

使用数据包着色技术或平台计数器排除数据包丢弃故障

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[拓扑](#)

[选项1.使用Flow-id的ERSPAN设置](#)

[步骤1. ESPAN目标设置](#)

[第 2a 步：为直接连接到SRC的流量创建Span源](#)

[步骤2b.为直接连接到DST的流量创建SPAN源](#)

[步骤3.快速Wireshark分析](#)

[选项2.平台计数器](#)

[清除平台计数器](#)

[使用低数据包或零数据包确定数据包大小](#)

[跟踪流量](#)

[相关信息](#)

简介

本文档介绍如何使用数据包着色技术跟踪网络流。

先决条件

要求

- ACI基础知识
- 终端组和合同
- Wireshark基础知识

使用的组件

本文档不限于特定的硬件和软件版本。

使用的设备：

- 运行版本5.3(2)的思科ACI
- Span目标

- 第2代交换机

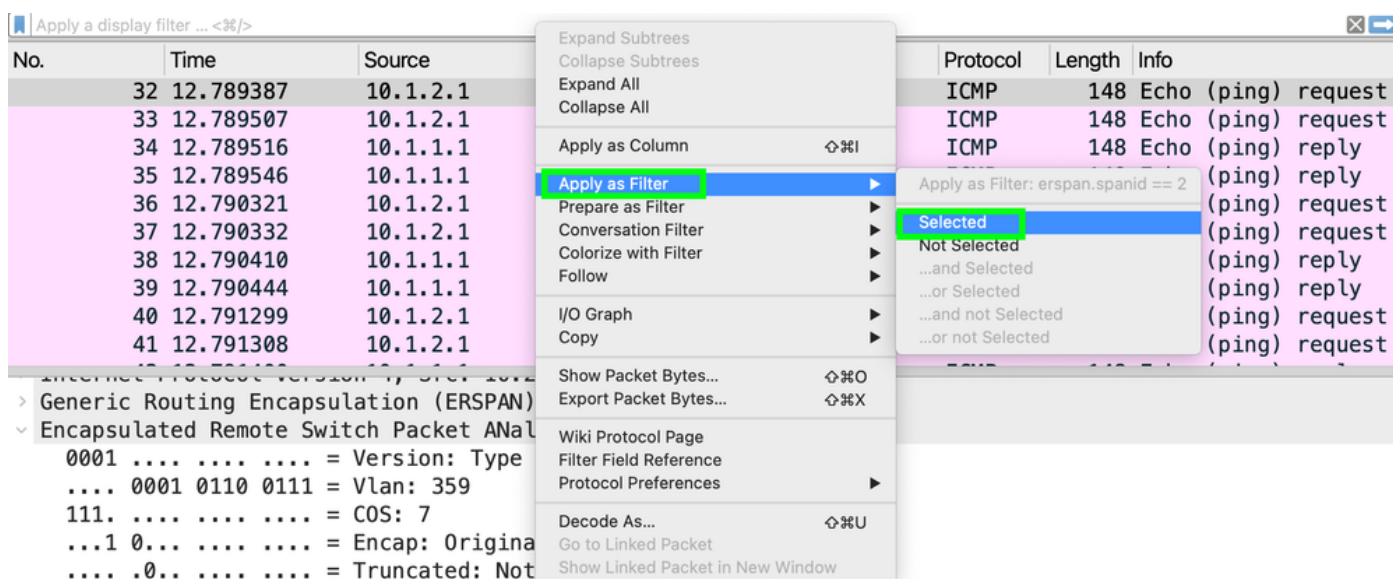
本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

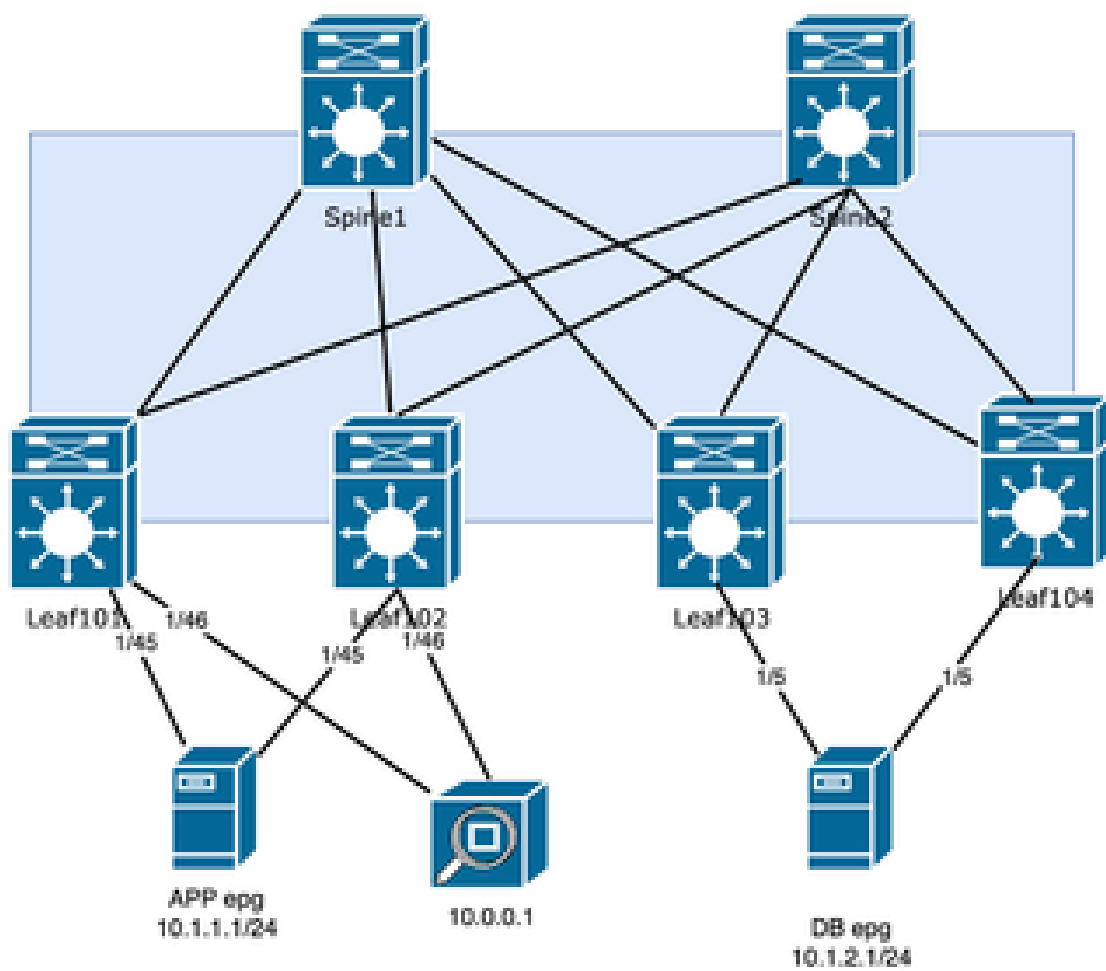
如何在Wireshark中创建过滤器。

打开capture。使用封装远程交换机数据包中的帧，选择SpanID行并右键单击。

选择Apply as Filter > Selected，如图所示：



拓扑



选项1.使用Flow-id的ERSPAN设置

如果目标服务器能够处理所有流量，则ERSPAN报头包含用于定义流ID的选项。可以配置此流ID以标识到交换矩阵的传入流量，也可以为传出流量设置不同的流ID。

步骤1. ESPAN目标设置

一个目标组的流ID为1

在Fabric > Access Policies > Policies > Troubleshooting > SPAN > SPAN Destination Groups下

Create SPAN Destination Group



Name: All-dst-jr-flowid

Description: optional

Destination Type: EPG Access Interface

Destination EPG: jr

Tenant



ALL

Application Profile

monitor

EPG

SPAN Version: Version 1 Version 2

Enforce SPAN Version: ☐

Destination IP: 10.0.0.1

Source IP/Prefix: 10.255.0.0/16

Flow ID: 1

TTL: 64

MTU: 8000

DSCP: Unspecified

Cancel

Submit

在第二个目标组上，将flow-id配置为2:

Create SPAN Destination Group

Name:

All-dst-jr-flowid2

Description:

optional

Destination Type:

EPG

Access Interface

Destination EPG:

jr

Tenant

ALL

Application Profile

monitor

EPG

SPAN Version:

Version 1

Version 2

Enforce SPAN Version:

☐

Destination IP:

10.0.0.1

Source IP/Prefix:

10.255.0.0/16

Flow ID:

2

TTL:

64

MTU:

8000

DSCP:

Unspecified

Cancel

Submit

第 2a 步：为直接连接到SRC的流量创建SPAN源

在Fabric > Access Policies > Policies > Troubleshooting > SPAN > SPAN Source Groups下

Create SPAN Source Group

Name:

Src-jr-1

Description:

optional

Admin State:

Disabled

Enabled

Filter Group:

select an option

Destination Group:

All-dst-jr-flowid1

Create Sources

Name	Direction	Source EPG	Source Paths
------	-----------	------------	--------------

通过添加路径和EPG来过滤更多流量。实验室示例是Tenant jr Application Profile ALL和EPG应用。

Create SPAN Source

✕

Name:

APP-epg-jr

Description:

optional

Direction:

Both

Incoming

Outgoing

Filter Group:

select an option

▼

Span Drop Packets:

☐

Type:

None

EPG

Routed Outside

Source EPG:

jr

▼

Tenant

ALL

▼

Application Profile

app

▼

EPG

Add Source Access Paths

🗑️

+

Source Access Path
Pod-1/Node-101/VPC-ESX-169
Pod-1/Node-102/VPC-ESX-169

Cancel

OK

步骤2b.为直接连接到DST的流量创建SPAN源

在Fabric > Access Policies > Policies > Troubleshooting > SPAN > SPAN Source Groups下

Create SPAN Source

×

Description: optional

Direction: Both Incoming Outgoing

Filter Group: select an option

Span Drop Packets: ☐

Type: None EPG Routed Outside

Source EPG: jr ALL db

Tenant Application Profile EPG

Add Source Access Paths

⌵ +

Source Access Path
Pod-1/Node-103/eth1/6

通过不仅添加路径，而且添加EPG数据库，可以更好地过滤流量：

Create SPAN Source Group

×

Name: Src-epg-2

Description: optional

Admin State: Disabled Enabled

Filter Group: select an option

Destination Group: All-dst-jr-flowid2

Create Sources

⌵ +

Name	Direction	Source EPG	Source Paths
------	-----------	------------	--------------

步骤3.快速Wireshark分析

在本示例中，您将验证ICMP请求数据包的数量与ICMP响应数据包的数量匹配，确保ACI交换矩阵内没有丢包。

在wireshark上打开捕获，使用与SRC和DST IP一起配置的SPAN ID /Flow-ID创建过滤器：

```
<#root>
(erspan.spanid ==

and

) && (ip.src==

and ip.dst ==

)
```

用于实验室测试流程的过滤器：

```
<#root>
(erspan.spanid == 1 and icmp) && (ip.src== 10.1.2.1 and ip.dst == 10.1.1.1)
```

验证Displayed数据包是否与发送的数据包相同：

(erspan.spanid == 1 and icmp) && (ip.src == 10.1.2.1 and ip.dst == 10.1.1.1)

No.	Time	Source	Destination	Protocol	Length	Info
33	12.789507	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
37	12.790332	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
41	12.791308	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
45	12.792088	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
49	12.792891	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
53	12.793663	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
57	12.794455	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
61	12.795259	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
65	12.796080	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
69	12.796812	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request

> Frame 33: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits)
 > Ethernet II, Src: Cisco_f8:19:ff (00:22:bd:f8:19:ff), Dst: VMware_b7:4c:66 (00:50:56:b7:4c:66)
 > Internet Protocol Version 4, Src: 10.255.0.102, Dst: 10.0.0.1
 > Generic Routing Encapsulation (ERSPAN)
 > Encapsulated Remote Switch Packet Analysis Type II
 0001 = Version: Type II (1)
 1010 0111 1110 = Vlan: 2686
 000. = COS: 0
 ...1 0... = Encap: Originally 802.1Q encapsulated (2)
0.. = Truncated: Not truncated (0)
00 0000 0001 = SpanID: 1
 0000 0000 0000 = Reserved: 0

SpanID (erspan.spanid), 10 bits Packets: 4109 Displayed: 1000 (24.3%) Profile:

下一个SPAN ID必须具有相同的数量；如果没有，则数据包在交换矩阵内被丢弃。

过滤器：

(erspan.spanid == 2 and icmp) && (ip.src == 10.1.2.1 and ip.dst == 10.1.1.1)

(erspan.spanid == 2 and icmp) && (ip.src == 10.1.2.1 and ip.dst == 10.1.1.1)

No.	Time	Source	Destination	Protocol	Length	Info
32	12.789387	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
36	12.790321	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
40	12.791299	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
44	12.792076	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
48	12.792880	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
52	12.793654	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
56	12.794434	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
60	12.795250	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
64	12.796038	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
68	12.796797	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request

> Frame 32: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits)
 > Ethernet II, Src: Cisco_f8:19:ff (00:22:bd:f8:19:ff), Dst: VMware_b7:4c:66 (00:50:56:b7:4c:66)
 > Internet Protocol Version 4, Src: 10.255.0.103, Dst: 10.0.0.1
 > Generic Routing Encapsulation (ERSPAN)
 > Encapsulated Remote Switch Packet Analysis Type II
 0001 = Version: Type II (1)
 0001 0110 0111 = Vlan: 359
 111. = COS: 7
 ...1 0... = Encap: Originally 802.1Q encapsulated (2)
0.. = Truncated: Not truncated (0)
00 0000 0010 = SpanID: 2
 0000 0000 0000 = Reserved: 0

SpanID (erspan.spanid), 10 bits Packets: 4109 Displayed: 1000 (24.3%) Profile:

选项2.平台计数器

此方法利用Nexus跟踪具有不同数据包大小的各个接口的性能，但该方法要求至少队列具有很低的流量（如果不是零的话）。

清除平台计数器

进入单个交换机并清除连接到设备的单个接口。

```
<#root>
```

```
Switch#
```

```
vsh_lc -c "clear platform internal counters port
```

```
"
```

```
<#root>
```

```
LEAF3#
```

```
vsh_lc -c "clear platform internal counters port 6"
```

```
LEAF1#
```

```
vsh_lc -c "clear platform internal counters port 45"
```

```
LEAF2#
```

```
vsh_lc -c "clear platform internal counters port 45"
```

使用低数据包或零数据包确定数据包大小

查找在所有RX和TX枝叶中可能没有计数器的数据包大小：

```
<#root>
```

```
vsh_lc -c 'show platform internal counters port
```

```
' | grep X_PKT
```

在下一个示例中，数据包大小大于512且小于1024:

```
<#root>
```

```
LEAF101#
```

```
vsh_lc -c "show platform internal counters port 45 " | grep X_PKT
```

RX_PKTOK	1187
RX_PKTTOTAL	1187
RX_PKT_LT64	0
RX_PKT_64	0
RX_PKT_65	1179
RX_PKT_128	8
RX_PKT_256	0
RX_PKT_512	0 <<
RX_PKT_1024	0
RX_PKT_1519	0
RX_PKT_2048	0
RX_PKT_4096	7
RX_PKT_8192	43
RX_PKT_GT9216	0
TX_PKTOK	3865
TX_PKTTOTAL	3865
TX_PKT_LT64	0
TX_PKT_64	0
TX_PKT_65	3842
TX_PKT_128	17
TX_PKT_256	6
TX_PKT_512	0 <<
TX_PKT_1024	10
TX_PKT_1519	3
TX_PKT_2048	662
TX_PKT_4096	0
TX_PKT_8192	0
TX_PKT_GT9216	0

此步骤需要在转发数据包的链路中执行。

跟踪流量

从服务器10.1.2.1发送1000个数据包，数据包大小为520。

在枝叶103接口1/6上验证，RX上的流量在此接口上发起：

<#root>

MXS2-LF103#

```
vsh_lc -c "show platform internal counters port 6 " | grep X_PKT_512
```

RX_PKT_512	1000
TX_PKT_512	647

1000个数据包RX，但只有647个数据包作为应答发送。

下一步是检查其他服务器的传出接口：

对于Leaf102:

<#root>

MXS2-LF102#

```
vsh_lc -c "show platform internal counters port 45 " | grep X_PKT_512
```

RX_PKT_512	0
TX_PKT_512	1000

交换矩阵未丢弃请求。

对于枝叶101,RX数据包647，与ACI TX的数据包数量相同。

<#root>

MXS2-LF101#

```
vsh_lc -c "show platform internal counters port 45 " | grep X_PKT_512
```

RX_PKT_512	647
TX_PKT_512	0

相关信息

[排除ACI交换矩阵内转发故障 — 间歇丢弃](#)

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）。