在ACI上配置交换端口分析器

目录

简介

背景信息

思科ACI中的SPAN类型

- 限制和准则

配置

- 接入SPAN(ERSPAN)
  - 拓扑示例
  - 配置示例
- 接入SPAN（本地）
  - 拓扑示例
  - 配置示例
- 访问SPAN — 使用ACL过滤器
- 租户SPAN(ERSPAN)
  - 拓扑示例
- 配置示例
- 交换矩阵SPAN(ERSPAN)
  - 拓扑示例
  - 配置示例
  - SPAN源组目标和源
- GUI验证

选择ACI SPAN类型

接入SPAN(ERSPAN)

- 例 1
- 案例 2
- 案例 3
- 案例 4

接入SPAN（本地SPAN）

- 例 1
- 案例 2
- 案例 3
- 案例 4
- 案例 5
- 案例 6
- 案例 7

租户SPAN(ERSPAN)

- 例 1

交换矩阵SPAN(ERSPAN)

- 例 1
- 案例 2
- 案例 3

SPAN到CPU
- 配置步骤
- 支持的平台
- 限制和限制
- 配置汇总
  - 步骤 1：在GUI中配置SPAN到CPU
  - 步骤 2：检验枝叶上的SPAN会话
  - 步骤 3：验证应用的过滤器
  - 步骤 4：检验HX上的SPAN统计信息
  - 步骤 5：在枝叶上捕获和解码SPAN流量
- 从何处查看SPAN流量
  - 捕获和解码示例
  - 将输出保存到PCAP
- 问题

您需要在SPAN目标设备上执行什么操作？
- 对于ERSPAN
- 对于本地SPAN

如何读取ERSPAN数据
- ERSPAN版本（类型）
  - ERSPAN类型I（由Broadcom Trident 2使用）
  - ERSPAN类型II或III
- ERSPAN数据示例
  - 租户SPAN/接入SPAN(ERSPAN)
  - 捕获数据包的详细信息（ERSPAN类型I）
  - 交换矩阵SPAN(ERSPAN)
  - 捕获的数据包的详细信息（ERSPAN类型II）
- 如何解码ERSPAN类型I
- 如何解码iVxLAN报头

简介

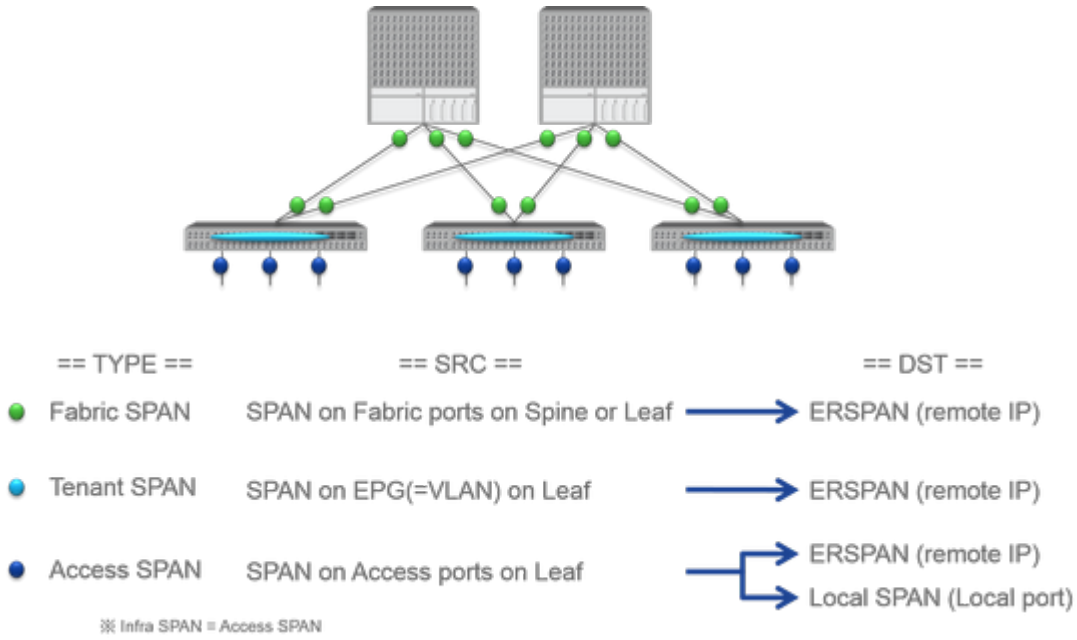
本文档介绍如何在思科以应用为中心的基础设施(ACI)上配置交换端口分析器(SPAN)。

背景信息

一般来说，SPAN分为三种类型。本地SPAN、远程SPAN(RSPAN)和封装远程SPAN(ERSPAN)。这些SPAN之间的差异主要是复制数据包的目的地。Cisco ACI支持本地SPAN和ERSPAN。

注意：本文档假设读者一般已经熟悉SPAN，例如本地SPAN和ERSPAN差异。

思科ACI中的SPAN类型



Cisco ACI有三种类型的SPANFabric 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，

- 交换矩阵SPAN捕获从枝叶和主干交换机之间的接口传入和传出的数据包。
- 访问SPAN捕获从枝叶交换机和外部设备之间的接口传入和传出的数据包。
- 租户SPAN用于捕获ACI枝叶交换机上的终端组(EPG)传入和传出的数据包。

此SPAN名称对应于Cisco ACI GUI上的配置位置。

- 交换矩阵SPAN在交换矩阵>交换矩阵策略下配置。
- 访问SPAN在Fabric > Access Policies下配置。
- 租户SPAN在Tenants > {each tenant}下配置。

对于每个SPAN的目标，只有接入SPAN能够同时支持本地SPAN和ERSPAN。其他两个SPAN（交换矩阵和租户）仅支持ERSPAN。

限制和准则

请查看思科APIC故障排除指南中的限制和指南。在故障排除工具和方法>使用SPAN中提到了该工

具。

配置

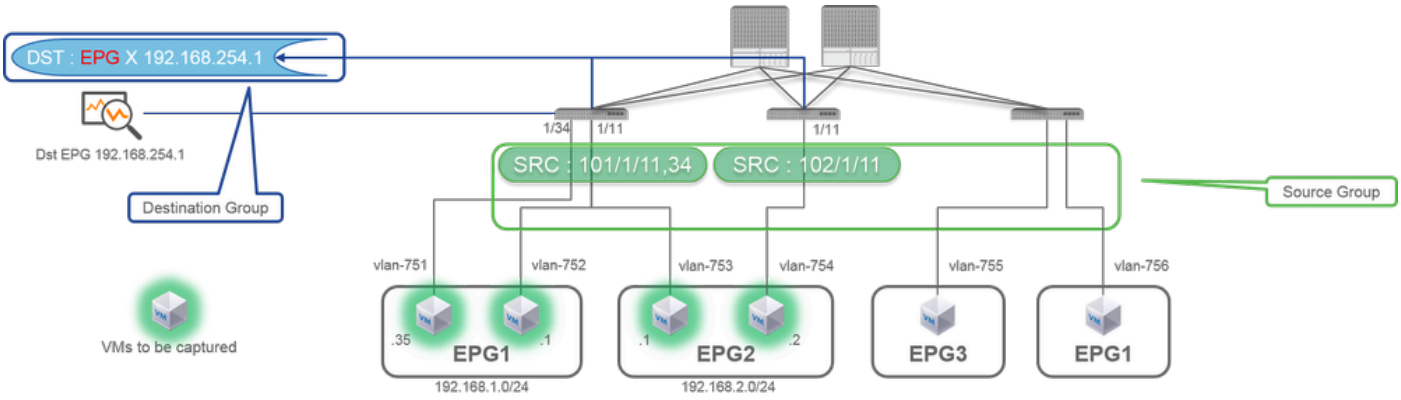
本节介绍与每个SPAN类型的配置相关的简短示例。在后面的部分中，有有关如何选择范围类型的特定示例案例。

思科APIC故障排除指南中也介绍了SPAN配置：Troubleshooting Tools and methodology > Using SPAN。

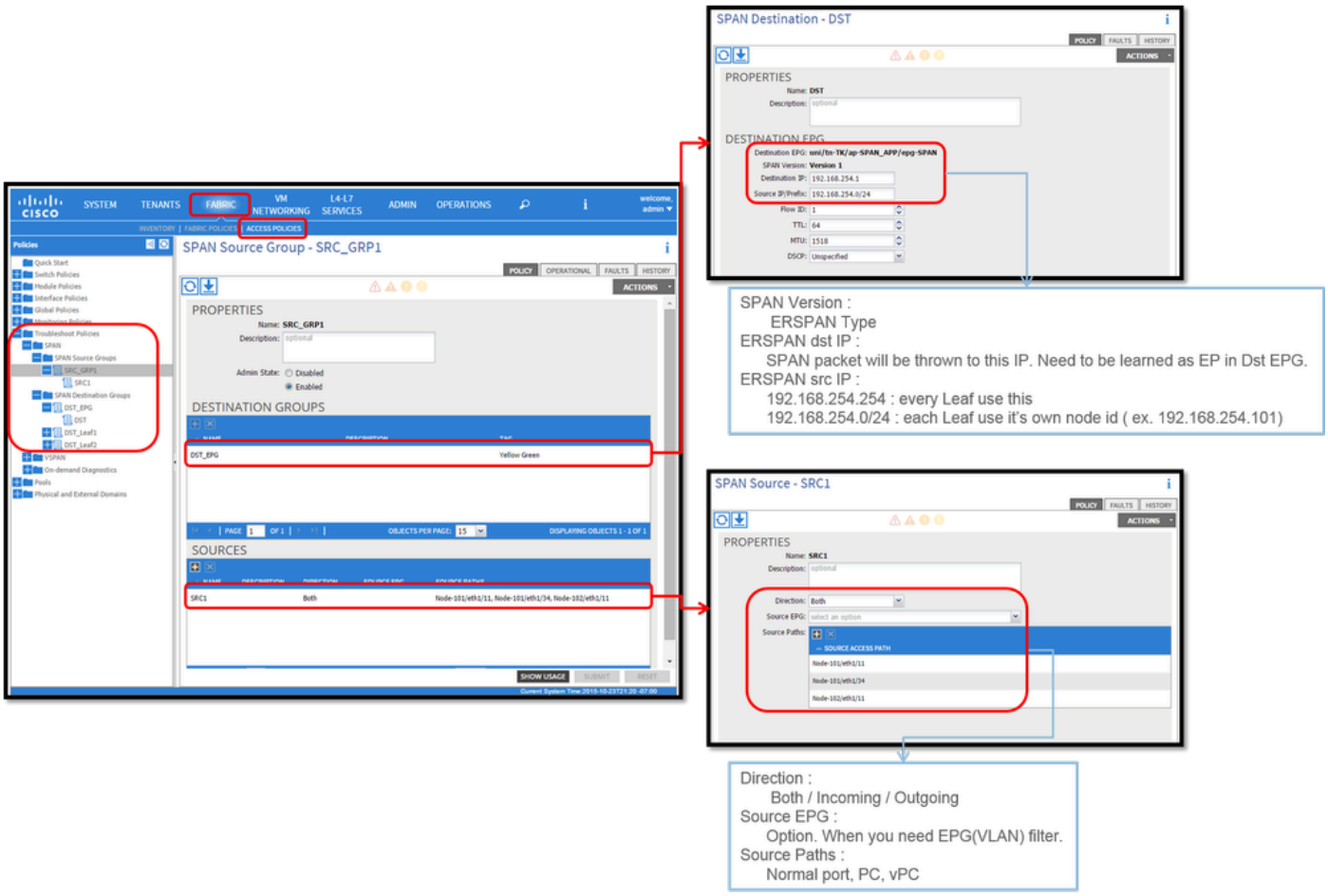
用户界面可能与当前版本不同，但配置方法相同。

接入SPAN(ERSPAN)

拓扑示例



配置示例



其中：

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

- SPAN源组
- SPAN目标组

SPAN Source Group 领带Destination和Sources。

方式：

1. 创建SPAN源组(SRC_GRP1)。
2. 在SPAN Source Group(SRC_GRP1)下创建SPAN Source(SRC1)。
3. 为SPAN源(SRC1)配置这些参数。
   — 方向 — 源EPG（选项）
   — 源路径（可以是多个接口）



注：请参阅图片了解每个参数的详细信息。

4. 创建SPAN目标组(DST_EPG)。
5. 创建SPAN目标(DST)。
6. 为SPAN目标(DST)配置这些参数
   — 目标EPG
   -目的 IP
   — 源IP/前缀(这可以是任何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。

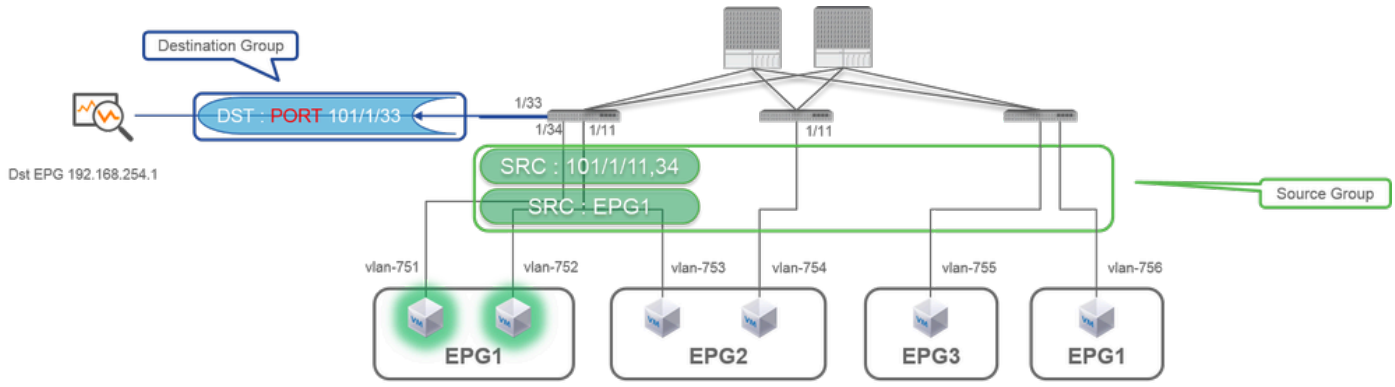
7. 确保SPAN Destination Group绑定到相应的SPAN Source Group。
8. 确保Admin State已启用。

注意：当您在此管理状态上选择Disabled时，SPAN停止。如果以后重复使用所有策略，则无需将其删除。

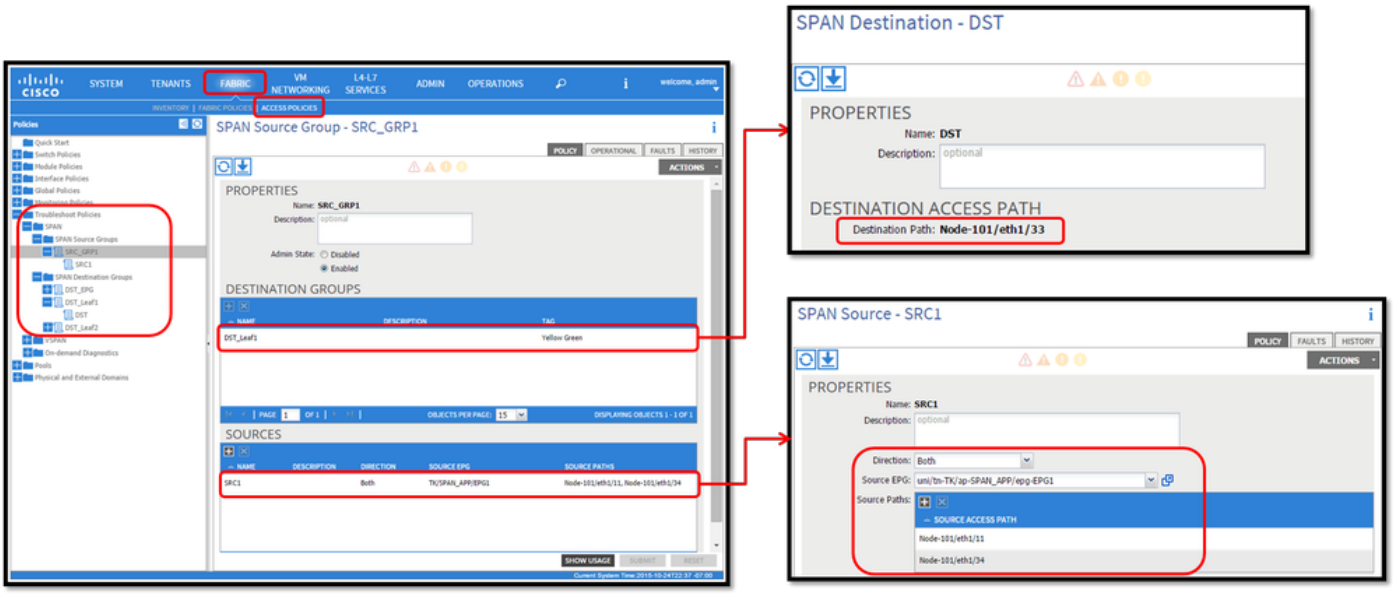
另请确保将ERSPAN的目标IP作为指定目标EPG下的终端获取。在上述示例中，必须在租户TK >应用配置文件SPAN_APP > EPG SPAN下获取192.168.254.1。或者，如果目标设备是静默主机，则目标IP可配置为此EPG下的静态终端。

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源组(SRC_GRP1)
2. 在SPAN Source Group(SRC_GRP1)下创建SPAN Source(SRC1)
3. 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。
4. 创建SPAN目标组(DST_Leaf1)
5. 创建SPAN目标(DST)
6. 为SPAN Destination(DST)配置这些参数
   — 目标接口和节点。
7. 确保SPAN Destination Group绑定到相应的SPAN Source Group。

8. 确保Admin State已启用。
   ※选择Disabled on this Admin State时，SPAN停止。如果以后重复使用所有策略，则无需将

其删除。

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枝叶上的接口时，该功能会发挥作用。

限制:

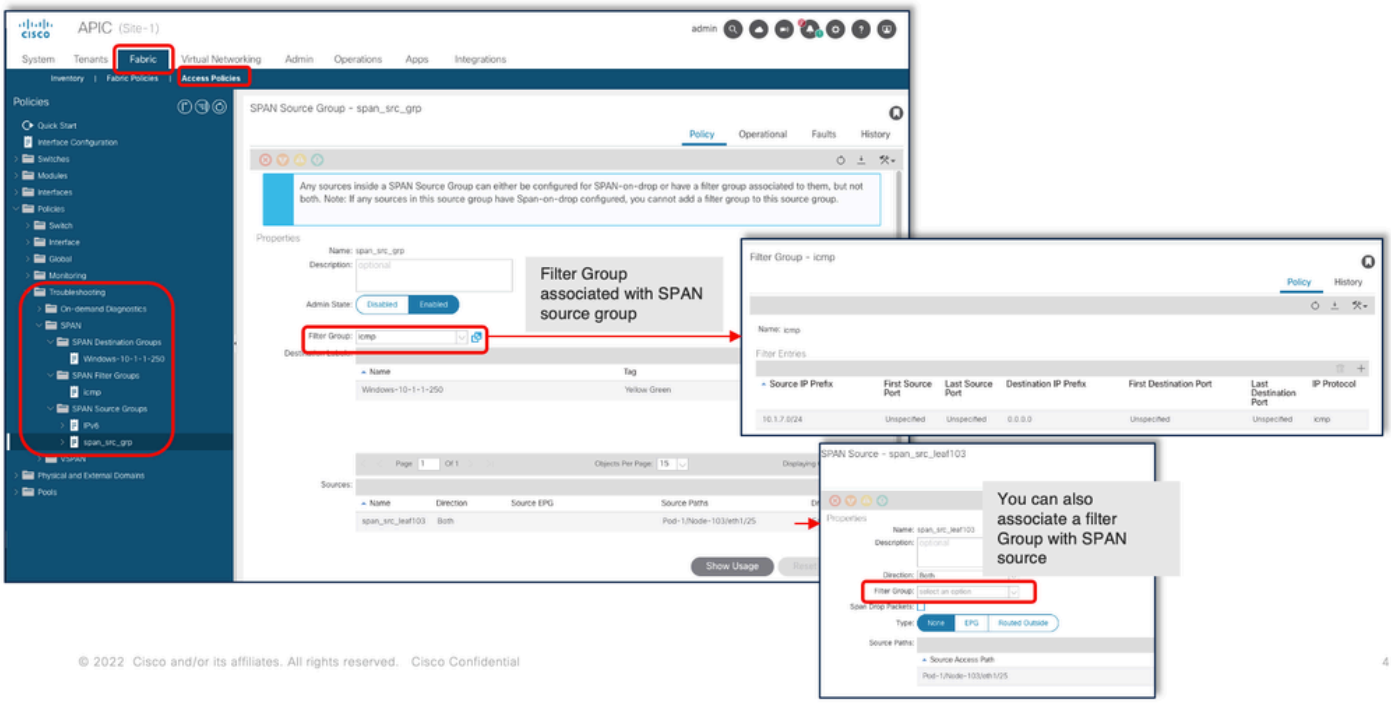
- 对于本地SPAN，必须在同一枝叶上配置目标接口和源接口。
- 目的接口只要处于UP状态，就不需要它位于EPG上。
- 当虚拟端口通道(vPC)接口指定为源端口时，无法使用本地SPAN
  但是，有一个解决方法。在第一代枝叶交换机上，作为vPC或PC成员的单个物理端口可以配置为SPAN源。使用此本地SPAN可用于vPC端口上的流量。
  但是，此选项在第二代枝叶上不可用(Cisco Bug ID CSCvc11053)。 相反，通过Cisco bug ID CSCvc44643在2.1(2e)、2.2(2e)和转发版本中添加了对“VPC组件PC”上SPAN的支持。这样，任何层代枝叶都可以将作为vPC成员的端口通道配置为SPAN源。这允许任何层代枝叶对vPC端口上的流量使用本地SPAN。
- 指定第二代枝叶上端口通道的各个端口只会生成数据包的子集（也由于Cisco Bug ID CSCvc11053）。
- PC和vPC不能用作本地SPAN的目标端口。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

您可以在访问范围源上使用ACL过滤器。此功能可将SPAN特定流量或流量传入/传出SPAN源。
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/源中不支持此功能。
在过滤器组中添加过滤器条目时必须小心，因为它可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）用于过滤此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。

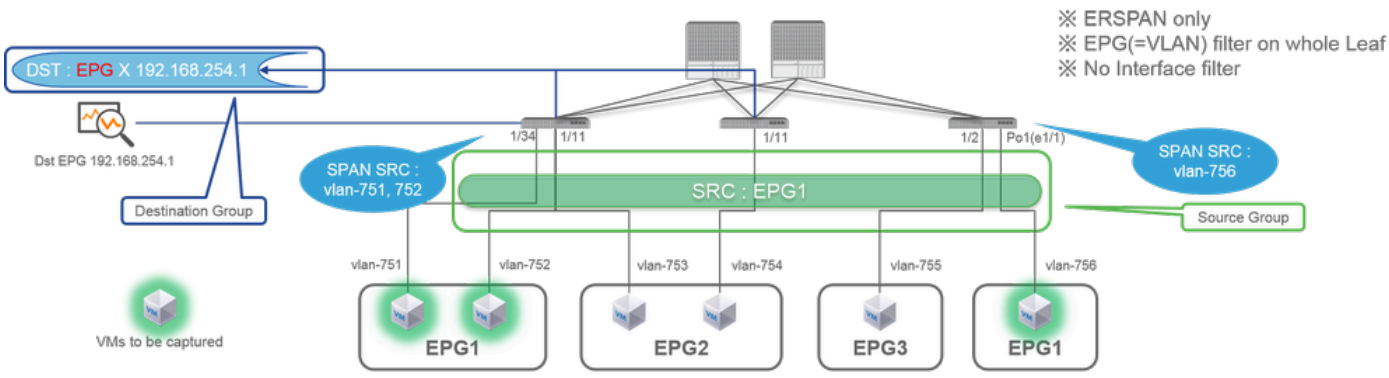


如果特定Span源已经与过滤器组（例如y）关联，则使用该过滤器组(y)来过滤此特定Span源下所有接口上的组

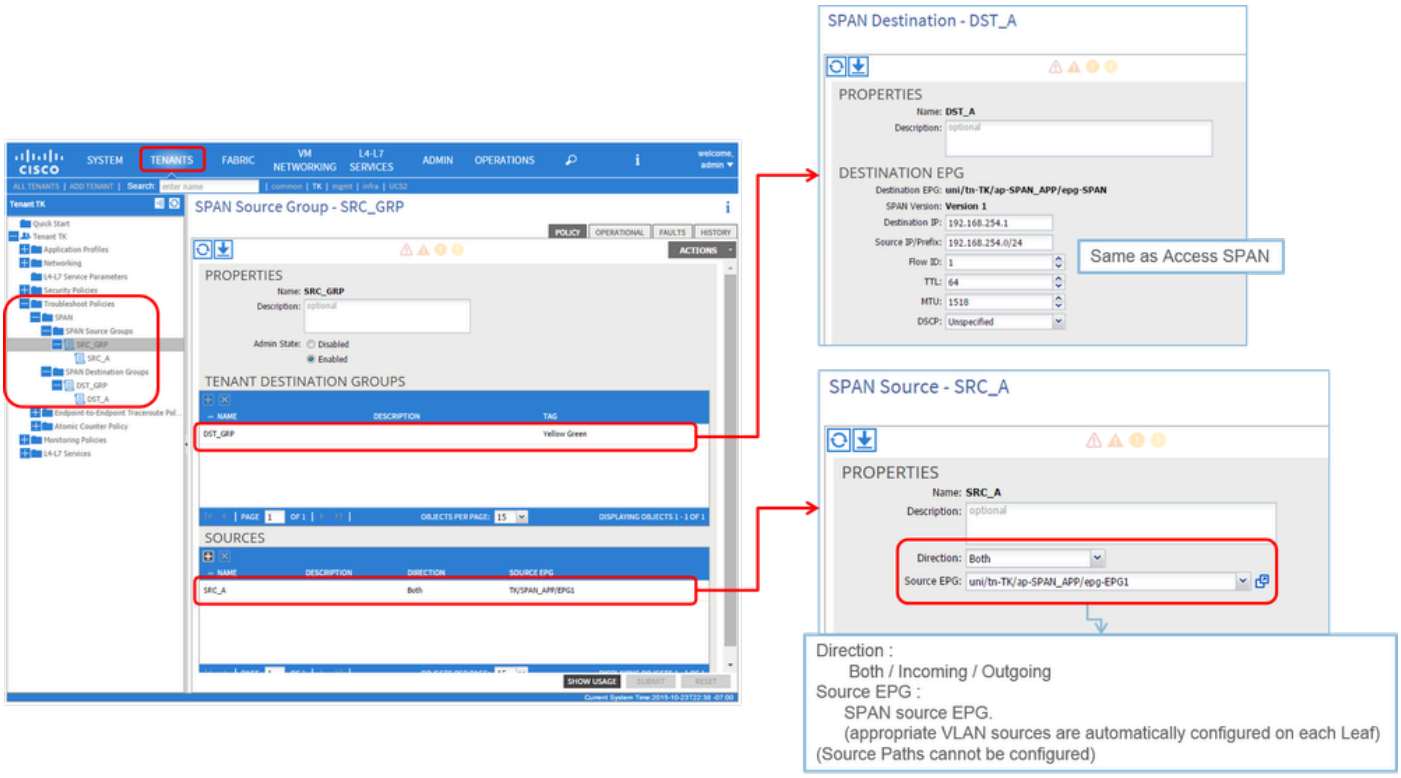
— 在源组应用的过滤器组自动应用于该源组中的所有源。
— 在源应用的过滤器组仅适用于该源。
— 过滤器组同时应用于源组以及该源组中的源，则应用于源的过滤器组优先。
— 删除在源上应用的过滤器组，自动应用父源组中应用的过滤器组。
— 删除应用于源组的过滤器组，即会从当前继承到该源组的所有源中删除该过滤器组。

租户SPAN(ERSPAN)

拓扑示例



配置示例



- 其中：

租户> {tenant name} >故障排除策略> SPAN

- SPAN源组

- SPAN目标组

※ SPAN源组关Destination系 Sources.

- 方式：

1. 创建SPAN源组(SRC_GRP)
2. 在SPAN Source Group(SRC_GRP)下创建SPAN源(SRC_A)
3. 为SPAN源(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。
4. 创建SPAN目标组(DST_GRP)
5. 创建SPAN目标(DST_A)
6. 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

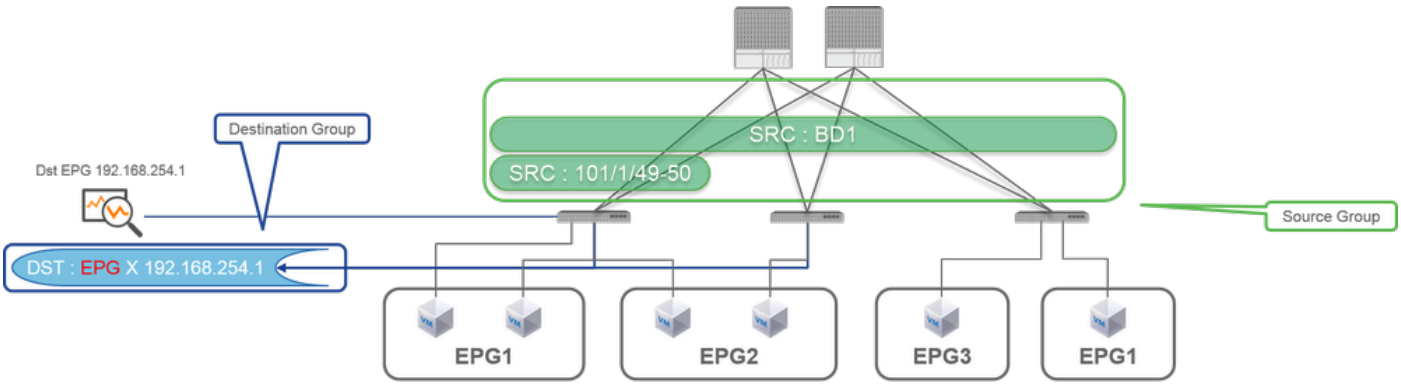
※请参阅图片了解每个参数的详细信息。

7. 确保SPAN Destination Group绑定到相应的SPAN Source Group。
8. 确保Admin State已启用。

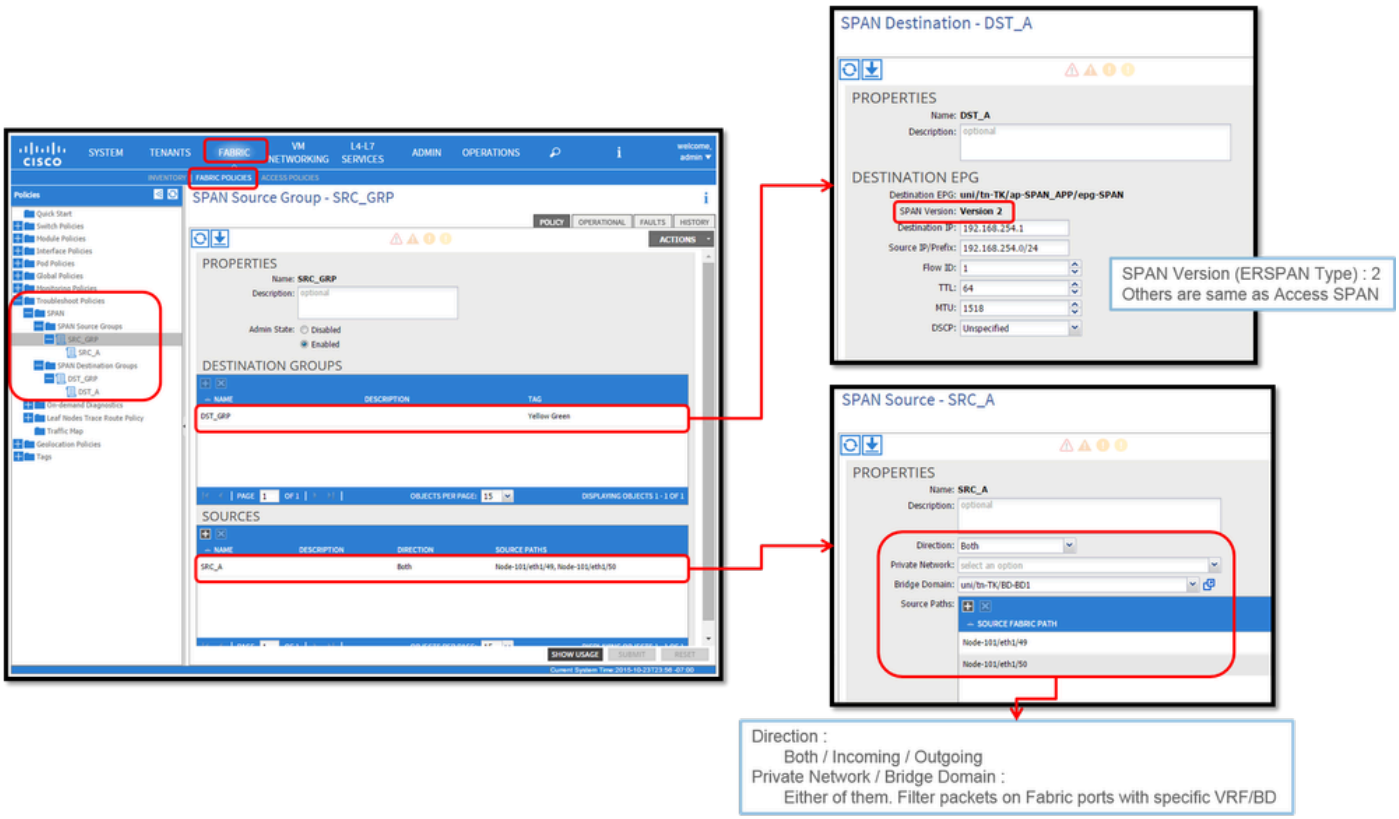
※选择Disabled on this Admin State时，SPAN停止。如果以后重复使用所有策略，则无需将其删除。

交换矩阵SPAN(ERSPAN)

拓扑示例



配置示例



- 其中：

交换矩阵>交换矩阵策略>故障排除策略> SPAN

— 交换矩阵

- SPAN目标组

SPAN源组将目标与源关联

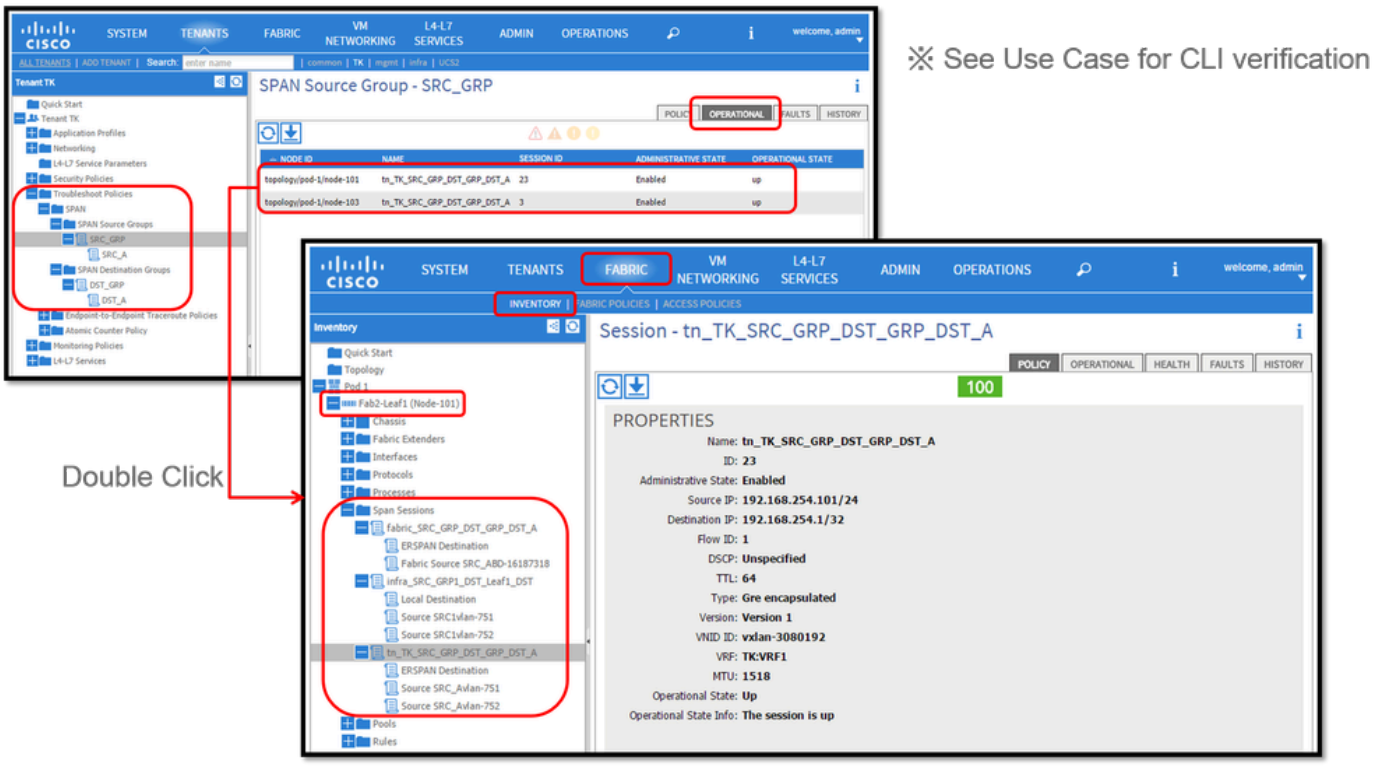
- 方式：

1. 创建SPAN源组(SRC_GRP)
2. 在SPAN Source Group(SRC_GRP)下创建SPAN Source(SRC_A)
3. 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。
4. 创建SPAN目标组(DST_GRP)

5. 创建SPAN目标(DST_A)
6. 为SPAN目标(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。
7. 确保SPAN Destination Group绑定到相应的SPAN Source Group。
8. 确保Admin State已启用。
   ※选择在此管理状态上禁用(Disabled)时，SPAN将停止。如果以后重复使用所有策略，则无需将其删除。

尽管后面的ERSPAN版本（类型）部分对此进行了说明，但您可以知道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 > Operations选项卡
2. 交换矩阵>交换矩阵策略>故障排除策略> SPAN > SPAN源组>操作选项卡
3. Tenants > {tenant name} > Troubleshoot Policies > SPAN > SPAN Source Groups > Operational选项卡

请确保运行状态为up。

- 在节点自身上验证SPAN会话

1. 从SPAN配置策略或Fabric > INVENTORY > Node > Span Sessions > { SPAN session name }

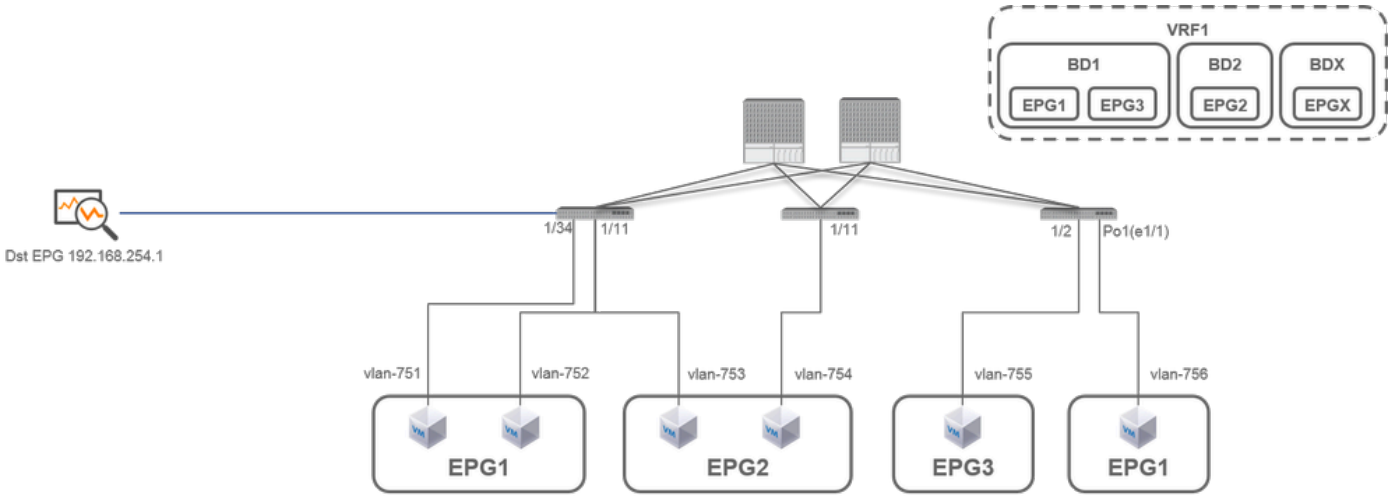
请确保运行状态为up。

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



在本部分中，详细描述了每个ACI SPAN类型(Access、Tenant、Fabric)的场景。上一部分中介绍了每个场景的基本拓扑。

如果您了解这些场景，则可以根据您的要求选择适当的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，而不考虑必须捕获接口等等。

在思科ACI中，SPAN配置了源组和目标组。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。

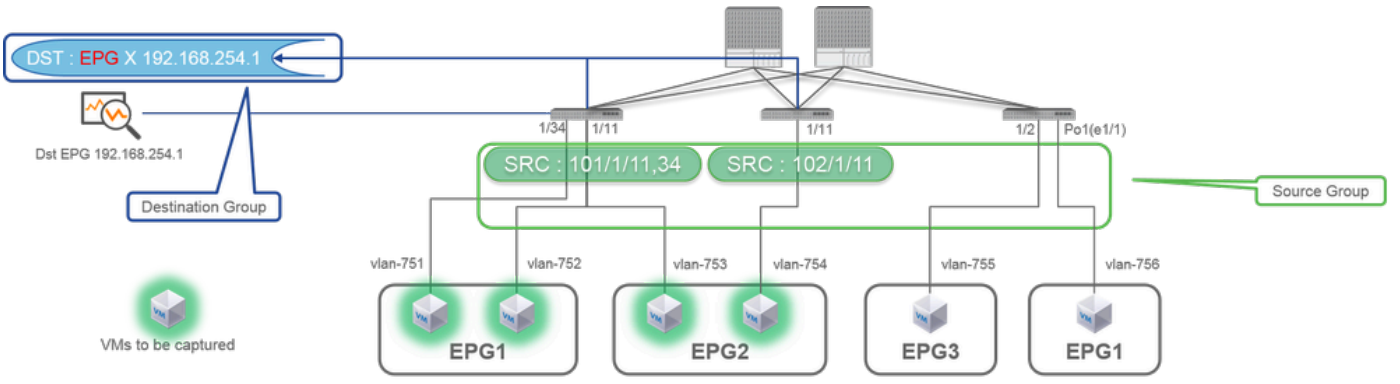
捕获数据包后，请参阅“如何读取SPAN数据”部分以解码捕获的数据包。

注意：请关注在每个拓扑中以绿色指示灯突出显示的虚拟机。每种场景都是从这些突出显示的虚拟机捕获数据包。

接入SPAN(ERSPAN)

例 1

源“Leaf1 e1/11 e1/34和Leaf2 e1/11” | Dst "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
```

- 源组
  - Leaf1 e1/11
  - 枝叶1 e1/34
  - 枝叶2 e1/11
- 目标组
  - EPG X上的192.168.254.1

访问SPAN可以为单个SPAN会话指定多个接口。它可以捕获所有进出指定接口的数据包，而不管其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。

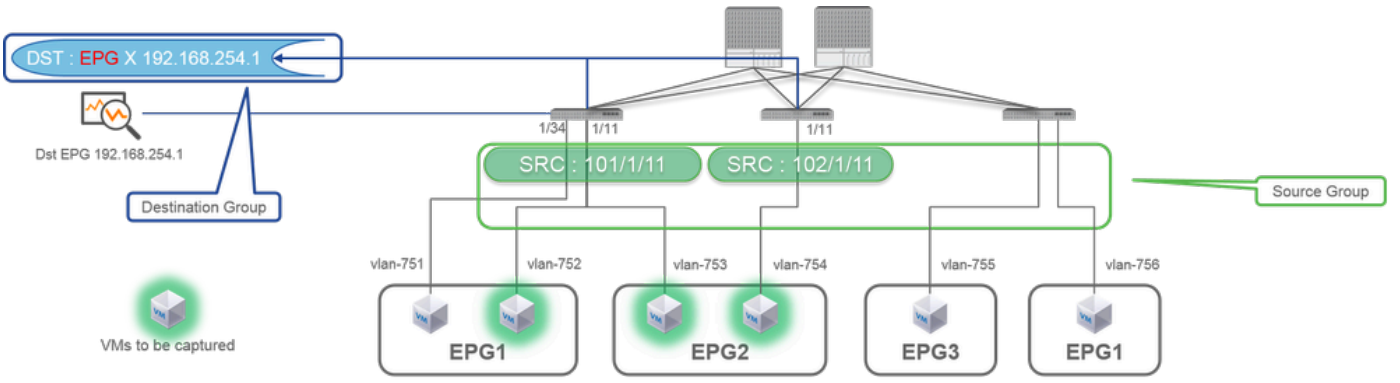
在本示例中，它会从EPG1和EPG2上的所有VM复制数据包。

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”｜Dst "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
  - Leaf1 e1/11
  - 枝叶2 e1/11
- 目标组
  - EPG X上的192.168.254.1

在本例中，Leaf1 e1/34从之前的Case1上配置的SPAN源组中删除。

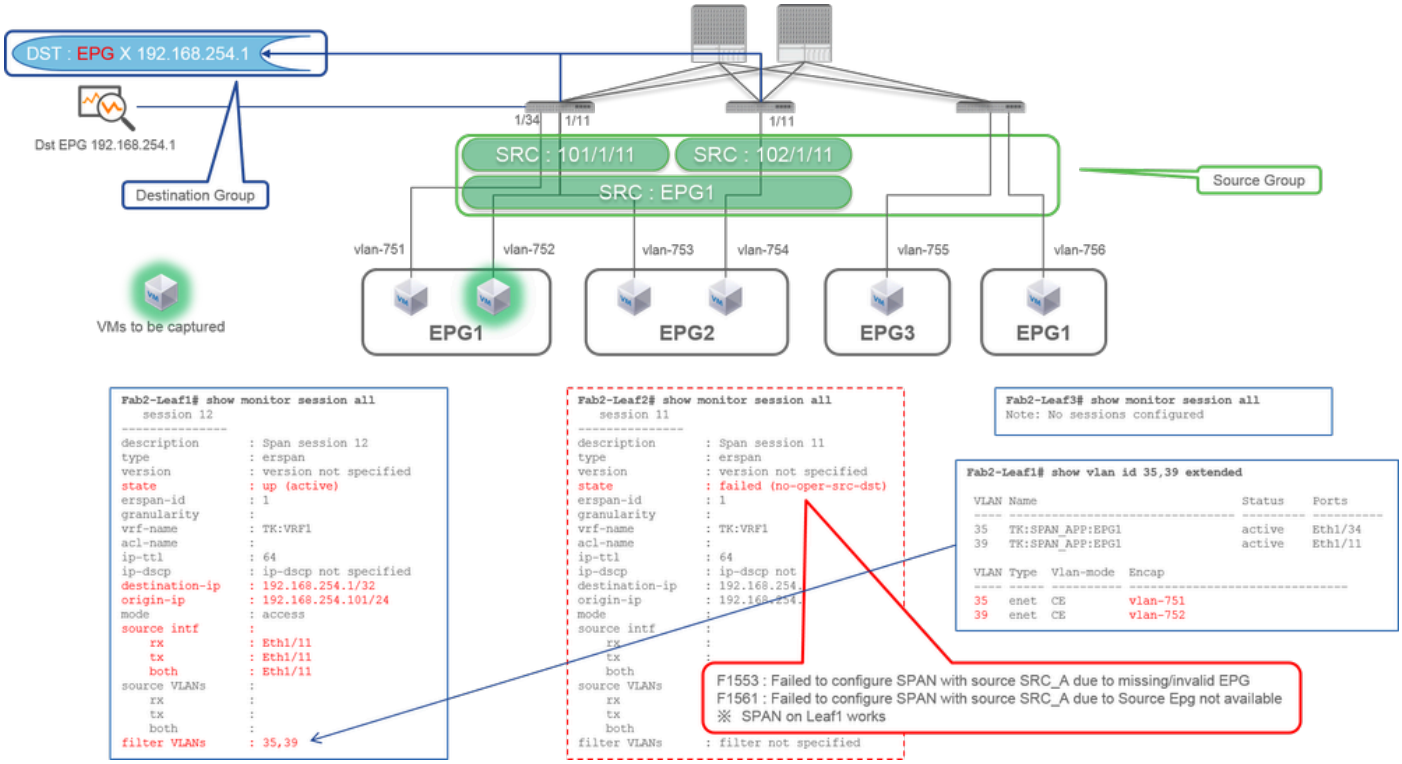
此示例的关键点是，Access SPAN可以指定源接口，而无论EPG如何。

CLI检查点：

- leaf1上的源接口从“Eth1/11 Eth1/34”更改为“Eth1/11”

案例 3

源“Leaf1 e1/11和Leaf2 e1/11和EPG1过滤器”｜Dst "192.168.254.1"



- 源组
  - Leaf1 e1/11
  - 枝叶2 e1/11
  - 过滤器EPG1
- 目标组
  - EPG X上的192.168.254.1

此示例显示访问SPAN还可以在源端口上指定特定EPG。当多个EPG在单个接口上流动时，此选项非常有用，并且只需要捕获该接口上用于EPG1的流量。

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仍然正常工作。

此外，由于EPG1在Leaf1上使用两个VLAN(VLAN-751、752)，因此会自动为Leaf1上的SPAN会话添加两个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命令显示实际封装VLAN ID和PI-VLAN的映射。

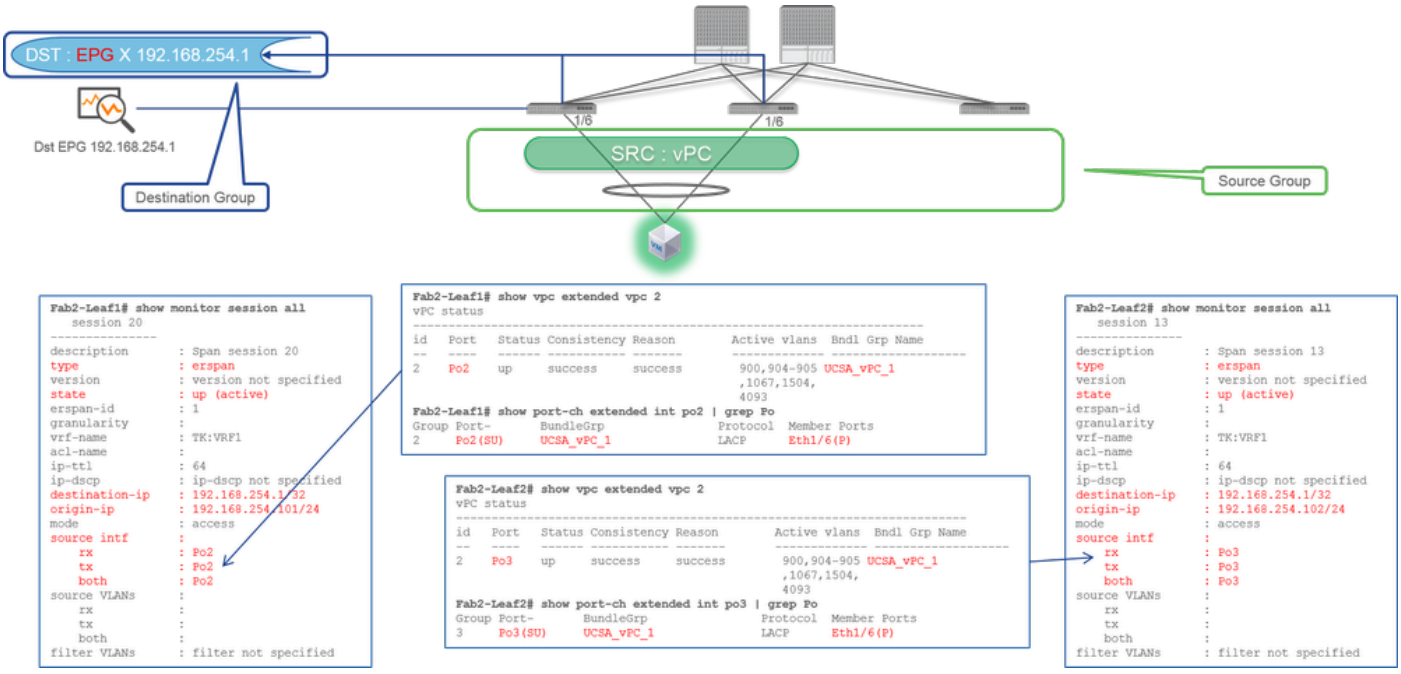
此SPAN会话允许我们仅捕获枝叶1 e1/11上EPG1(VLAN-752)的数据包，即使EPG2(VLAN-753)在同一接口上流动。

CLI检查点：

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

案例 4

源“Leaf1-Leaf2 vPC” | Dst "192.168.254.1"



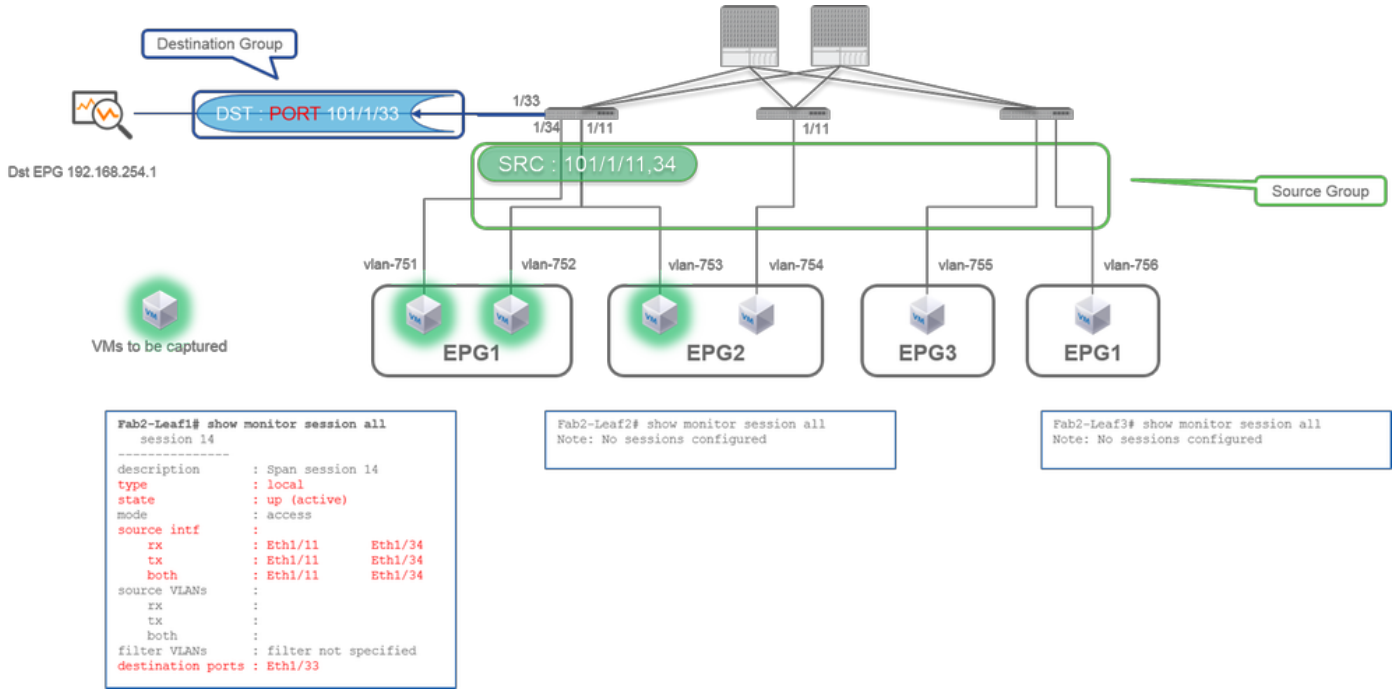
- 源组
  - 枝叶1 - 2e1/11
- 目标组
  - EPG X上的192.168.254.1

将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” | Dst "Leaf1 e1/33"



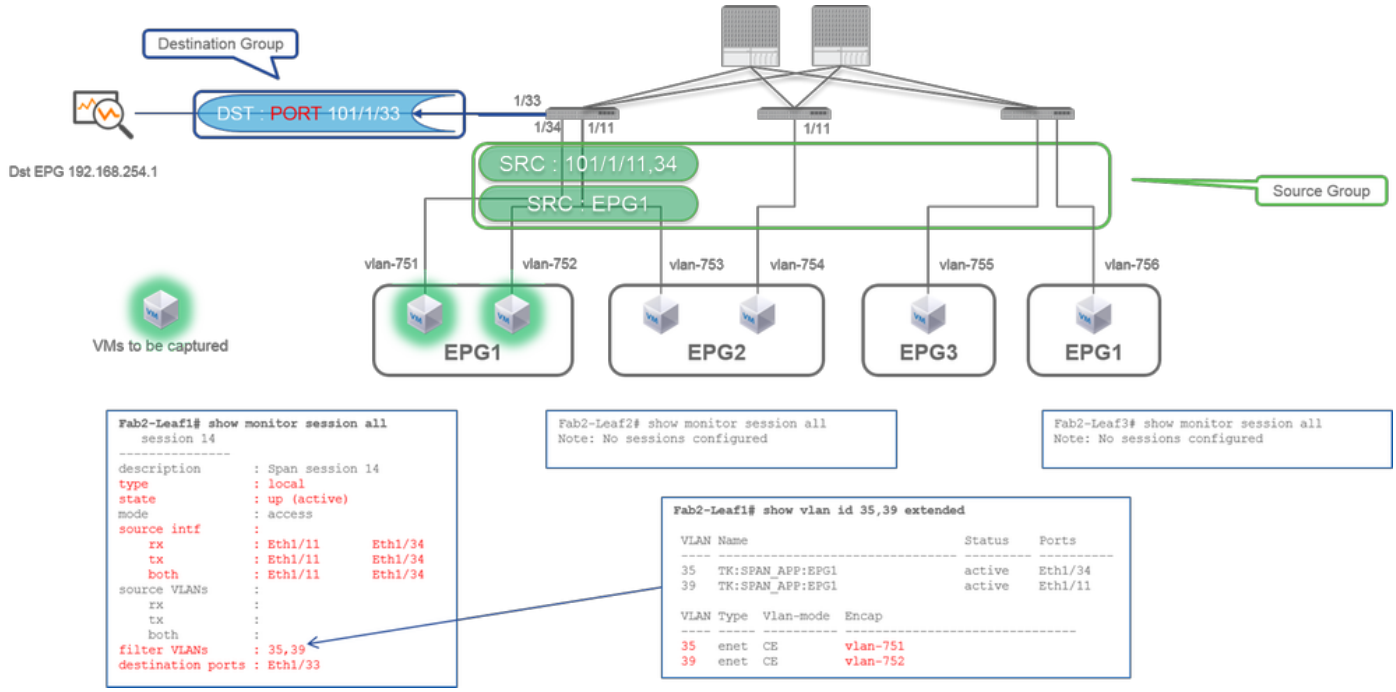
- 源组
  - Leaf1 e1/11
  - 枝叶1 e1/34
- 目标组
  - 枝叶1 e1/33

访问SPAN还可以使用本地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

源“Leaf1 e1/11 e1/34和EPG1过滤器” | Dst " Leaf1 e1/33"

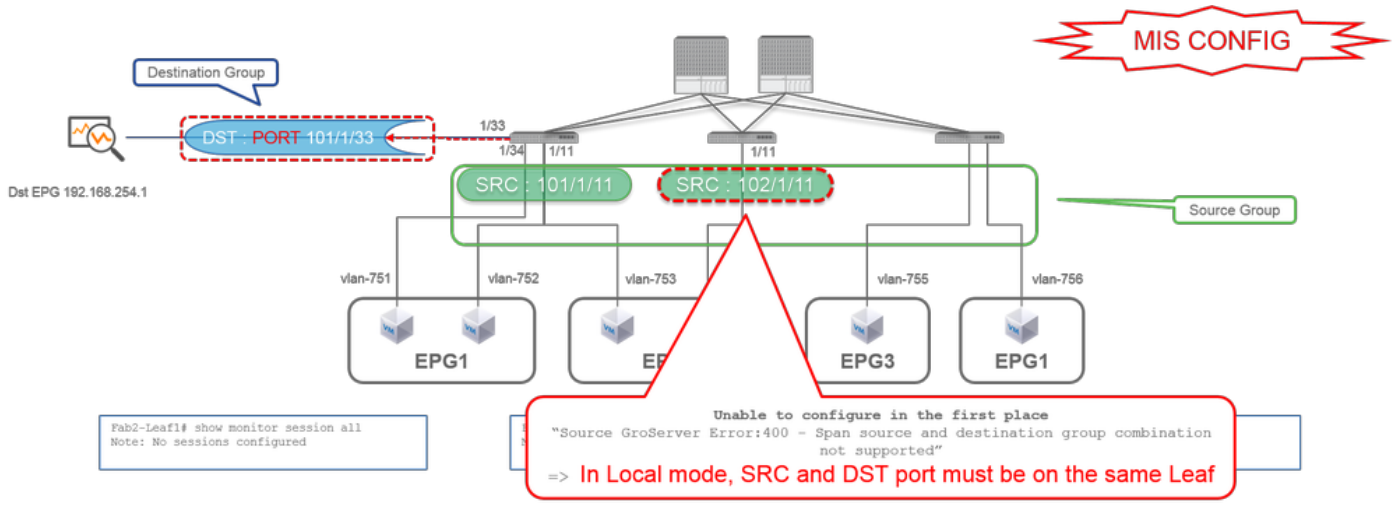


- 源组
  - Leaf1 e1/11
  - 枝叶1 e1/34
  - EPG1过滤器
- 目标组
  - 枝叶1 e1/33

通过本地SPAN访问SPAN还可以使用EPG过滤器和ERSPAN。

案例 3

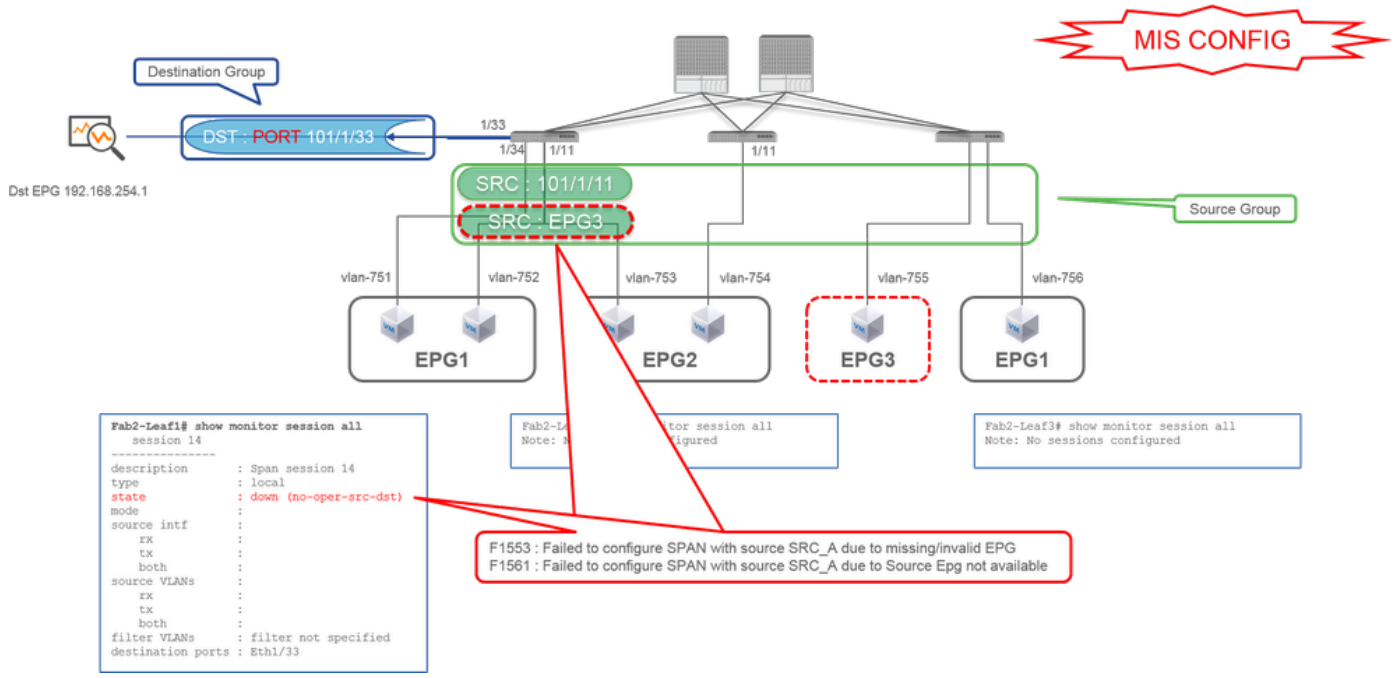
源“Leaf1 e1/11和Leaf2 e/11”｜Dst "Leaf1 e1/33"（错误案例）



- 源组
  - Leaf1 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）

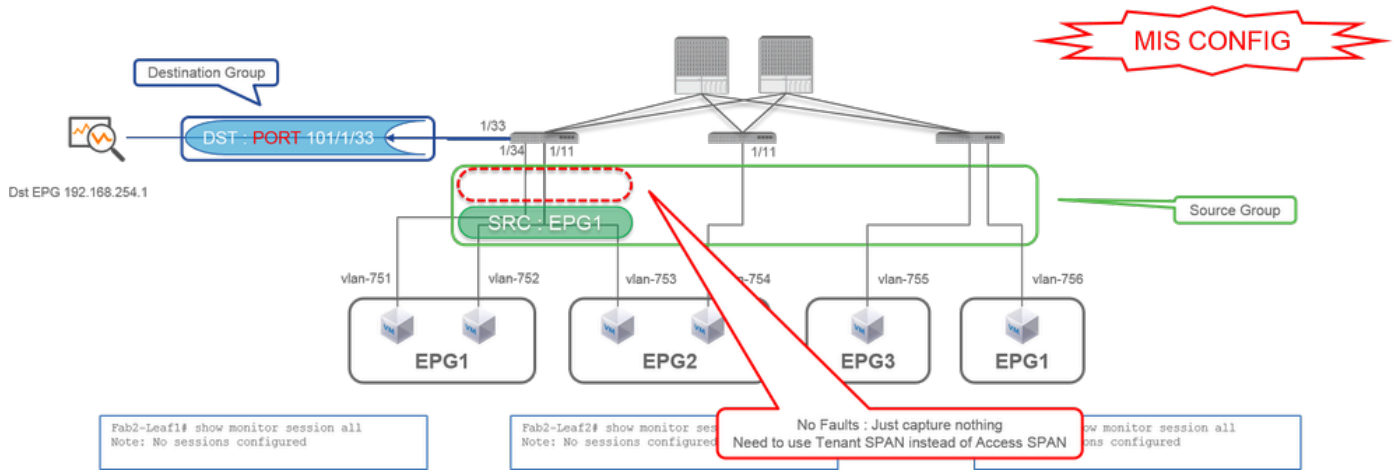


- 源组
  - Leaf1 e1/11
  - EPG3过滤器
- 目标组
  - 枝叶1 e1/33

这类似于访问SPAN(ERSPAN)上的情况3，但在本示例中，由于EPG3在Leaf1上不存在，Leaf1上的唯一SPAN会话失败。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）

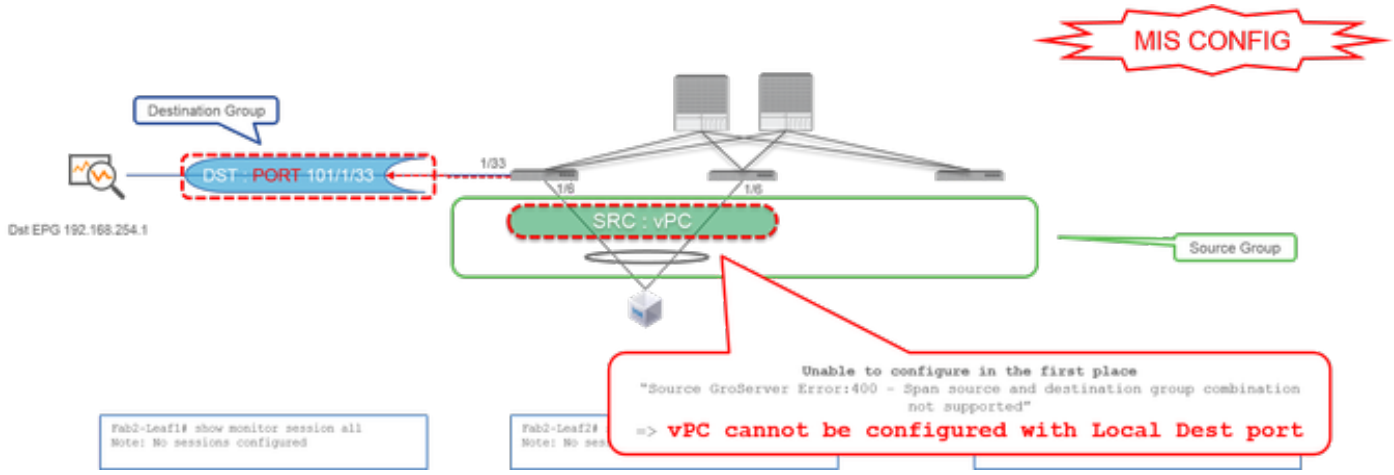


- 源组
  - EPG1过滤器
- 目标组
  - 枝叶1 e1/33

只有配置了源端口时，接入SPAN上的EPG过滤器才能工作。如果EPG是唯一要指定的源，则必须使用租户SPAN而非访问SPAN。

案例 6

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



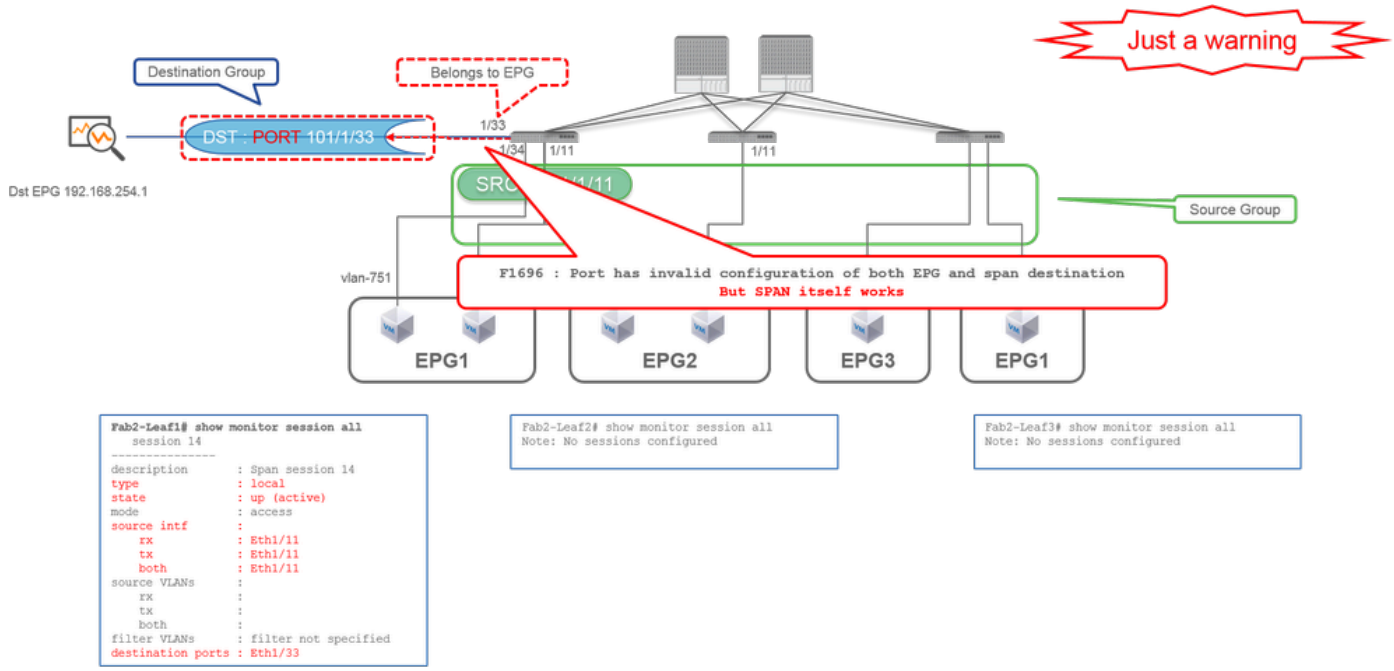
- 源组
  - 枝叶1-2 vPC
- 目标组
  - 枝叶1 e1/33

无法将vPC接口配置为具有本地SPAN的源。请使用ERSPAN。有关访问SPAN(ERSPAN)，请参阅

案例4。

案例 7

源“Leaf1 e1/11 ｜Dst "Leaf1 e1/33 & e1/33属于EPG"（工作有故障）



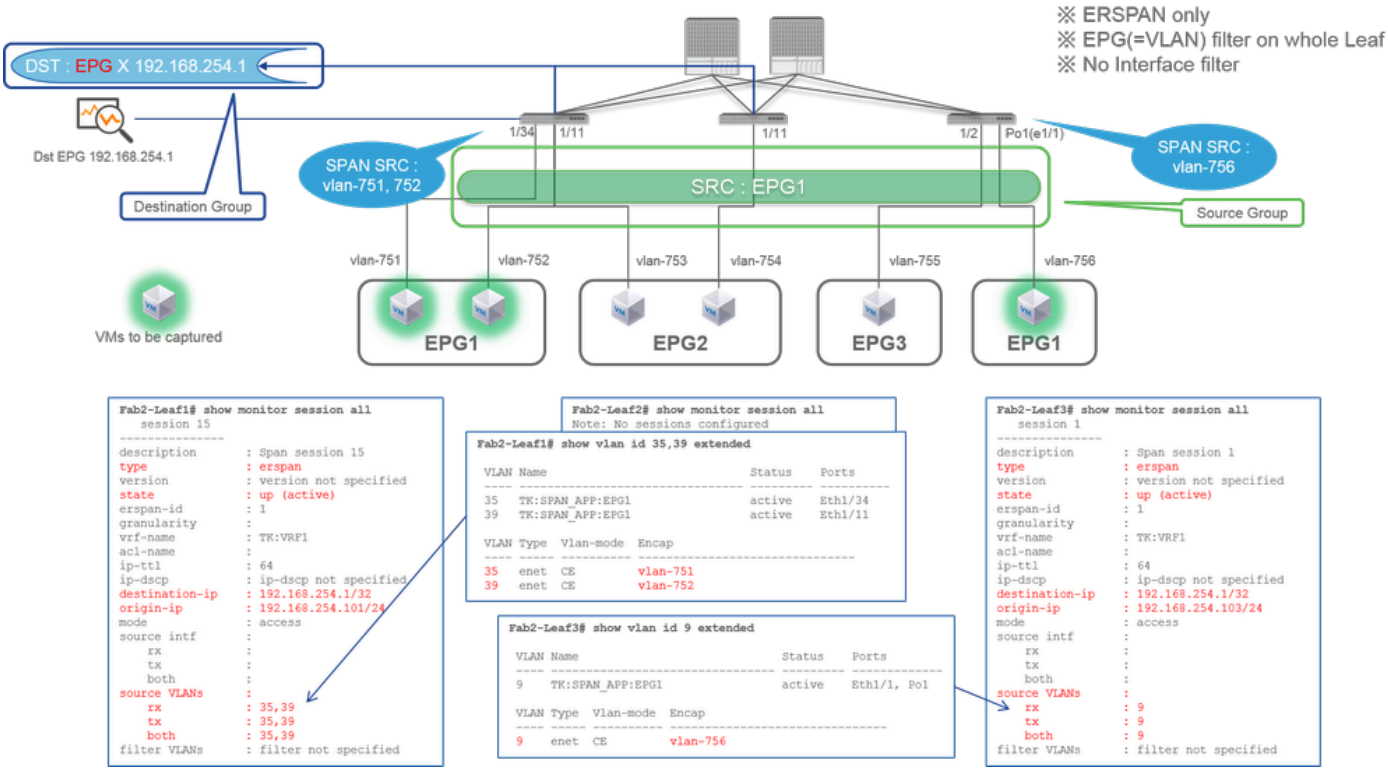
如果SPAN的目标I/F已经属于EPG，则出现故障“F1696 :端口的EPG和SPAN目标“ ”的配置无效，在物理I/F下引发。

但是，即使出现此故障，SPAN仍可正常运作。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” ｜Dst "192.168.254.1"



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

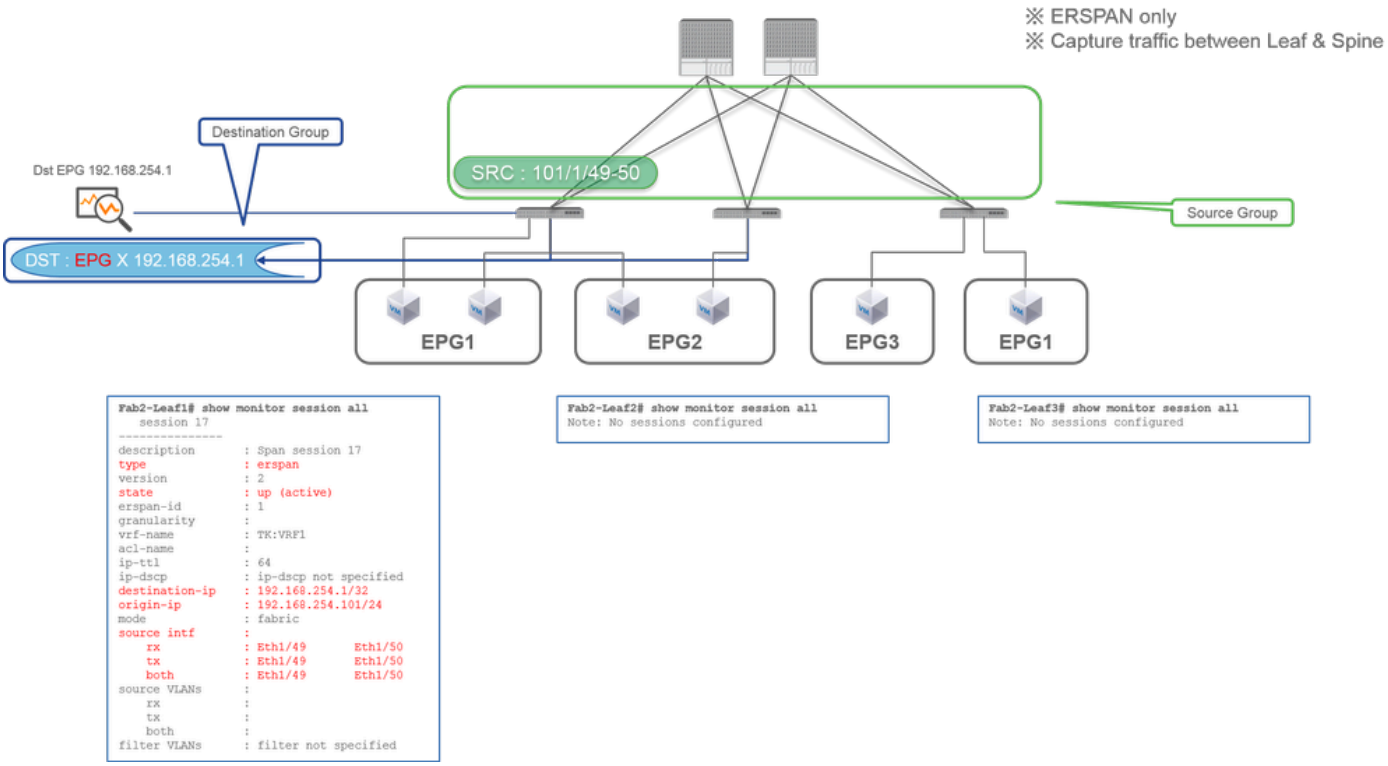
租户SPAN将EPG本身用作源，而接入SPAN仅将EPG用于过滤器。

租户SPAN的关键点是，您不必指定每个单独的端口，ACI会自动检测每台枝叶交换机上的相应VLAN。因此，当必须监控特定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，这将非常有用。

交换矩阵SPAN(ERSPAN)

例 1

源“Leaf1 e1/49-50” | Dst "192.168.254.1"



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

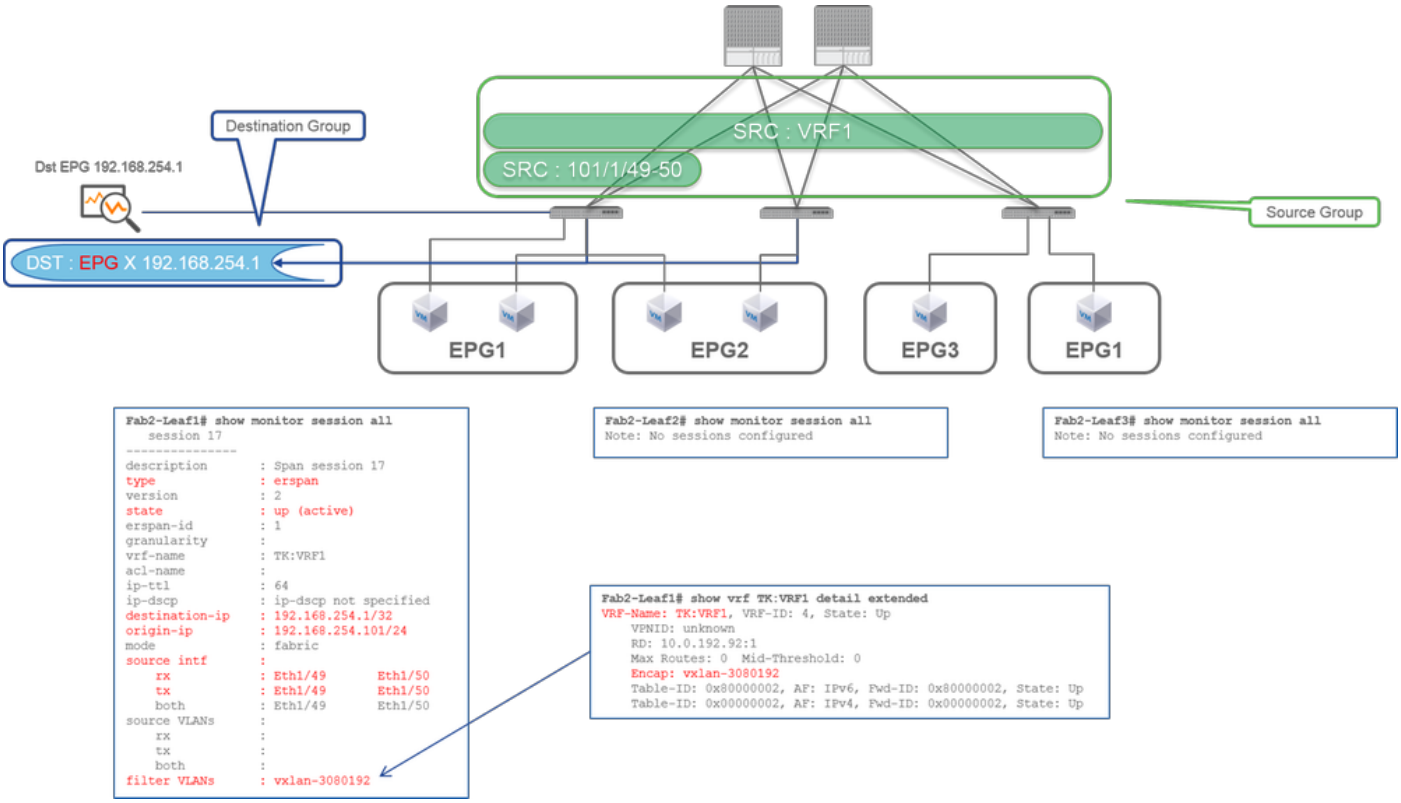
交换矩阵SPAN将交换矩阵端口指定为源，其中交换矩阵端口是枝叶和主干交换机之间的接口。

当需要在枝叶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。因此，需要一些技巧才能读到它。请参阅“如何读取SPAN数据”。

注意：iVxLAN报头是仅用于ACI交换矩阵内部使用的增强型VxLAN报头。

案例 2

源“Leaf1 e1/49-50和VRF过滤器”| Dst "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

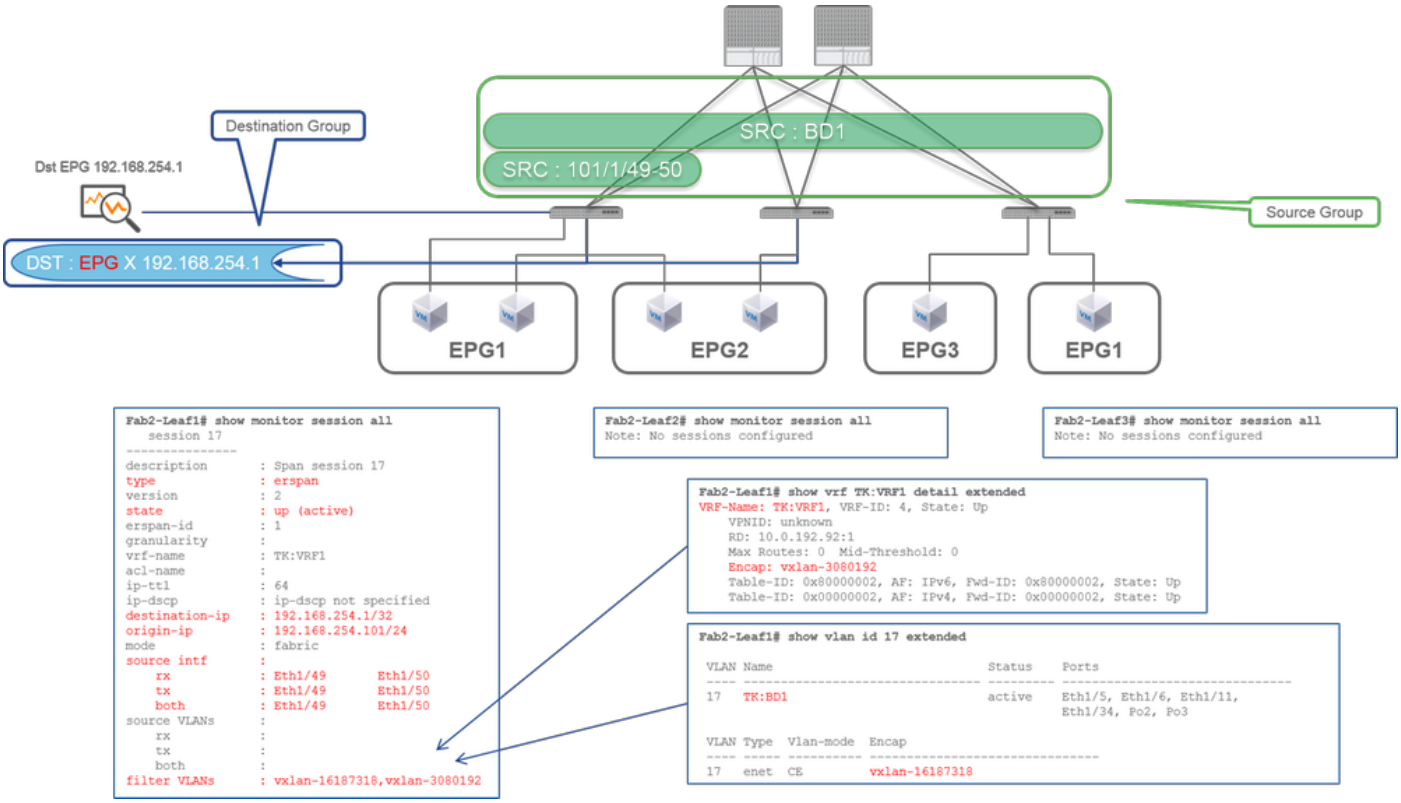
交换矩阵SPAN可以使用过滤器和访问SPAN。但过滤器类型不同。交换矩阵SPAN使用虚拟路由和转发(VRF)或BD作为过滤器。

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” | Dst "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

如前面的案例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。

SPAN到CPU

- 此功能从版本6.2(1)及更高版本开始
- 提供SPAN目标选项，用于将镜像流量发送到枝叶交换机的CPU端口(sup-eth0)，而不是外部收集器。
- 包括一个内置速率限制器，以防止SPAN流量超过CPU。
- SPAN统计信息（仅限 — HX）：HX平台交换机上的statistics命令可确认流量正被镜像到CPU。

配置步骤

- 创建新的SPAN目标组。
- 指定目标接口时，现在可以选择或输入sup-eth0（CPU端口）作为目标。
- 将此目标组与SPAN源组（定义要镜像的流量）关联。

视频链接：

https://community.cisco.com/t5/data-center-and-cloud-knowledge-base/aci-span-to-cpu-in-cisco-aci/ta-p/5372105

支持的平台

仅在这些平台上支持SPAN到CPU。此列表不是静态的，可以随时间而更改。

- FX2
- FX3
- GX
- GX2
- HX

SPAN统计支持

- 仅限HX

没有CLI来检查监察器速率，它使用500 Kbps硬编码。

限制和限制

请检查版本特定的版本信息，如思科APIC基本配置指南6.2(x)版。

- SPAN到CPU仅支持基于ACL的SPAN，不支持基于端口的SPAN。这意味着在缩放模式下不支持SPAN到CPU，仅在过滤器模式下支持。
- Cisco Nexus 9000 FX2交换机及更高版本支持SPAN到CPU功能
- 如果在不支持的交换机上配置了此功能，则会引发故障。检查APIC GUI上System > Faults下的Faults选项卡。
- 枝叶上的交换矩阵端口和vPC端口不支持作为SPAN到CPU会话的源接口。对于vPC端口，请使用路径类型VPC组件PC作为每台交换机的源，而不是虚拟端口通道。
- 仅支持接入端口和接入端口通道；接入端口可以是第2层或第3层端口。

- 不支持TX组播。
- 在SPAN到CPU会话和其他会话之间共享源时，策略器应用于两个会话。因此，不建议与其他会话共享SPAN到CPU会话的源。

配置汇总

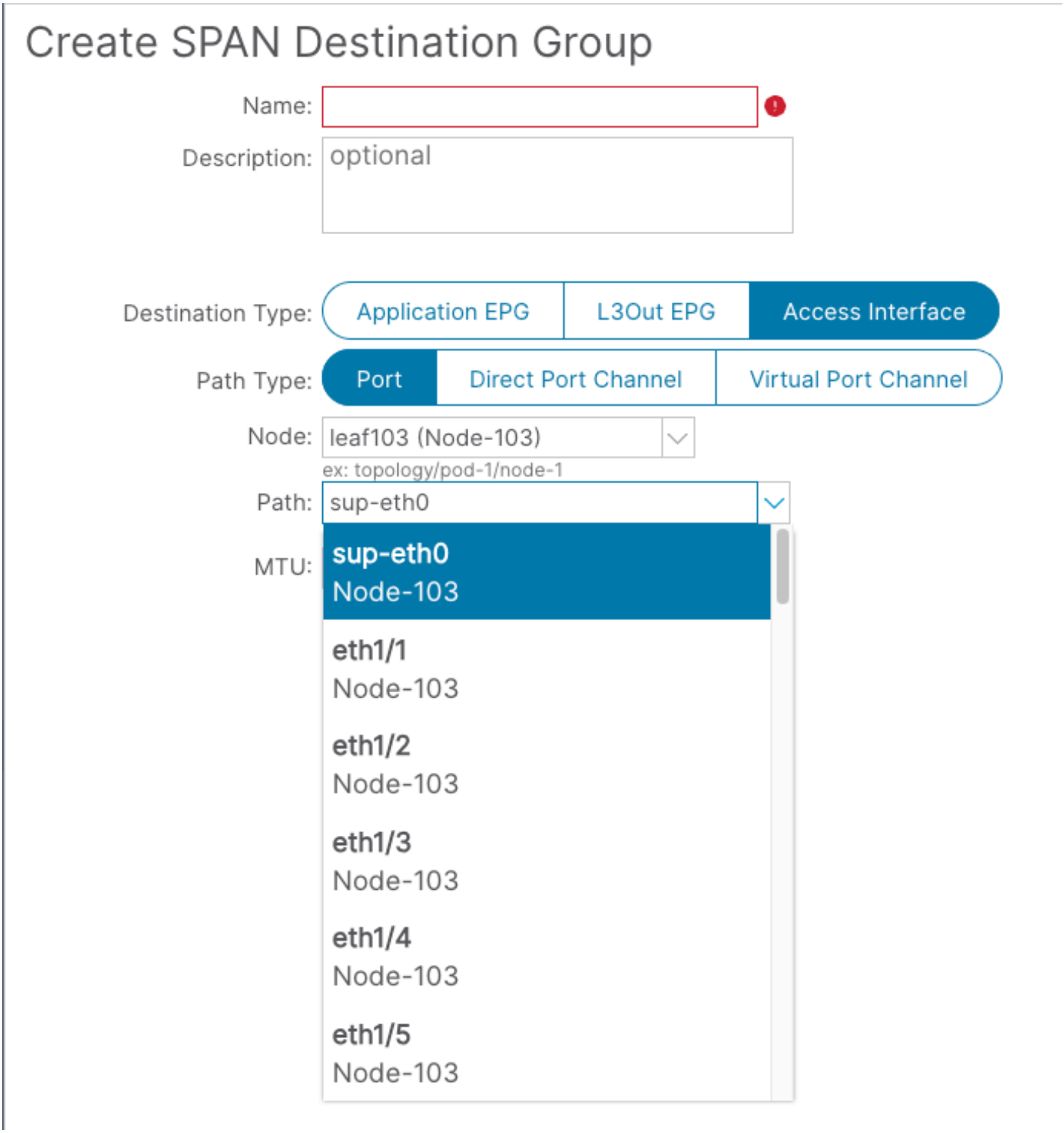
与配置本地SPAN一样，您可以配置：

- 以访问端口作为源路径的源组
- 目标为`sup-eth0`的目标组

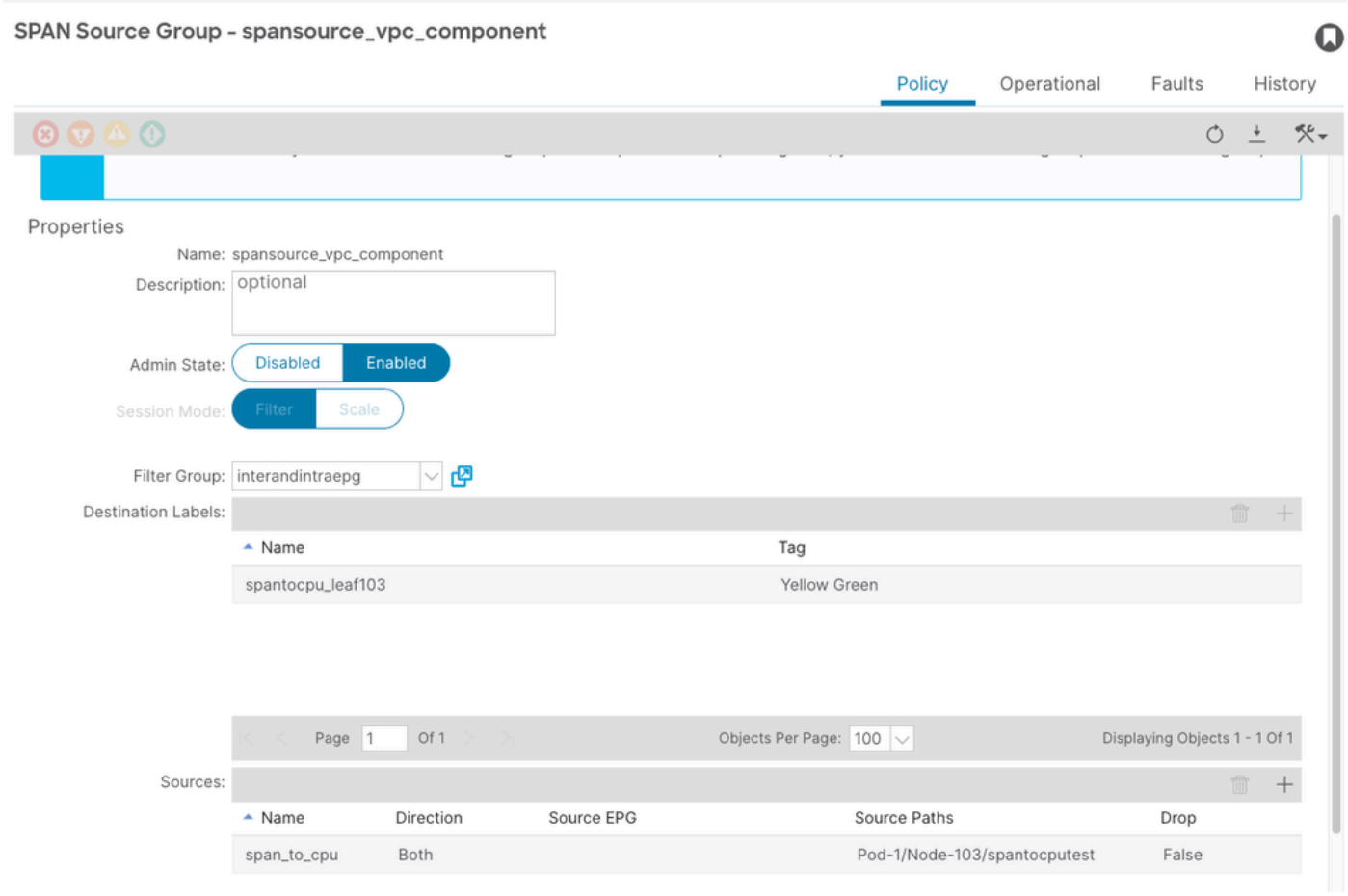
与典型的本地SPAN目标相比，唯一的区别是选择`sup-eth0`作为目标接口。

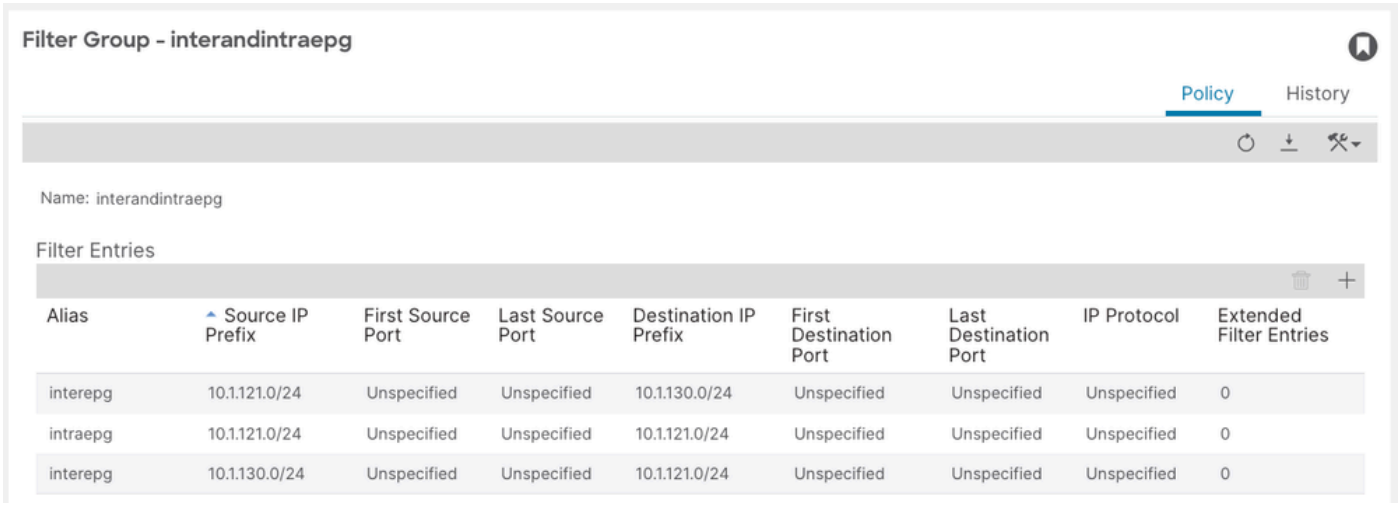
步骤 1：在GUI中配置SPAN到CPU

单击Fabric > Access Policies > Troubleshooting > SPAN > SPAN Destination Groups。



包含源、目标和过滤器信息的跨度配置：





步骤 2：检验枝叶上的SPAN会话

与任何SPAN配置一样，您可以使用交换机CLI验证会话：

```
leaf103# show monitor session all
```

```
 session 5

---------------

name : spansource_vpc_component

description : Span session 5

type : sup

scale-mode : filter

state : up (active)

mode : access

Filter Group : 4

source intf :

rx : [Po3, FG:4]

tx : [Po3, FG:4]

both : [Po3, FG:4]

source VLANs :

rx :

tx :

both :

filter VLANs : filter not specified

filter L3Outs : filter not specified

destination ports : sup-eth0
tx :
both :
filter VLANs : filter not specified
```

```
filter L3Outs : filter not specified
destination ports : sup-eth0
```

步骤 3：验证应用的过滤器

验证在枝叶交换机上编程的过滤器条目：

<#root>

```
leaf103#

show monitor filter 4
```

```
_____________________________________________________________________________
Filter name (id): interandintraepg (4)
Filter Entries:
Src IP                                          Dst IP                                          Src por
------                                          ------                                          -------
10.1.121.0/24                                   10.1.130.0/24                                   -
10.1.121.0/24                                   10.1.121.0/24                                   -
10.1.130.0/24                                   10.1.121.0/24                                   -
_____________________________________________________________________________
```

步骤 4：检验HX上的SPAN统计信息

每个会话的镜像数据包计数器显示流量是否匹配且被镜像。仅HX平台交换机。

```
aci-HX-leaf# vsh_lc
module-1# show platform internal hal span session all stats
SPAN hardware session id:30 asic:0 slice:0 packet counts:2557
SPAN hardware session id:31 asic:0 slice:0 packet counts:510634359
```

步骤 5：在枝叶上捕获和解码SPAN流量

配置后，所有匹配的流量(例如：通过eth1/4的子网之间的ICMP是SPAN到CPU。

从何处查看SPAN流量

- 可使用tahoe0接口上的tcpdump捕获SPAN流量。
- 可以使用内置脚本对其进行解码和过滤：
    - /isan/utils/span_to_cpu_parser.py

捕获和解码示例

捕获示例来自与配置部分所示不同的设置

```
pod16-leaf3# tcpdump -xxxvi tahoe0 | python3 isan/utils/span_to_cpu_parser.py --decoder tahoe --verbose
tcpdump: listening on tahoe0, link-type EN10MB (Ethernet), capture size 262144 bytes
Frame 9
Time: 1970-01-01T19:01:53.944622+00:00
ieth-extn:
sup_code=0x2b sup_qnum=0x6




icmp-request(8): id:0x7710, seq:0xf00
Payload:
Internal Header:
0x0000: 0101 0101 0101 0000 0000 0000 a00e 0000 ................
0x0010: 0404 0404 8889 0000 0000 ae00 0000 0006 ................
0x0020: 2b00 0000 fb00 0000 003f ffff ffc0 0000 +........?......
0x0030: 1000 0000 ....
Frame Data:
0x0000: 0022 bdf8 19ff acf2 c5f8 2e01 8100 0a6d ."............m
0x0010: 0800 4500 0054 ff89 0000 ff01 aec9 c0a8 ..E..T..........
0x0020: 7802 c0a8 1402 0800 a049 7710 0f00 114e x........Iw....N
0x0030: 5d69 7f67 0e00 cdab 0000 cdab 0000 cdab ]i.g............
0x0040: 0000 cdab 0000 cdab 0000 cdab 0000 cdab ................
0x0050: 0000 cdab 0000 cdab 0000 cdab ............
```



注意：span_to_cpu_parser.py是knet_parser.py，具有额外的函数来过滤具有ISTACK_SUP_CODE_SPAN(0x2b)（SPAN流量）的数据包。

将输出保存到PCAP

您可以使用 — write选项将解码输出保存到pcap文件：

```
pod16-leaf3# tcpdump -xxxvi tahoe0 | python3 /isan/utils/span_to_cpu_parser.py --decoder tahoe --write
```

```
tcpdump: listening on tahoe0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
Bye!
659 packets captured
662 packets received by filter
0 packets dropped by kernel
```

然后读回来：

```
pod16-leaf3# tcpdump -r /data/techsupport/mualshei-SPAN-CPU.pcap
reading from file /data/techsupport/mualshei-SPAN-CPU.pcap, link-type EN10MB (Ethernet)
20:07:43.966489 IP truncated-ip - 10 bytes missing! 192.168.120.2 > 192.168.20.2: ICMP echo request, id
20:07:43.966901 IP truncated-ip - 10 bytes missing! 192.168.20.2 > 192.168.120.2: ICMP echo reply, id 3
20:07:44.966505 IP truncated-ip - 10 bytes missing! 192.168.120.2 > 192.168.20.2: ICMP echo request, id
20:07:44.966843 IP truncated-ip - 10 bytes missing! 192.168.20.2 > 192.168.120.2: ICMP echo reply, id 3
```

问题

- SPAN会话关闭/“no-oper-src-dst”:确认源接口运行正常，平台支持SPAN到CPU功能和支持的源
- 不可用的目标路径Ep故障：验证APIC和枝叶交换机均处于支持该功能的版本，并确保平台支持SPAN到CPU功能
- 在tahoe0：上未观察到数据包:确认会话状态为up(active)，确认过滤器正确且流量匹配，确保您使用Filter Mode，并确认目标配置为sup-eth0，会话为local access SPAN（不是主干/不是交换矩阵/不是vPC源）

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tcpdump, wireshark，如此。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，请确保使用目标IP在ERSPAN的接口上运行捕获工具。

收到的数据包将封装为GRE报头。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）

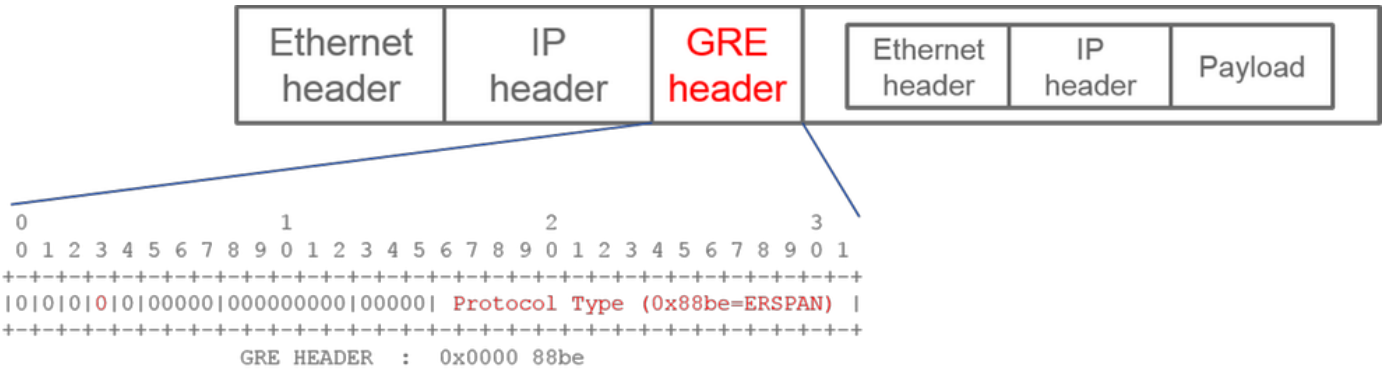
ERSPAN封装复制的数据包以将其转发到远程目标。GRE用于此封装。GRE报头上ERSPAN的协议类型为0x88be。

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。一、二、三。此RFC草案中提到ERSPAN类型。此外，此GRE RFC1701也有助于了解每个ERSPAN类型。

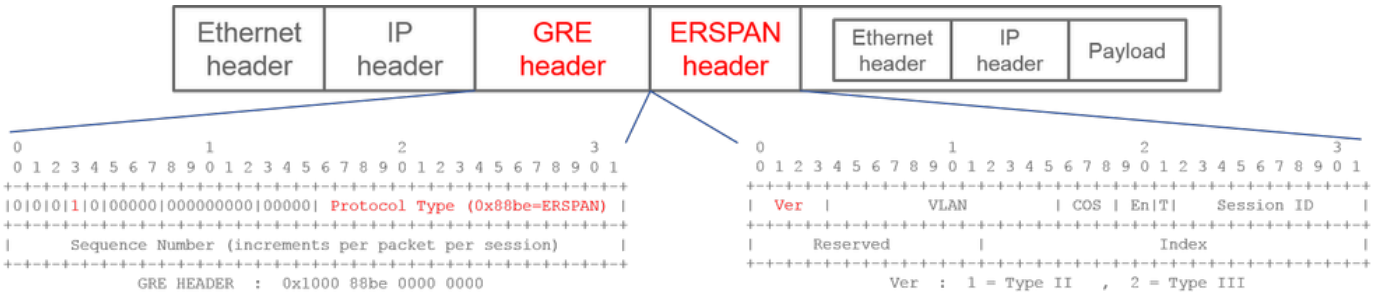
以下是每种类型的数据包格式：

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ERSPAN报头（如果是ERSPAN类型II和III，则此报头必须接替GRE报头）。Broadcom Trident 2仅支持此ERSPAN类型I。

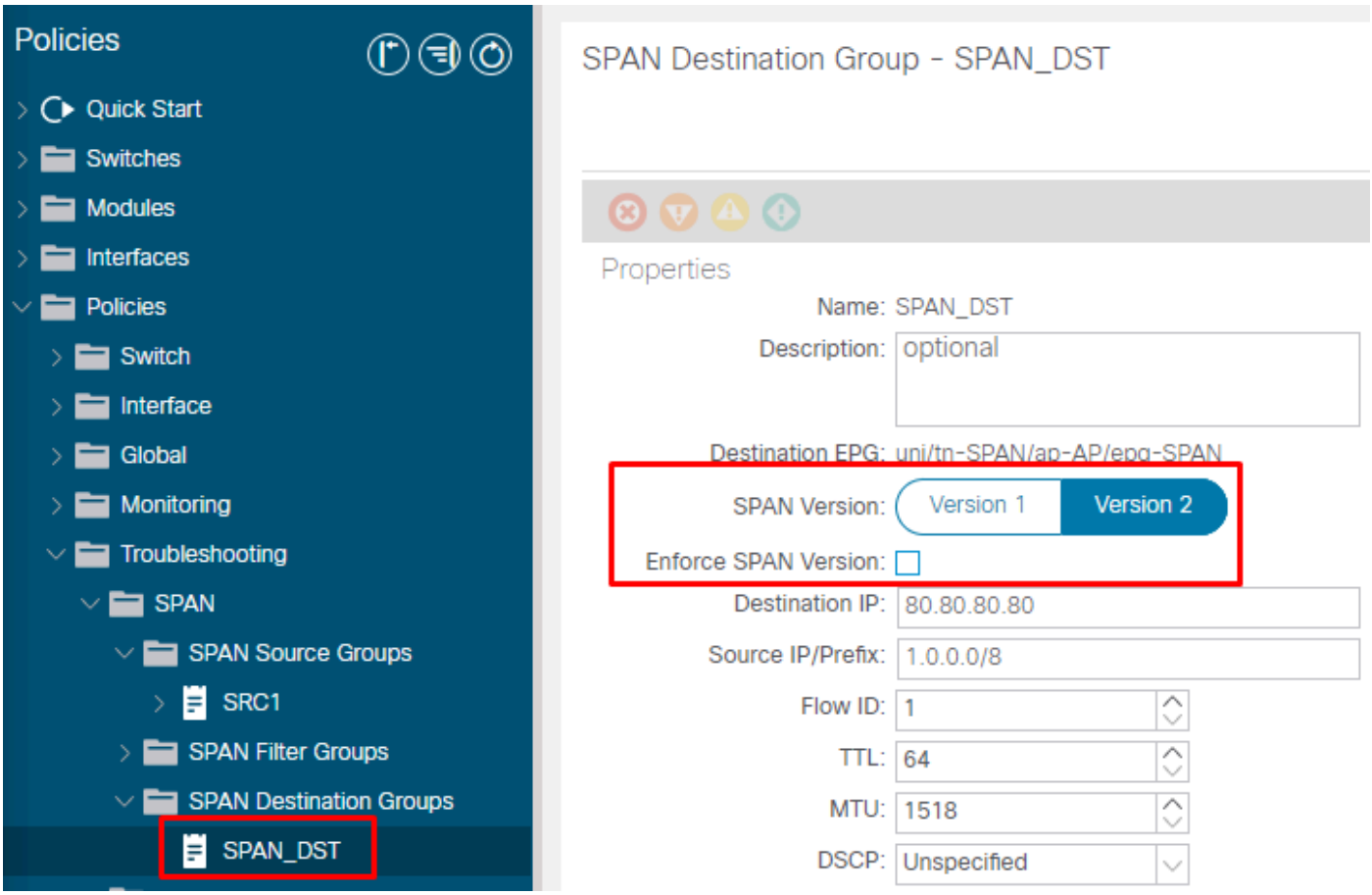
ERSPAN类型II或III



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。在ACI中，自2016年3月20日起，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，它仅解码ERSPAN类型II。如果启用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）：这指的是ERSPAN类型I或II
- 实施SPAN版本（选中或未选中）：这将决定SPAN会话是否必须失败，以防源节点硬件上不支持配置的ERSPAN类型。

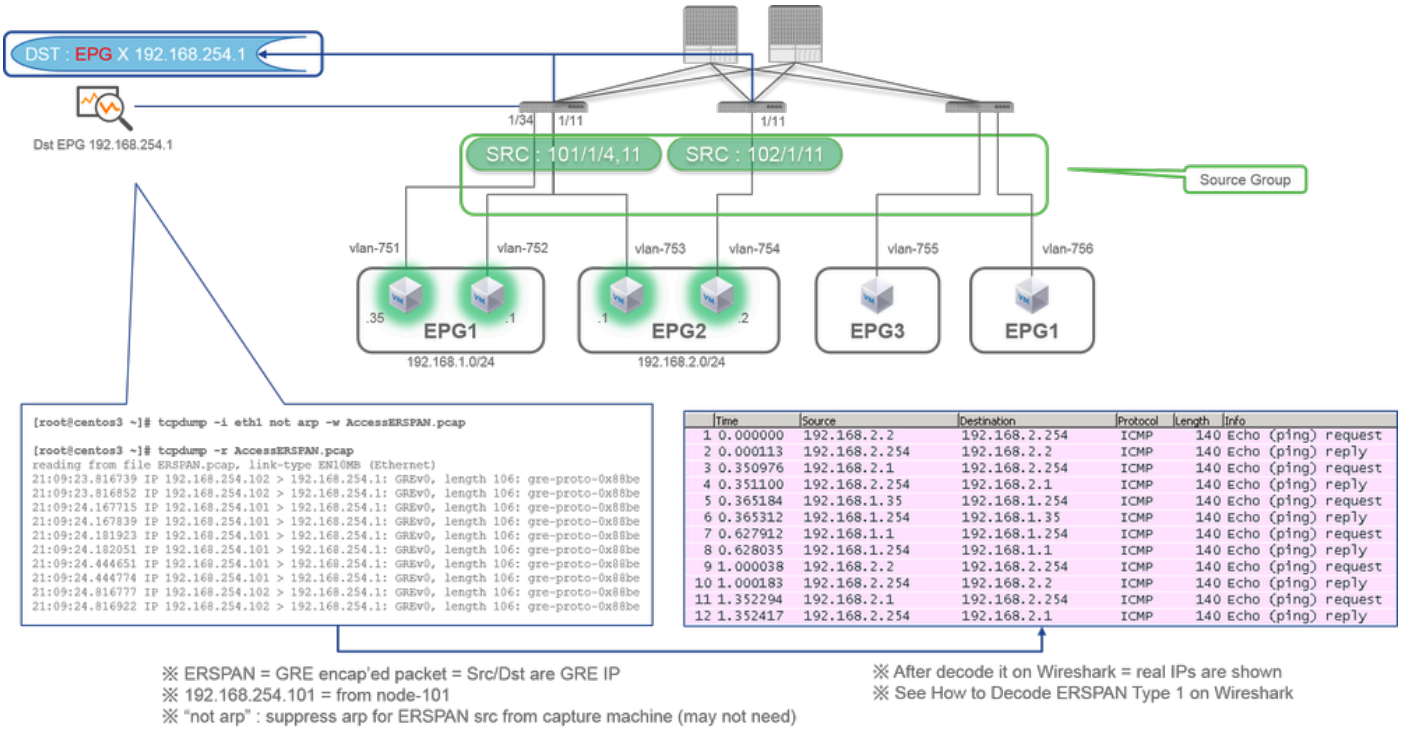
默认情况下，SPAN版本为版本2，而实施SPAN版本未选中。这意味着如果源节点是支持ERSPAN类型II的第2代或更高版本时，它会生成带有类型II的ERSPAN。如果源节点是不支持ERSPAN类型II的第1代（交换矩阵SPAN除外），它将回退到类型I，因为未选中Enforce SPAN Version。因此，ERSPAN目标收到混合类型的ERSPAN。

下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
2 版	未选中	使用类型I	使用类型II
2 版	已选中	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	已选中	使用类型I	使用类型I

ERSPAN数据示例

租户SPAN/接入SPAN(ERSPAN)



数据包需要解码，因为它由ERSPAN类型I封装。这可以通过Wireshark完成。请参阅“如何解码ERSPAN类型1”部分。

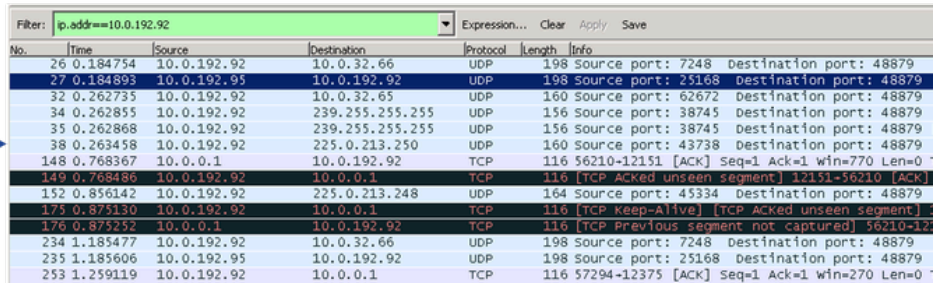
捕获数据包的详细信息（ERSPAN类型I）

```
[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



Wireshark自动解码ERSPAN类型II。但是，它仍由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）

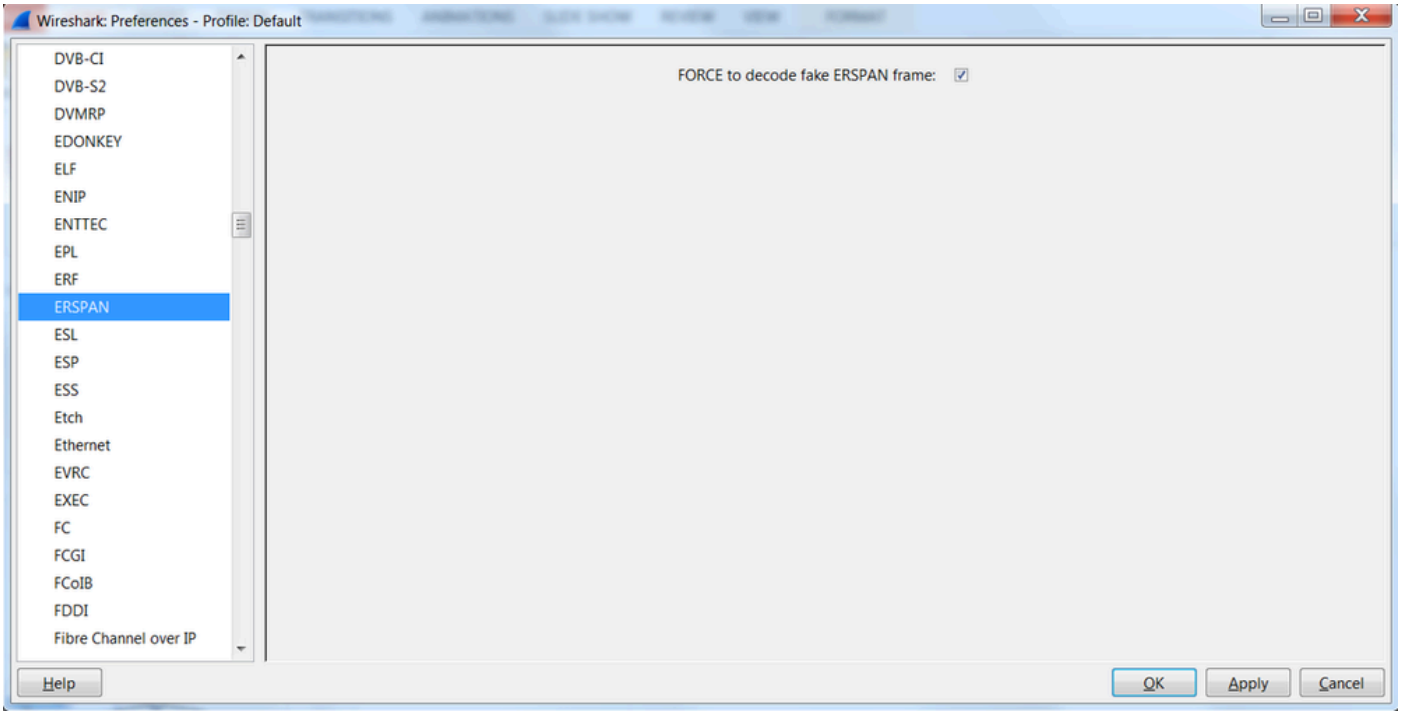
```
[root@centos3 ~]# tcpdump -xxr FabricERSPAN.pcap -c 1
reading from file FabricERSPAN.pcap, link-type EN10MB (Ethernet)
23:25:00.962224 IP 192.168.254.101 > 192.168.254.1: GREv0, seq 53341, length 164: gre-proto-0x88be
        0x0000:  0050 56bb 3096 0022 bdf8 19ff 0800 4500
        0x0010:  00b8 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

选项1.导航到编辑>首选项>协议> ERSPAN，然后选中FORCE以解码虚构ERSPAN帧。

- Wireshark(GUI)

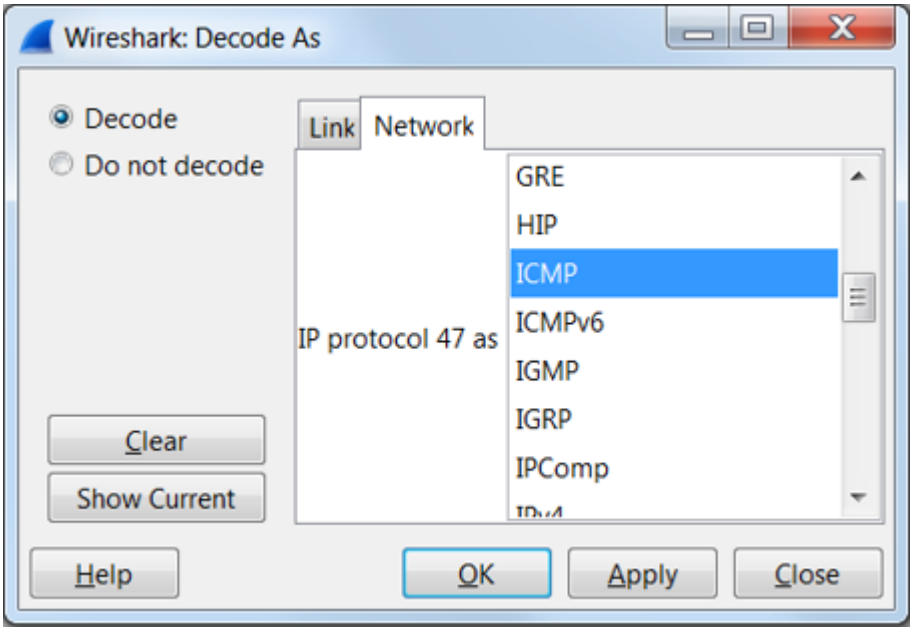


- Tshark（CLI版本的Wireshark）：

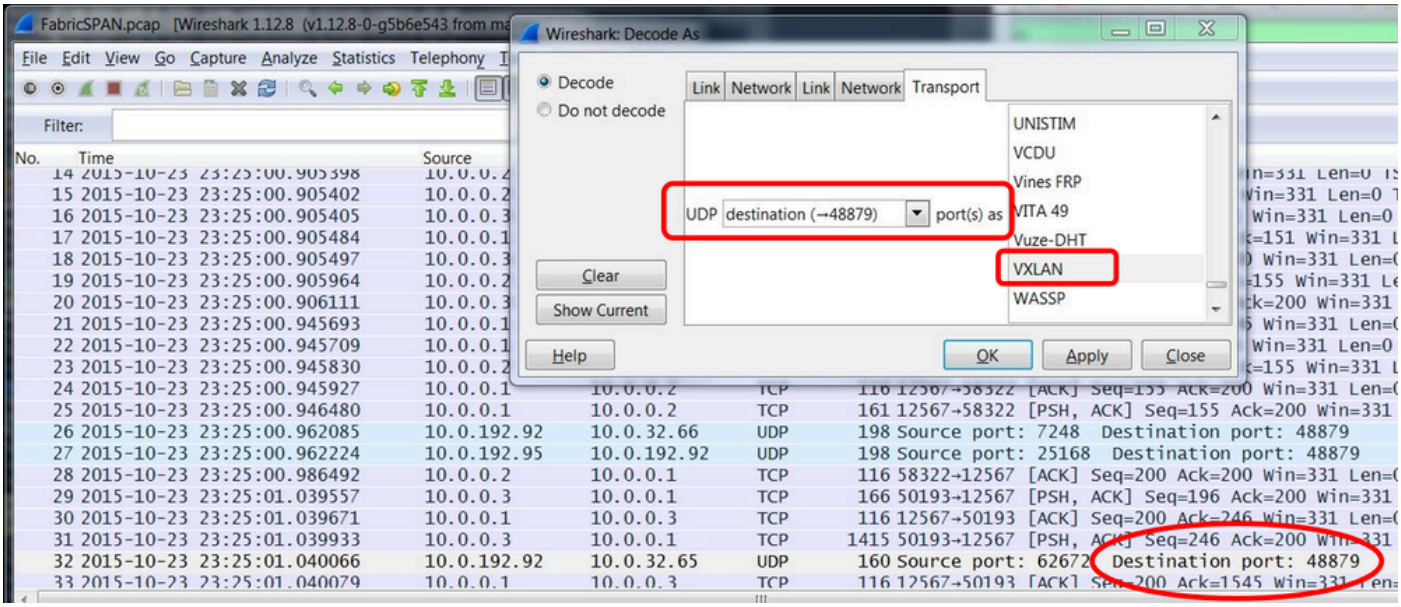
```
user1@linux# tshark -f 'proto GRE' -nV -i eth0 -o erspan.fake_erspan:true
```

注意：请务必在阅读ERSPAN类型II或III时禁用此选项。

选项2.导航至 Decode As > Network > ICMP (if it's ICMP).



如何解码iVxLAN报头

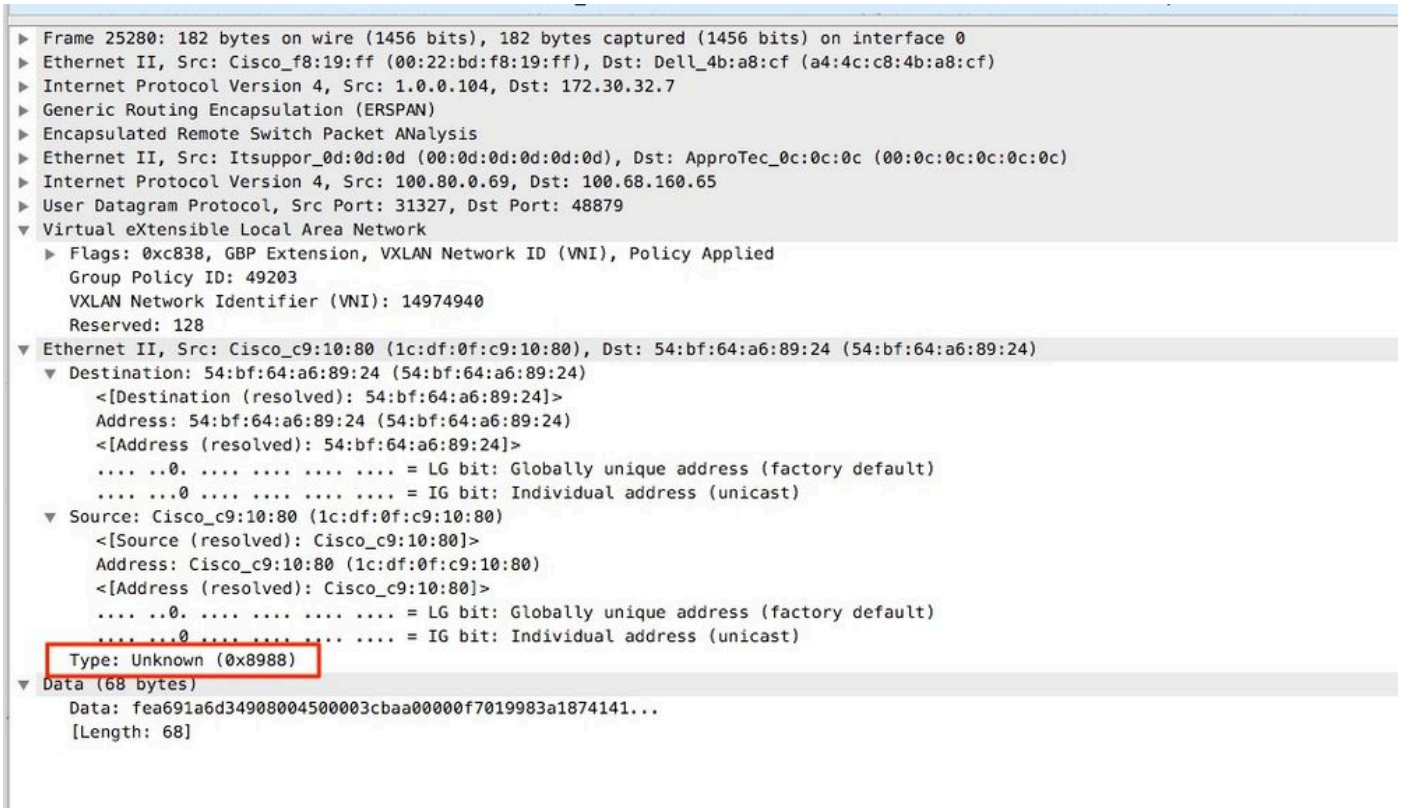


iVxLAN报头使用目标端口48879。因此，如果在Wireshark上将UDP目标端口48879配置为VxLAN，则可以解码iVxLAN报头和VxLAN。

1. 请确保首先选择iVxLAN封装数据包。
2. 导航到分析>解码为>传输> UDP目标(48879)> VxLAN。
3. 然后应用。

注意：交换矩阵端口上的APIC之间存在通信数据包。这些数据包不通过iVxLAN报头进行封装。

在运行精确时间协议(PTP)的用户网络上执行erspan捕获时，有时会看到Wireshark由于GRE封装(0x8988)中的未知以太网类型而无法解释数据。0x8988是在启用PTP时插入到数据平面数据包的时间标记的ethertype。将ethertype 0x8988解码为“Cisco ttag”，以显示数据包的详细信息。



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）。