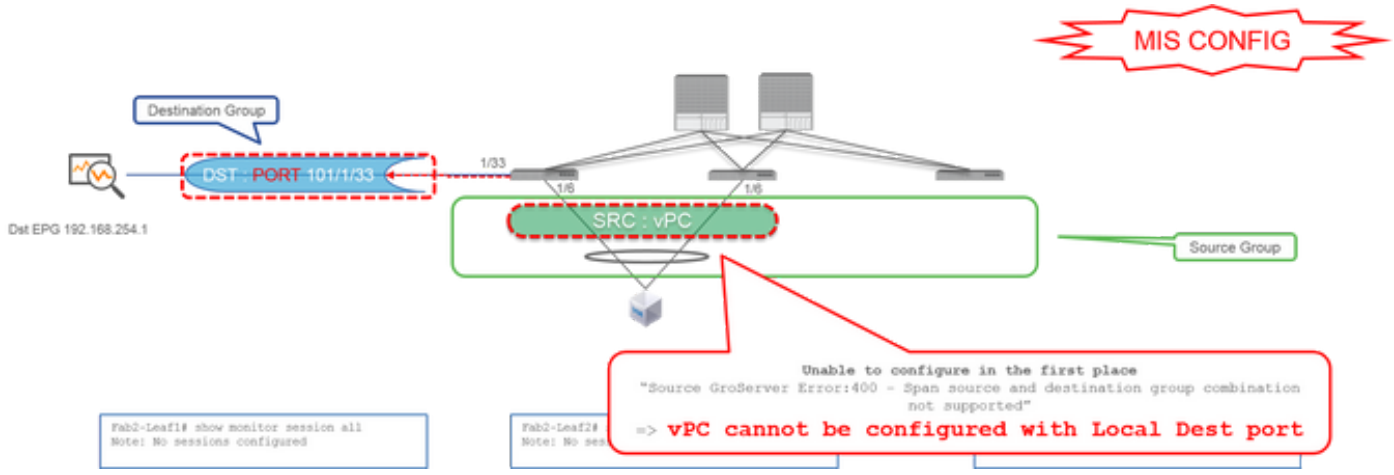


- **源组**
  - EPG1过滤器

- **目标组**
    - 枝叶1 e1/33

接入SPAN上的EPG过滤器仅在配置了源端口时有效。如果EPG是唯一要指定的源，则必须使用租户SPAN而不是访问SPAN。

案例 6.源“Leaf1 - Leaf2 vPC” | Dst“Leaf1 e1/33”（错误案例）



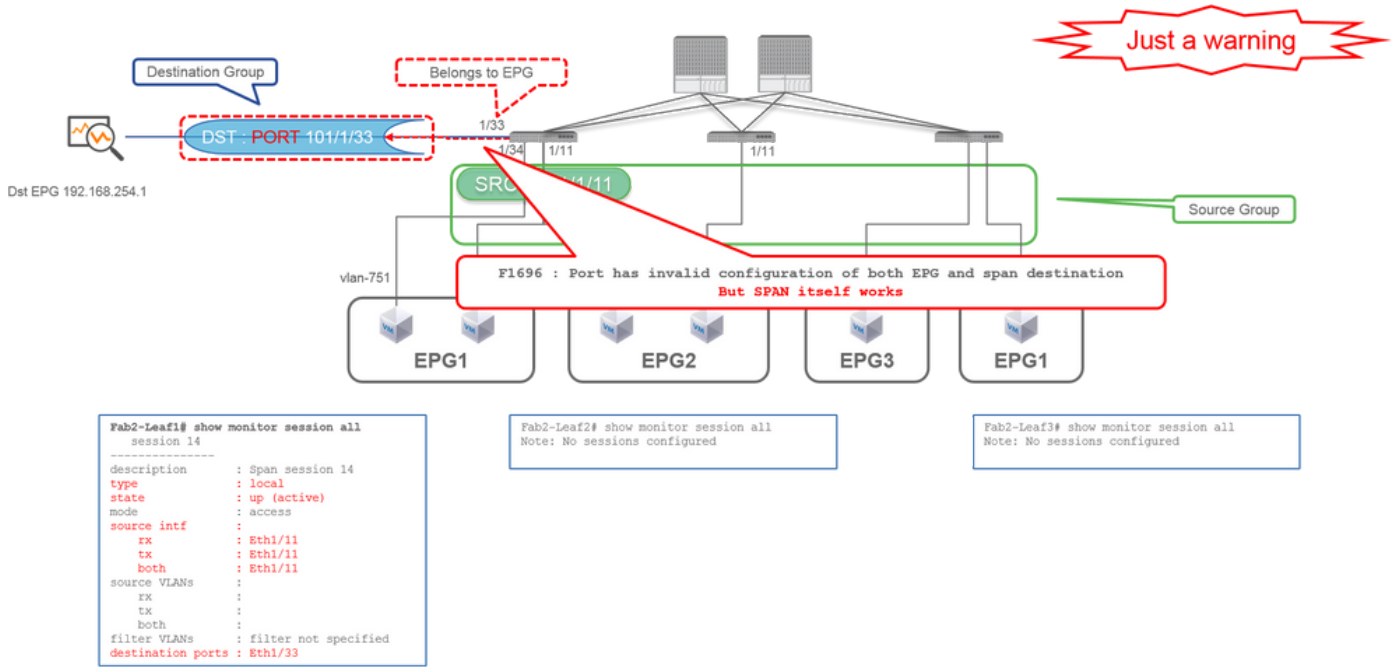
- **源组**
    - 枝叶1-2 vPC

- **目标组**

- 枝叶1 e1/33

无法将vPC接口配置为具有本地SPAN的源。请使用ERSPAN。请参阅案例4了解接入SPAN (ERSPAN)。

案例 7.源“Leaf1 e1/11 | Dst“Leaf1 e1/33和e1/33属于EPG”（工作期间出现故障）

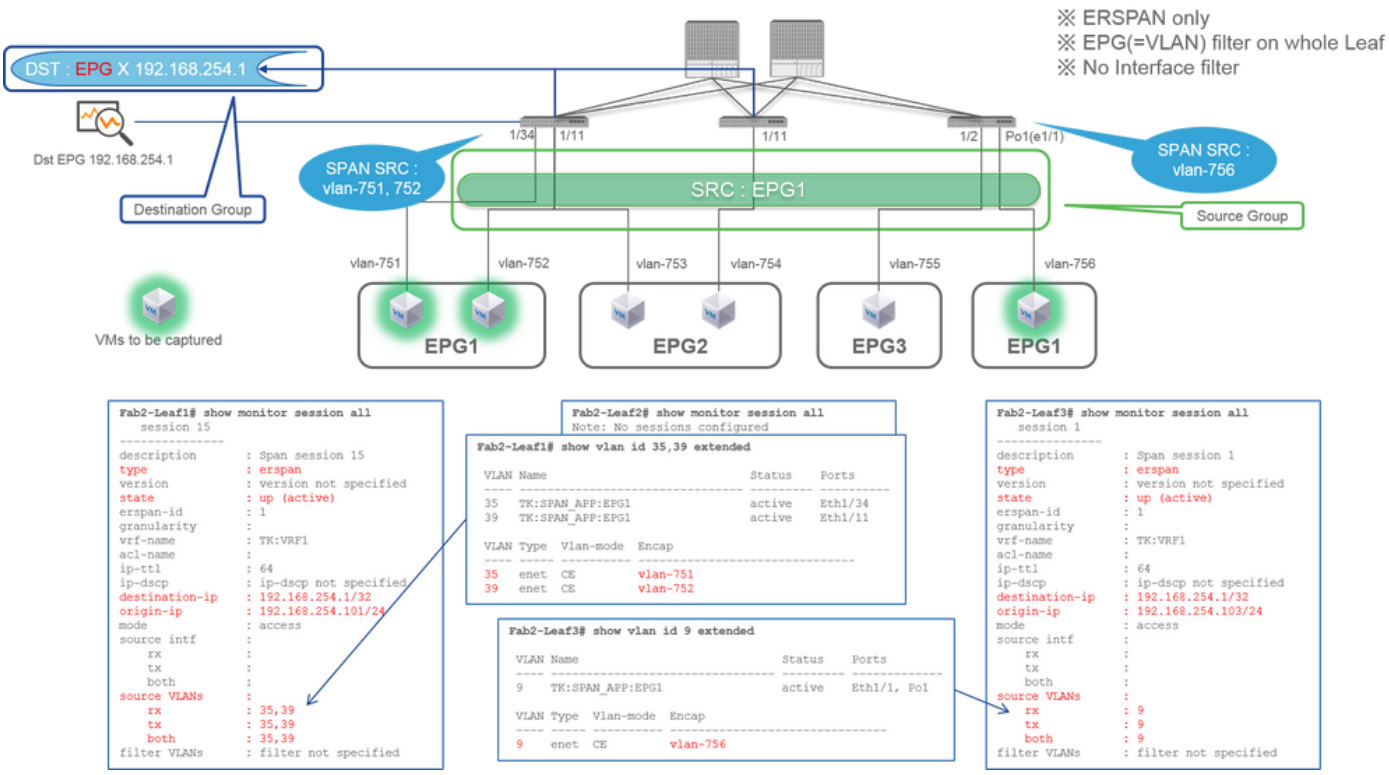


如果SPAN的目标I/F已属于EPG，则在物理I/F下会出现“F1696 ：端口的EPG和SPAN目标配置无效”故障。

但是，即使发生此故障，SPAN也可以正常工作。此故障只是对SPAN引起的额外流量的警告，因为它可能会影响同一I/F上客户的正常EPG流量。

租户SPAN (ERSPAN)

例 1.源“EPG1” | 目标 “192.168.254.1”



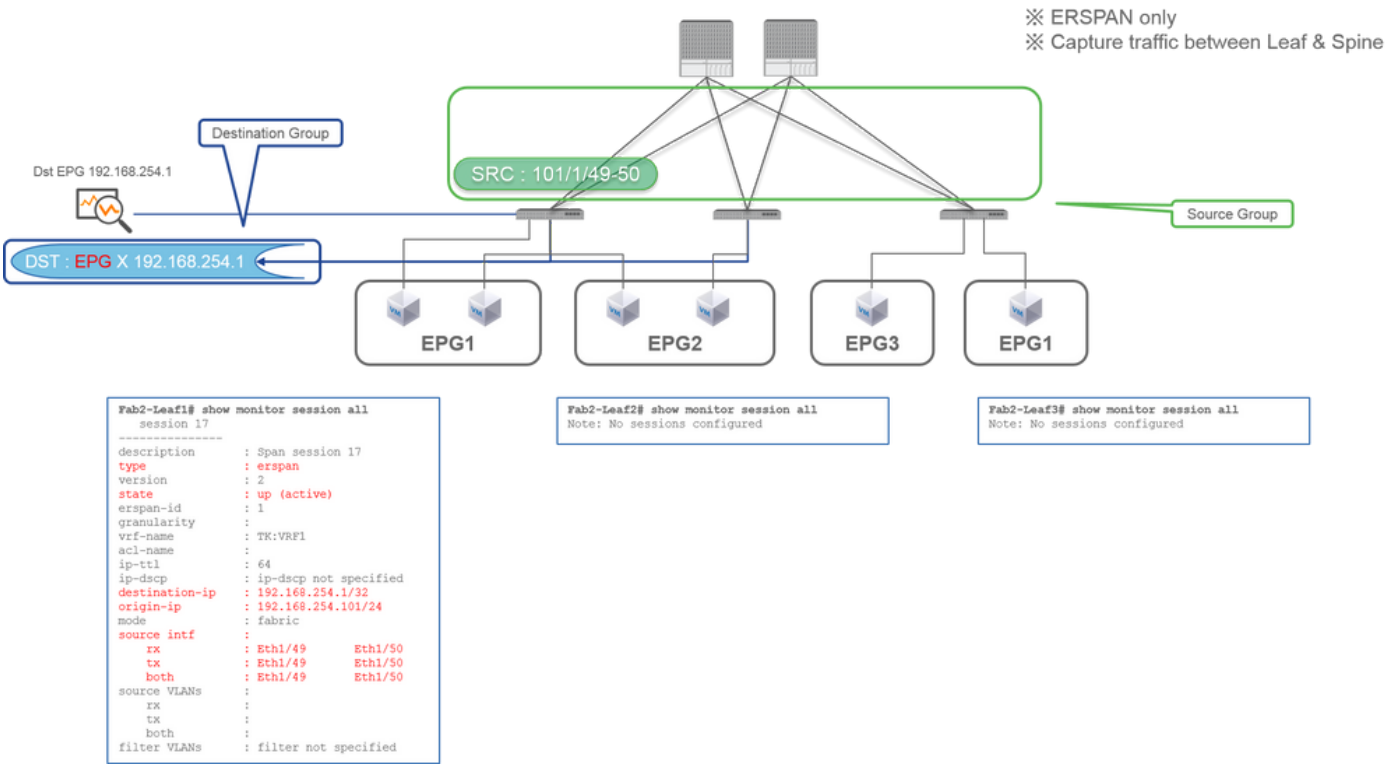
- **源组**
  - EPG1（无过滤器）
- **目标组**
  - EPG X上的192.168.254.1

租户SPAN使用EPG本身作为源，而接入SPAN仅使用EPG作为过滤器。

租户SPAN的关键点在于，您无需指定每个单独的端口，ACI会自动检测每个枝叶交换机上的适当VLAN。因此，在必须监控特定EPG的所有数据包且该EPG的终端属于枝叶交换机上的多个接口时，这将很有用。

交换矩阵SPAN (ERSPAN)

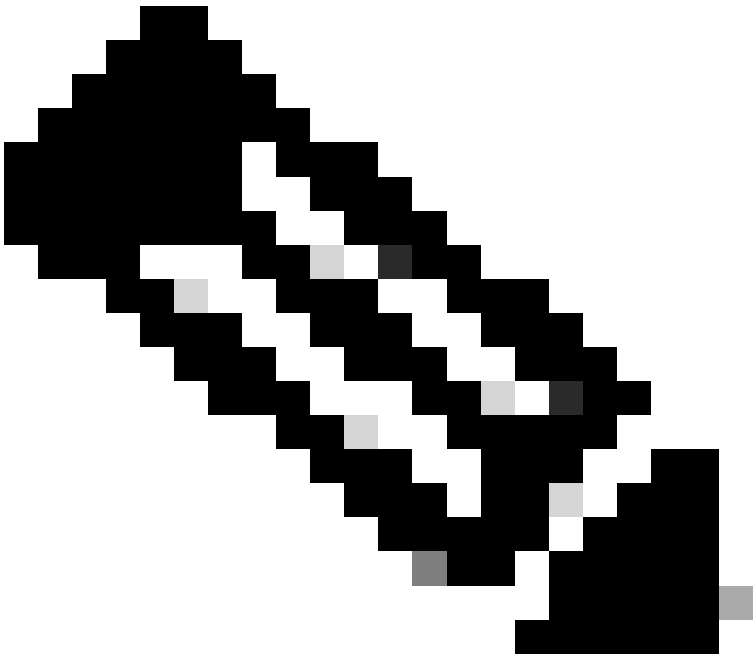
例 1.源“Leaf1 e1/49-50”｜目标“192.168.254.1”



- **源组**
  - 枝叶1 e1/49-50
- **目标组**
  - EPG X上的192.168.254.1

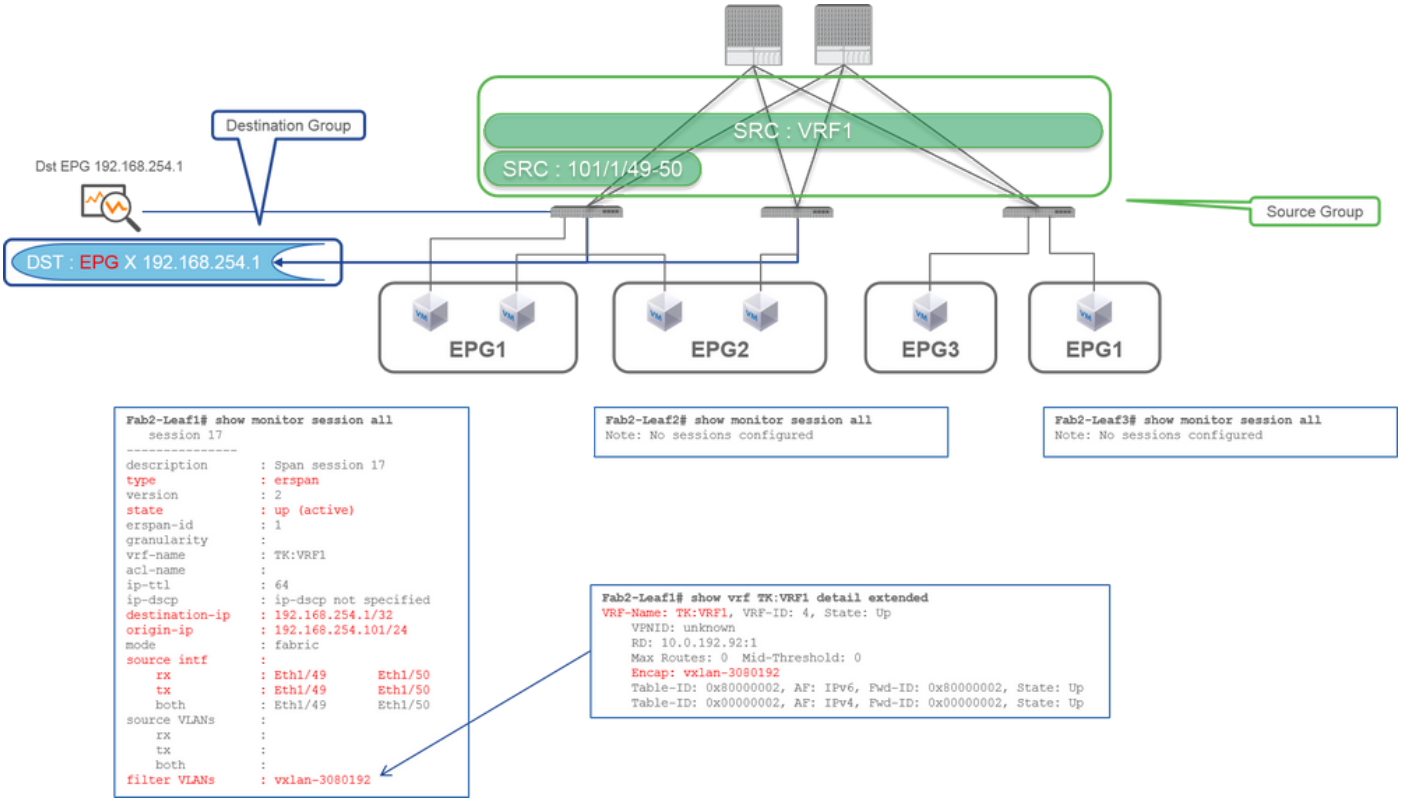
交换矩阵SPAN将交换矩阵端口指定为源，其中交换矩阵端口是枝叶交换机和主干交换机之间的接口。

当需要在枝叶交换机和主干交换机之间复制数据包时，此SPAN非常有用。但是，枝叶和主干交换机之间的数据包使用iVxLAN报头进行封装。所以要看它需要一点技巧。请参考“如何读取SPAN数据”。



注意：iVxLAN报头是增强型VxLAN报头，仅供ACI交换矩阵内部使用。

案例 2.源“Leaf1 e1/49-50和VRF过滤器” ｜目标“192.168.254.1”



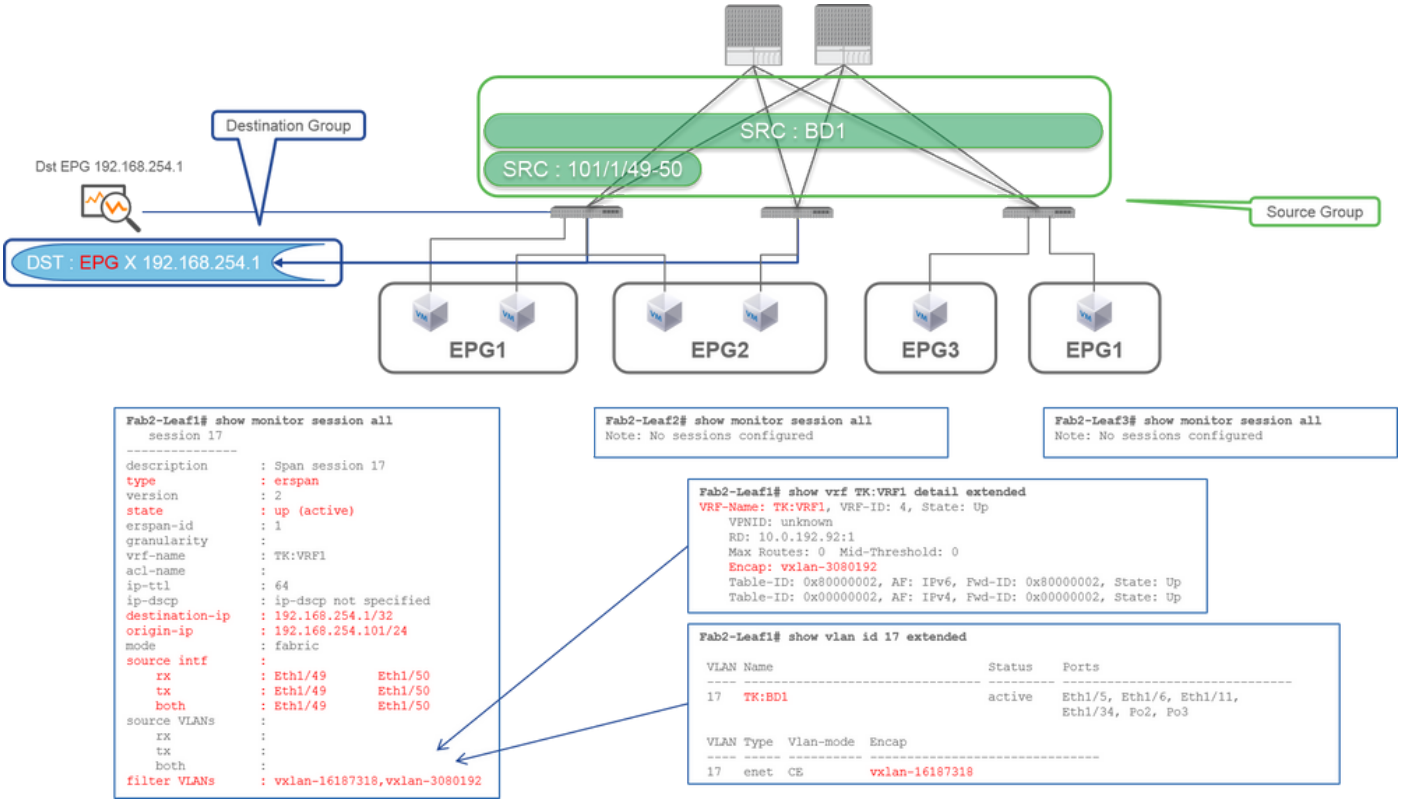
- **源组**
  - 枝叶1 e1/49-50
  - VRF过滤器
- **目标组**
  - EPG X上的192.168.254.1

交换矩阵SPAN可以使用过滤器和访问SPAN。但过滤类型不同。交换矩阵SPAN使用虚拟路由和转发(VRF)或BD作为过滤器。

如前所述，在思科ACI中，通过交换矩阵端口的数据包使用iVxLAN报头进行封装。此iVxLAN报头将VRF或BD信息作为虚拟网络标识符(VNID)。当数据包作为第2层(L2)转发时，iVxLAN VNID代表BD。当数据包作为第3层(L3)转发时，iVxLAN VNID代表VRF。

因此，当需要捕获交换矩阵端口上的路由流量时，请使用VRF作为过滤器。

案例 3.源“Leaf1 e1/49-50和BD过滤器” ｜目标“192.168.254.1”

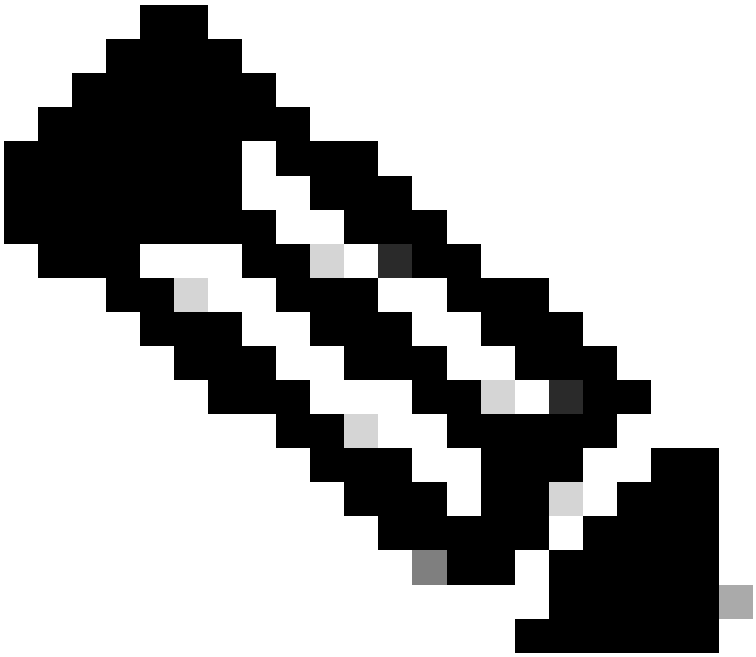


- **源组**
  - 枝叶1 e1/49-50
  - BD过滤器
- **目标组**

- EPG X上的192.168.254.1

如前例2所述，交换矩阵SPAN可以使用BD作为过滤器。

当需要捕获交换矩阵端口上的桥接流量时，使用BD作为过滤器。



注意：一次只能配置一个BD或VRF过滤器。

您需要在SPAN目标设备上执行什么操作？

只需在其上运行数据包捕获应用(例如tcpdump, wireshark)。无需配置ERSPAN目标会话或任何内容。

对于ERSPAN

由于SPAN数据包被转发到目标IP，请确保在具有目标IP的接口上运行ERSPAN捕获工具。

收到的数据包使用GRE报头进行封装。请参阅有关如何解码ERSPAN GRE报头的“如何读取ERSPAN数据”部分。

对于本地SPAN

请确保在连接到ACI枝叶交换机上的SPAN目标接口的接口上运行捕获工具。

此接口接收原始数据包。不需要处理ERSPAN报头。

如何读取ERSPAN数据

ERSPAN版本（类型）

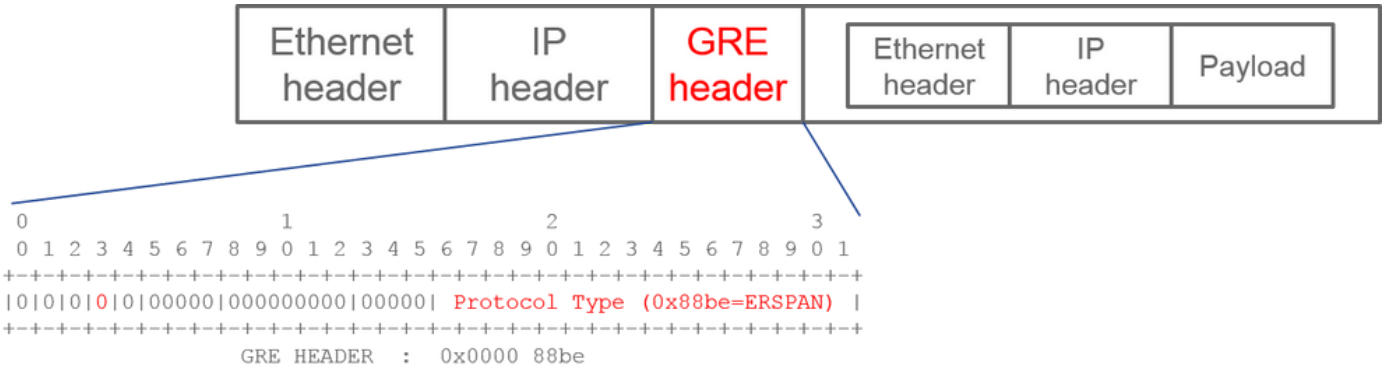
ERSPAN封装复制的数据包，将其转发到远程目标。GRE用于此封装。GRE报头上ERSPAN的协议类型为0x88be。

在Internet工程任务组(IETF)文档中，ERSPAN版本被描述为“类型”而不是“版本”。

ERSPAN有三种类型。I、II和III。此RFC草案中提及了ERSPAN类型。此外，此GRE RFC1701还有助于理解每种ERSPAN类型。

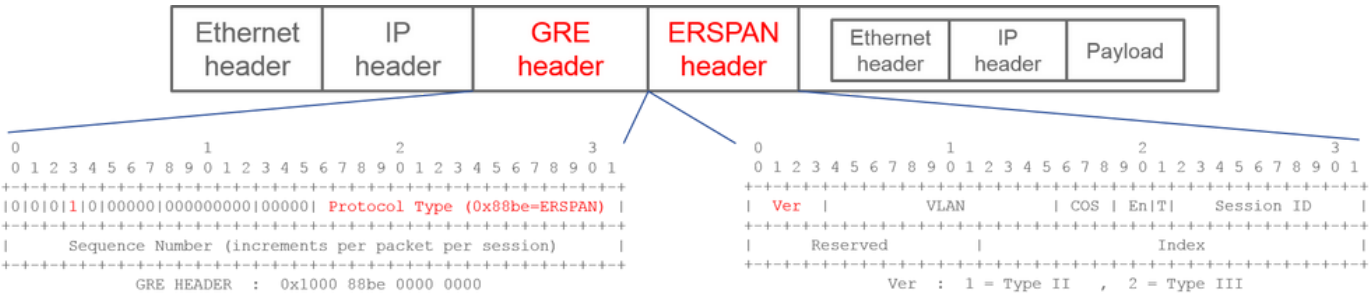
以下是每种类型的数据包格式：

ERSPAN类型I（由Broadcom Trident 2使用）



类型I不使用GRE报头上的序列字段。它甚至不使用GRE报头（如果是ERSPAN类型II和III）后面必须跟随的ERSPAN报头。Broadcom Trident 2仅支持此ERSPAN类型I。

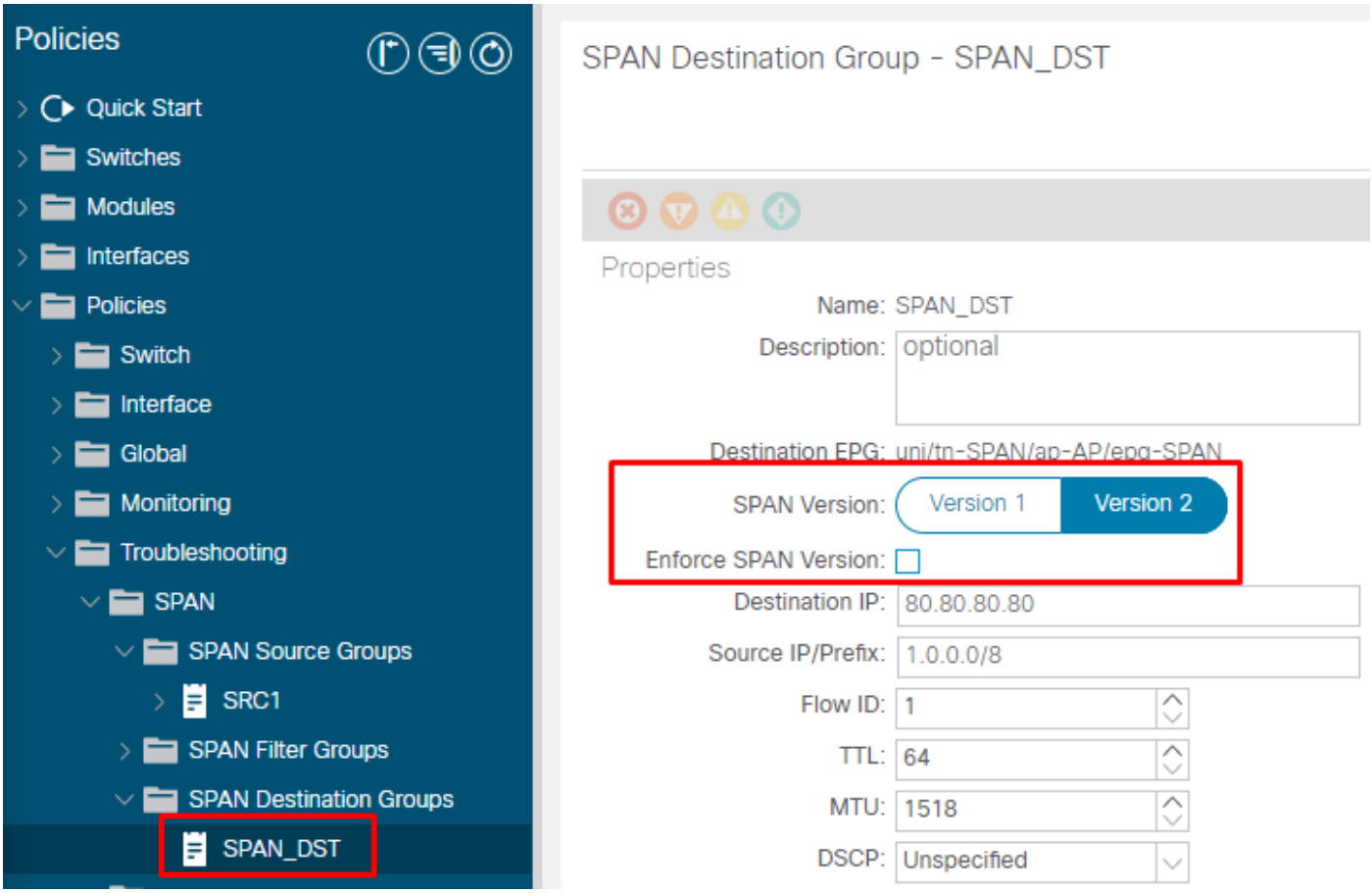
ERSPAN类型II或III



如果序列字段由S位激活，则该字段必须是ERSPAN类型II或III。ERSPAN报头上的版本字段用于标识ERSPAN类型。截至2016年3月20日，ACI不支持类型III。

如果接入或租户SPAN的SPAN源组在第一代和第二代节点上都存在源，则ERSPAN目标会从每一代节点接收两个ERSPAN类型I和II数据包。但是，Wireshark一次只能解码其中一种ERSPAN类型。默认情况下，它仅对ERSPAN类型II进行解码。如果您启用ERSPAN类型I的解码，Wireshark不会解码ERSPAN类型II。请参阅后面有关如何解码Wireshark上的ERSPAN类型I的部分。

为避免此类问题，您可以在SPAN目标组上配置ERSPAN类型。



- SPAN版本（版本1或版本2）：指ERSPAN类型I或II

- Enforce SPAN Version（选中或未选中）：如果源节点硬件上不支持所配置的ERSPAN类型，这决定了SPAN会话是否必须失败。

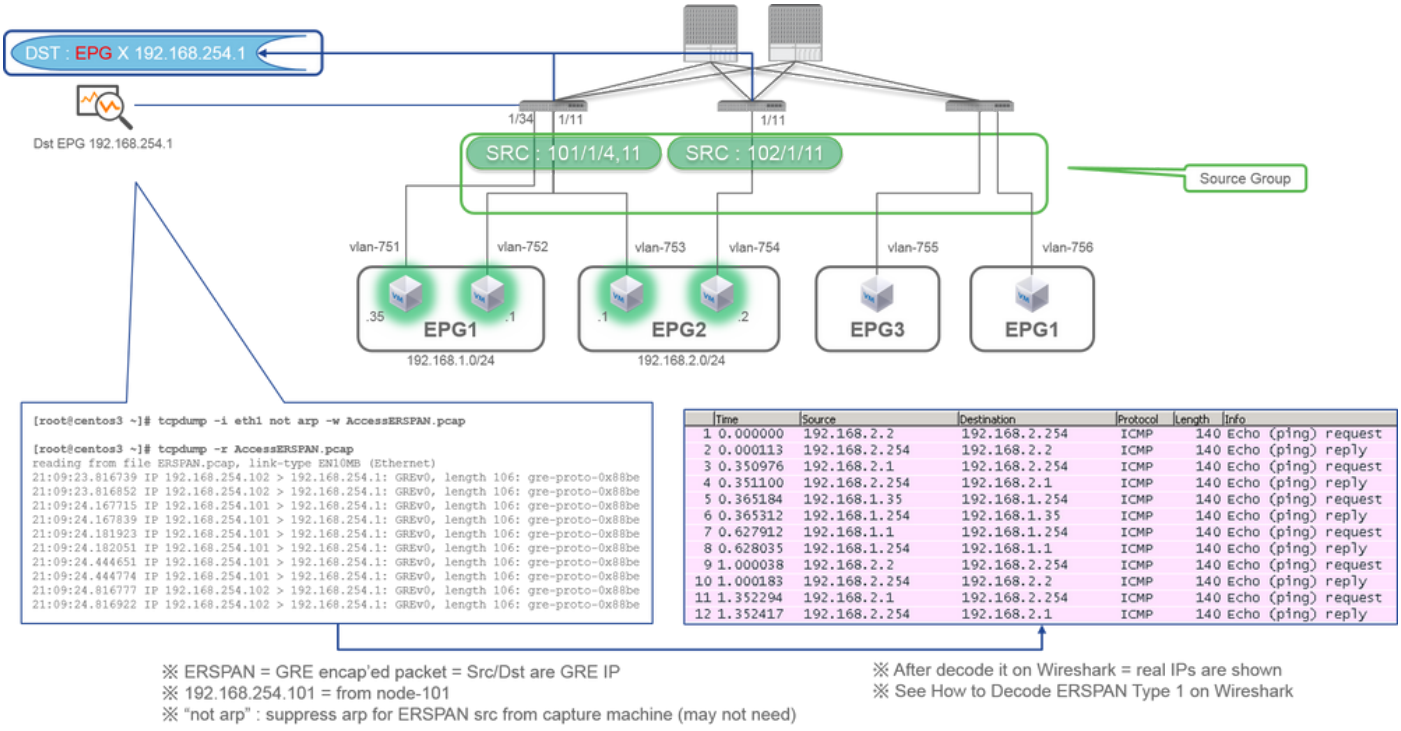
默认情况下，SPAN版本为版本2，而强制实施SPAN版本未选中。这意味着，如果源节点是支持ERSPAN类型II的第2代或更高版本，它将生成具有类型II的ERSPAN。如果源节点是不支持ERSPAN类型II的第1代（交换矩阵SPAN除外），它将回退到类型I，因为未选中Enforce SPAN Version。因此，ERSPAN目的地收到一种混合类型的ERSPAN。

此表说明了接入和租户SPAN的每个组合。

SPAN版本	实施**SPAN**版本	**第1代源**节点	**第2代源**节点
Version 2	未选中	使用类型I	使用类型II
Version 2	选中	失败	使用类型II
版本 1	未选中	使用类型I	使用类型I
版本 1	选中	使用类型I	使用类型I

ERSPAN数据示例

租户SPAN/接入SPAN (ERSPAN)



数据包由ERSPAN类型I封装，因此需要对其进行解码。这可以通过Wireshark完成。请参阅"如何解码ERSPAN类型1"部分。

捕获的数据包的详细信息（ERSPAN类型I）

```
[root@centos3 ~]# tcpdump -xxr AccessERSPAN.pcap -c 1
reading from file AccessERSPAN.pcap, link-type EN10MB (Ethernet)
21:09:23.816739 IP 192.168.254.102 > 192.168.254.1: GREv0, length 106: gre-proto-0x88be
        0x0000:  0050 56bb 3096 0022 bdf8 19ff 0800 4500
        0x0010:  007e 0000 0000 3d2f ff97 c0a8 fe66 c0a8                ESPAN Ethernet header         : Dst 0050.56bb.3096 , Src 0022.bdf8.19.ff
        0x0020:  fe01 0000 88be 0022 bdf8 19ff 0050 56bb                ERSPAN IP header              : Dst 192.168.254.1  , Src 192.168.254.102
        0x0030:  d6c2 8100 02f2 0800 4500 0054 0000 4000                GRE header (= ERSPAN Type I)  : 0x88be = ERSPAN (S bit off 0x0000)
        0x0040:  4001 b458 c0a8 0202 c0a8 02fe 0800 34cc                Ethernet header               : Dst 0022.bdf8.19ff , Src 0050.56bb.d6c2
        0x0050:  c847 0115 7404 2b56 0000 0000 8da9 0e00                Dot1Q header                  : VLAN 754
        0x0060:  0000 0000 1011 1213 1415 1617 1819 1a1b                IP header                     : Dst 192.168.2.254  , Src 192.168.2.2
        0x0070:  1c1d 1e1f 2021 2223 2425 2627 2829 2a2b
        0x0080:  2c2d 2e2f 3031 3233 3435 3637
```

交换矩阵SPAN (ERSPAN)

```
[root@centos3 ~]# tcpdump -r FabricERSPAN.pcap
reading from file FabricERSPAN.pcap, link-type EN10MB (Ethernet)
23:25:00.777331 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 54227, length 127: gre-proto-0x88be
23:25:00.777445 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 53328, length 82: gre-proto-0x88be
23:25:00.777567 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 54228, length 187: gre-proto-0x88be
23:25:00.777580 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 53329, length 82: gre-proto-0x88be
23:25:00.778068 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 53330, length 127: gre-proto-0x88be
23:25:00.817915 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 54229, length 82: gre-proto-0x88be
23:25:00.829676 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 54230, length 82: gre-proto-0x88be
23:25:00.829691 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 53331, length 82: gre-proto-0x88be
23:25:00.873953 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 54231, length 82: gre-proto-0x88be
23:25:00.873968 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 53332, length 82: gre-proto-0x88be
```

ERSPAN Type 2 is automatically decoded by Wireshark
※ be noted that this is still iVxLAN header

Filter: ip.addr==10.0.192.92 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
26	0.184754	10.0.192.92	10.0.32.66	UDP	198	Source port: 7248 Destination port: 48879
27	0.184893	10.0.192.95	10.0.192.92	UDP	198	Source port: 25168 Destination port: 48879
32	0.262735	10.0.192.92	10.0.32.65	UDP	160	Source port: 62672 Destination port: 48879
34	0.262855	10.0.192.92	239.255.255.255	UDP	156	Source port: 38745 Destination port: 48879
35	0.262868	10.0.192.92	239.255.255.255	UDP	156	Source port: 38745 Destination port: 48879
38	0.263458	10.0.192.92	225.0.213.250	UDP	160	Source port: 43738 Destination port: 48879
148	0.768367	10.0.0.1	10.0.192.92	TCP	116	56210-12151 [ACK] Seq=1 Ack=1 Win=770 Len=0
149	0.768486	10.0.192.92	10.0.0.1	TCP	116	[TCP ACKed unseen segment] 12151-56210 [ACK]
152	0.856142	10.0.192.92	225.0.213.248	UDP	164	Source port: 45334 Destination port: 48879
175	0.875130	10.0.192.92	10.0.0.1	TCP	116	[TCP Keep-Alive] [TCP ACKed unseen segment]
176	0.875252	10.0.0.1	10.0.192.92	TCP	116	[TCP Previous segment not captured] 56210-12
234	1.185477	10.0.192.92	10.0.32.66	UDP	198	Source port: 7248 Destination port: 48879
235	1.185606	10.0.192.95	10.0.192.92	UDP	198	Source port: 25168 Destination port: 48879
253	1.259119	10.0.192.92	10.0.0.1	TCP	116	57294-12375 [ACK] Seq=1 Ack=1 Win=270 Len=0

Wireshark自动对ERSPAN类型II进行解码。但是，它仍然由iVxLAN报头封装。

默认情况下，Wireshark无法理解iVxLAN报头，因为它是ACI内部报头。请参阅“如何解码iVxLAN报头”。

捕获的数据包的详细信息（ERSPAN类型II）

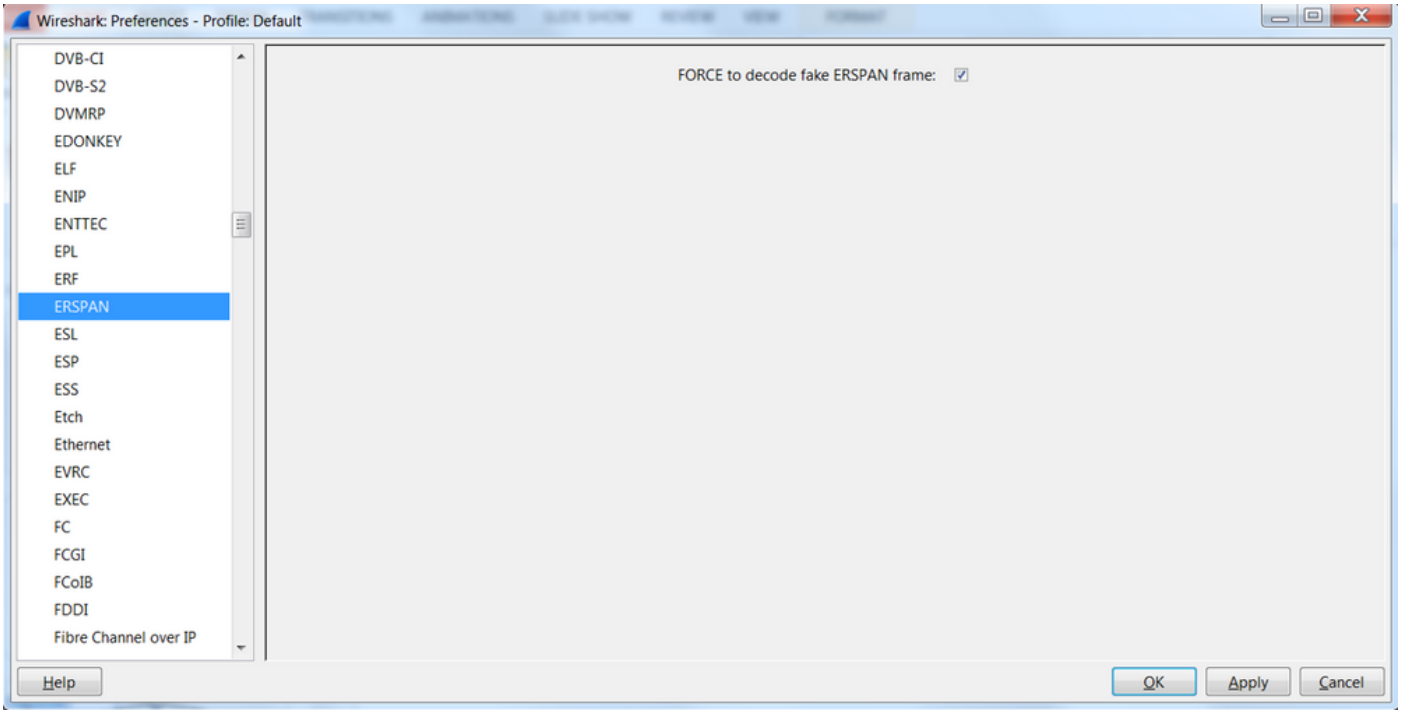
```
[root@centos3 ~]# tcpdump -xxr FabricERSPAN.pcap -c 1
reading from file FabricERSPAN.pcap, link-type EN10MB (Ethernet)
23:25:00.962224 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 53341, length 164: gre-proto-0x88be
        0x0000:  0050 56bb 3096 0022 bdf8 19ff 0800 4500
        0x0010:  00b8 0580 0000 3e2f f8de c0a8 fe65 c0a8
        0x0020:  fe01 1000 88be 0000 d05d 1002 1001 0001
        0x0030:  abcb 000c 0c0c 0c0c 0000 0000 0000 0800
        0x0040:  4500 0086 55aa 0000 1f11 b101 0a00 c05f
        0x0050:  0a00 c05c 6250 beef 0072 0000 c8a0 c007
        0x0060:  fd7f 8200 0050 56bb d95f 0050 56bb d6c2
        0x0070:  0800 4500 0054 799b 0000 4001 7bba c0a8
        0x0080:  0202 c0a8 0201 0000 4f21 b749 0027 3d24
        0x0090:  2b56 0000 0000 c720 0b00 0000 0000 1011
        0x00a0:  1213 1415 1617 1819 1a1b 1c1d 1e1f 2021
        0x00b0:  2223 2425 2627 2829 2a2b 2c2d 2e2f 3031
        0x00c0:  3233 3435 3637

ESPAN Ethernet header          : Dst 0050.56bb.3096 , Src 0022.bdf8.19.ff
ERSPAN IP header               : Dst 192.168.254.1  , Src 192.168.254.101
GRE header (= ERSPAN Type II)  : 0x88be = ERSPAN (S bit on 0x1000)
ERSPAN Type II header          : VLAN 2, ERSPAN ID 1
Ethernet header                : Dst 0022.bdf8.19ff , Src 0050.56bb.d6c2
IP header                      : Dst 10.0.192.95    , Src 10.0.192.92
UDP header                     : Dst 0xbeef(48879)  , Src 0x6250(25168)
iVXLAN header                  : sclass 0xc007 , VNID 0xfd7f82
Ethernet header                : Dst 0050.56bb.d95f , Src 0050.56bb.d6c2
IP header                      : Dst 192.168.2.254  , Src 192.168.2.2
```

如何解码ERSPAN类型I

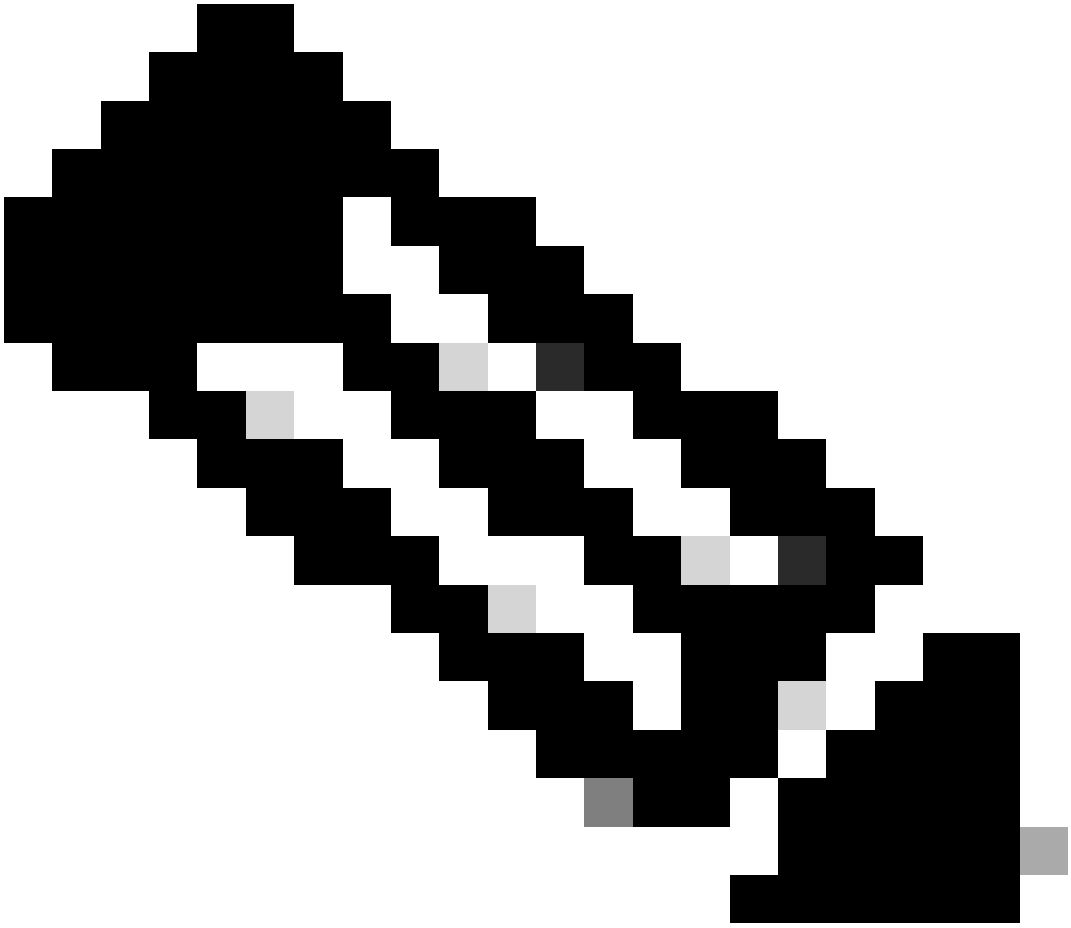
第 1 项.导航到Edit > Preference > Protocols > ERSPAN，然后选中“FORCE”以解码虚假ERSPAN帧。

- Wireshark (GUI)



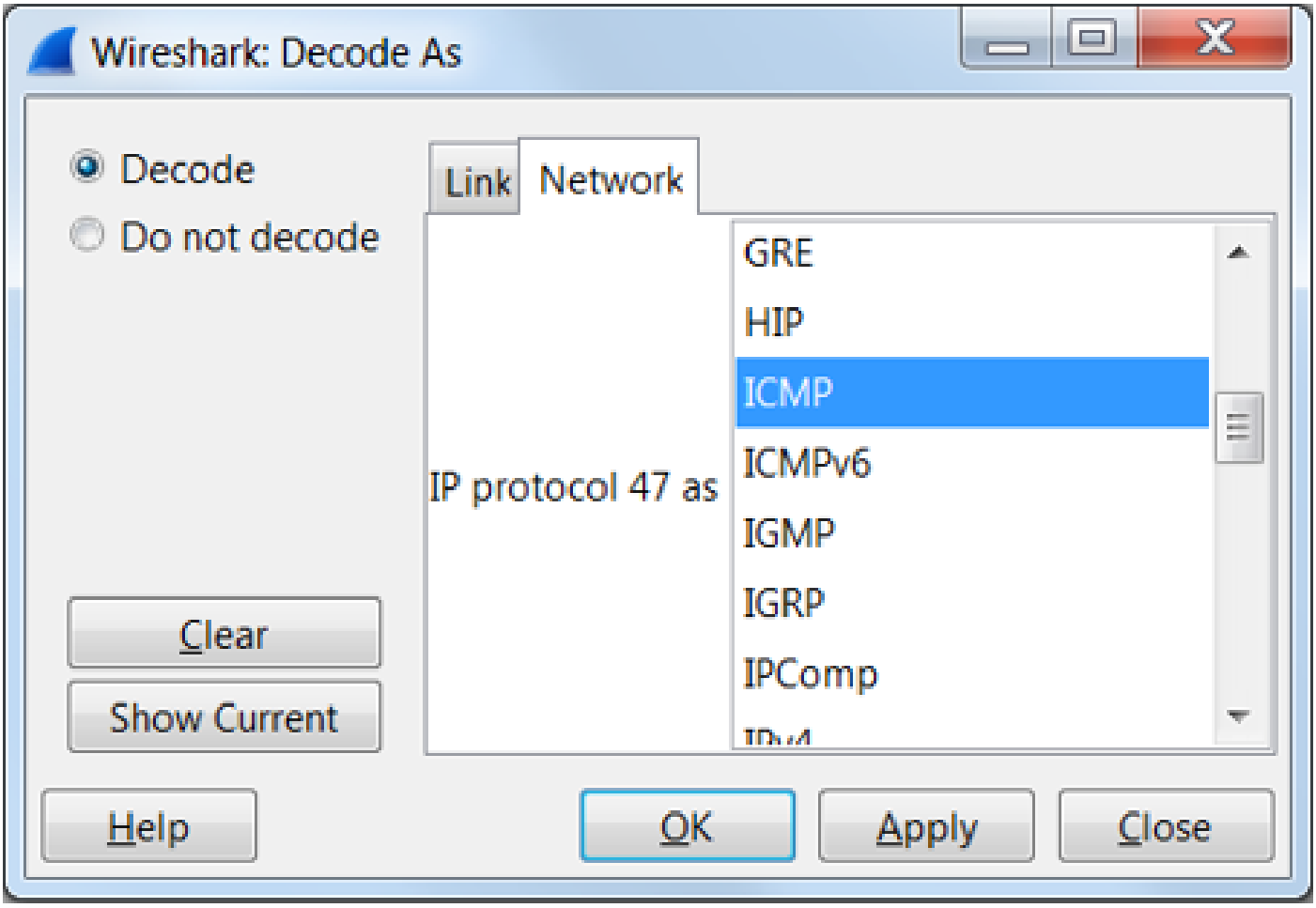
- Tshark（Wireshark的CLI版本）：

```
user1@linux# tshark -f 'proto GRE' -nV -i eth0 -o erspan.fake_erspan:true
```

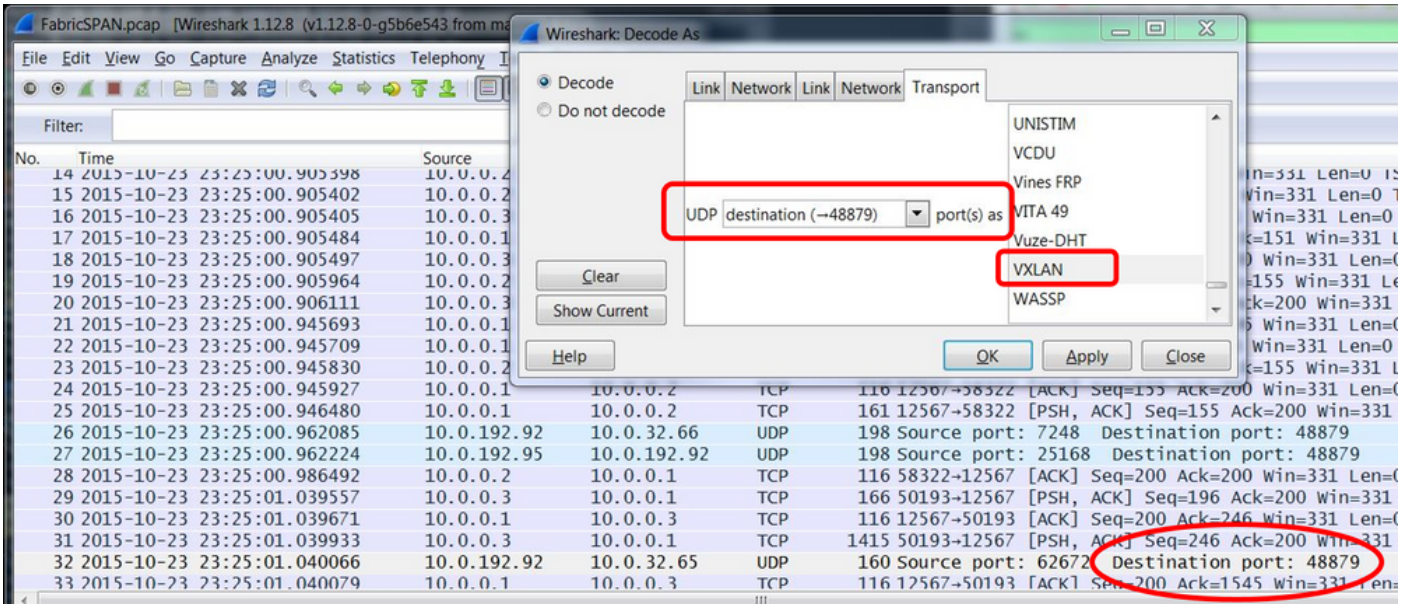


注意：请确保在阅读ERSPAN类型II或III时禁用此选项。

第 2 项.导航至 Decode As > Network > ICMP (if it’s ICMP).



如何解码iVxLAN报头

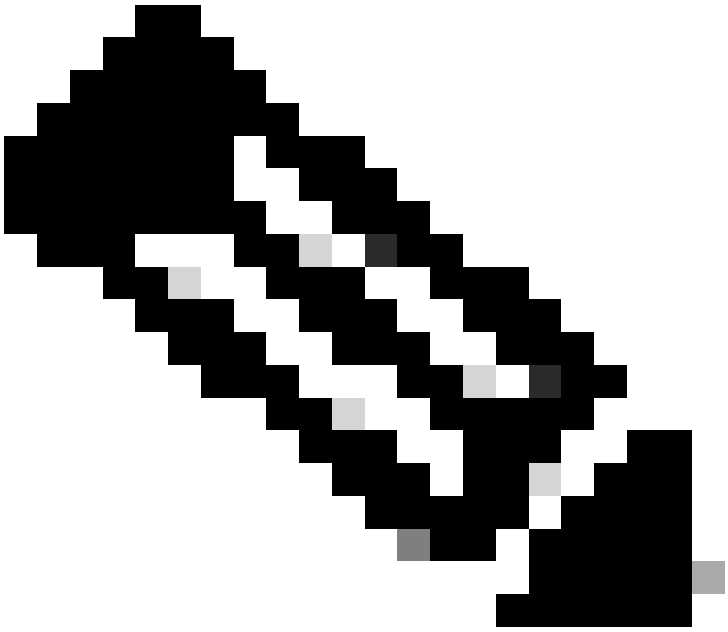


iVxLAN报头使用目的端口48879。因此，如果您在Wireshark上将UDP目标端口48879配置为VxLAN，则可以解码iVxLAN报头和VxLAN。

1. 请确保先选择iVxLAN封装的数据包。

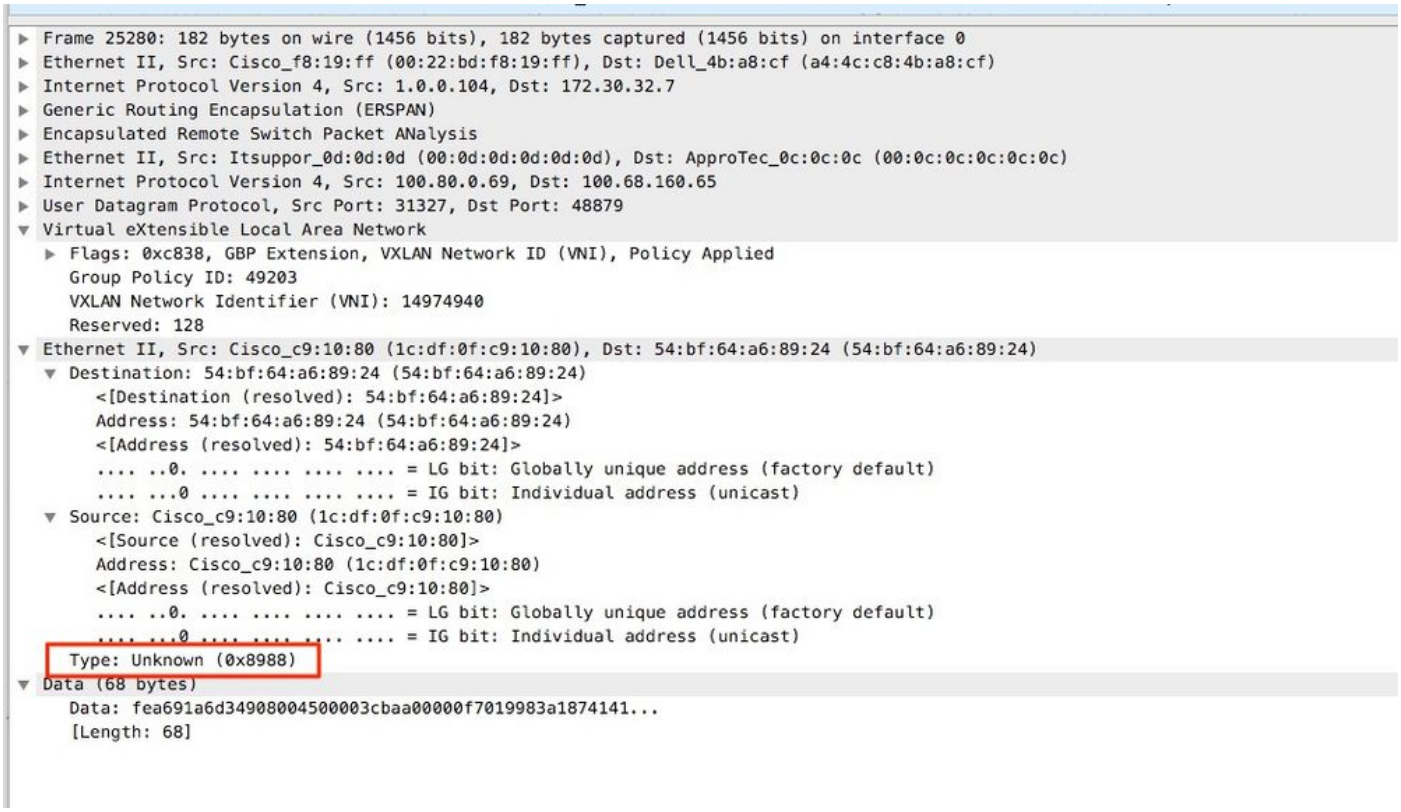
2. 导航到Analyze > Decode As > Transport > UDP destination (48879) > VxLAN。

- 然后Apply。



注意：交换矩阵端口上的APIC之间存在通信数据包。这些数据包未由iVxLAN报头封装。

当您在运行Precision时间协议(PTP)的用户网络上捕获erspan时，有时会看到Wireshark无法解释数据，因为GRE封装(0x8988)中存在未知的ethertype。0x8988是启用PTP时插入到数据平面数据包中的时间标记的ethertype。将ethertype 0x8988解码为“Cisco ttag”以显示数据包的详细信息。


Frame 25280: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits) on interface 0
Ethernet II, Src: Cisco_f8:19:ff (00:22:bd:f8:19:ff), Dst: Dell_4b:a8:cf (a4:4c:c8:4b:a8:cf)
Internet Protocol Version 4, Src: 1.0.0.104, Dst: 172.30.32.7
Generic Routing Encapsulation (ERSPAN)
Encapsulated Remote Switch Packet ANalysis
Ethernet II, Src: Itsuppor_0d:0d:0d (00:0d:0d:0d:0d:0d), Dst: ApproTec_0c:0c:0c (00:0c:0c:0c:0c:0c)
Internet Protocol Version 4, Src: 100.80.0.69, Dst: 100.68.160.65
User Datagram Protocol, Src Port: 31327, Dst Port: 48879
Virtual eXtensible Local Area Network
Flags: 0xc838, GBP Extension, VXLAN Network ID (VNI), Policy Applied
Group Policy ID: 49203
VXLAN Network Identifier (VNI): 14974940
Reserved: 128
Ethernet II, Src: Cisco_c9:10:80 (1c:df:0f:c9:10:80), Dst: 54:bf:64:a6:89:24 (54:bf:64:a6:89:24)
Destination: 54:bf:64:a6:89:24 (54:bf:64:a6:89:24)
<[Destination (resolved): 54:bf:64:a6:89:24]>
Address: 54:bf:64:a6:89:24 (54:bf:64:a6:89:24)
<[Address (resolved): 54:bf:64:a6:89:24]>
.... ..0. .... = LG bit: Globally unique address (factory default)
....0 .... = IG bit: Individual address (unicast)
Source: Cisco_c9:10:80 (1c:df:0f:c9:10:80)
<[Source (resolved): Cisco_c9:10:80]>
Address: Cisco_c9:10:80 (1c:df:0f:c9:10:80)
<[Address (resolved): Cisco_c9:10:80]>
.... ..0. .... = LG bit: Globally unique address (factory default)
....0 .... = IG bit: Individual address (unicast)
Type: Unknown (0x8988)
Data (68 bytes)
Data: fea691a6d34908004500003cbaa00000f7019983a1874141...
[Length: 68]

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）。