排除Nexus 7000系列交换机上的硬件转发问题

目录

简介

本文档介绍如何对Cisco Nexus 7000系列交换机的F3系列模块上的硬件转发问题进行故障排除。

先决条件

要求

思科建议您先熟悉Cisco Nexus操作系统(NX-OS)和基本Nexus架构，然后再继续了解本文档中介绍的信息。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Cisco Nexus 7000系列交换机(N7K)
- 思科N7K F3系列模块（N7K-F312FQ-25,12端口10/40千兆以太网模块）
- Cisco NX-OS版本6.2.8a及更高版本

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

本文档主要介绍一些内置工具，当您已用尽转发表或控制平面的软件部分时，这些工具用于硬件故障排除。嵌入式逻辑分析器模块(ELAM)是一种专用集成电路(ASIC)，用于捕获单个数据包并显示转发后入口数据包在数据总线(DBUS)和结果总线(RBUS)上的显示方式。

ASIC嵌入到转发管道中，它可以实时捕获数据包，而不会中断性能或控制平面资源。这有助于回答以下问题：

- 数据包是否到达转发引擎(FE)?
- 数据包在哪个端口和VLAN上接收？
- 数据包如何显示(第2层(L2)或第4层(L4)数据)?
- 数据包如何更改，以及它在何处发送？

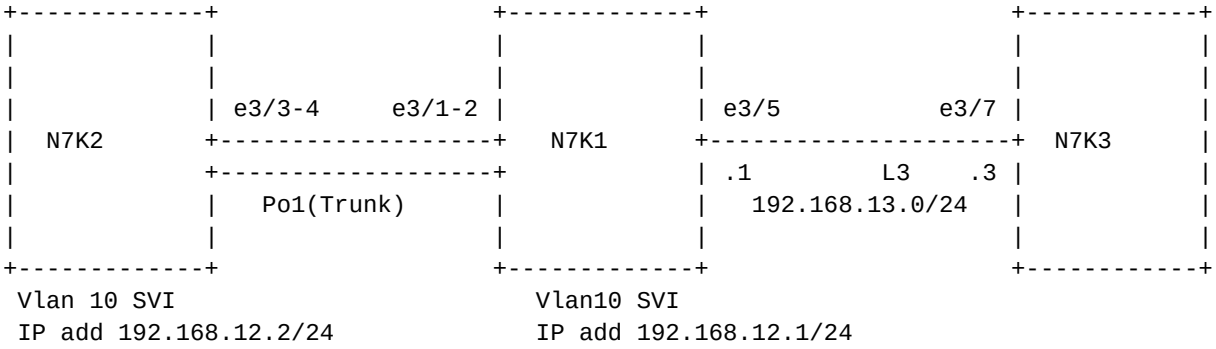
ELAM是功能强大、精细且非侵入性的工具，在硬件交换平台上工作的思科技术支持中心(TAC)工程师最常使用。但是，必须知道ELAM工具一次只捕获一个数据包。即，触发ELAM后收到的第一个数据包。

故障排除

本节介绍如何排除F3系列模块上的ELAM在不涉及使用分支电缆的部署以及使用分支电缆的部署中的故障。

排除F3系列模块上的ELAM故障，无分支电缆

以下是本节中用于示例的拓扑：



以下是有关此拓扑的一些说明：

- N7K运行NX-OS版本6.2.8a。
- 从N7K2 VLAN 10接口向远程IP地址192.168.12.1发送ping。
- ELAM捕获N7K1上的数据包。
- 使用N7K-F312FQ-25，该模块是插入插槽3的12端口10/40千兆以太网模块。

在对系统进行故障排除之前，您应确认基本连接：

```
N7K2# ping 192.168.13.3
PING 192.168.13.3 (192.168.13.3): 56 data bytes
64 bytes from 192.168.13.3: icmp_seq=0 ttl=253 time=1.513 ms
64 bytes from 192.168.13.3: icmp_seq=1 ttl=253 time=1.062 ms
64 bytes from 192.168.13.3: icmp_seq=2 ttl=253 time=0.822 ms
64 bytes from 192.168.13.3: icmp_seq=3 ttl=253 time=0.830 ms
64 bytes from 192.168.13.3: icmp_seq=4 ttl=253 time=0.845 ms

--- 192.168.13.3 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.822/1.014/1.513 ms


N7K2# show ip route 192.168.13.3
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

192.168.13.0/24, ubest/mbest: 1/0
 *via 192.168.12.1, [1/0], 01:20:36, static

!--- The next command verifies the Address Resolution Protocol (ARP) for the next hop.

N7K2# show ip arp 192.168.12.1
----SNIP----
IP ARP Table
Total number of entries: 1
Address         Age       MAC Address     Interface
192.168.12.1    00:10:29  e4c7.2210.a142  Vlan10
```

您还应验证Supervisor引擎(Sup)上的介质访问控制(MAC)地址学习和下一跳的模块：

```
N7K2# show mac address-table address e4c7.2210.a142

!--- This command output shows the MAC learning on the Sup (software).

Legend:
       * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
       age - seconds since last seen,+ - primary entry using vPC Peer-Link,
       (T) - True, (F) - False
  VLAN     MAC Address      Type      age     Secure NTFY Ports/SWID.SSID.LID
---------+-----------------+--------+---------+------+----+------------------
* 10       e4c7.2210.a142    dynamic   120        F    F    Po1
```

此输出显示模块/硬件上的MAC学习；但是，为了了解接口，必须转换索引：

```
N7K2# show hardware mac address-table 3 address e4c7.2210.a142
FE | Valid| PI|  BD  |      MAC      | Index | Stat| SW  | Modi| Age| Tmr| GM| Sec|
  |      |   |      |               |       |  ic |     | fied|Byte| Sel|   | ure|
---|------|---|------|---------------|-------|-----|-----|-----|----|----|---|----|

 1    1     1   41    e4c7.2210.a142  0x00a2a   0   0x089   1    185   1    0    0


| TR| NT| RM| RMA| Cap| Fld|Always| PV | RD| NN| UC|PI_E8| VIF | SWID| SSWID|  LID  |
| AP| FY|   |    |ture|    | Learn|    |   |   |   |      |     |     |      |       |
|---|---|---|----|----|----|------|----|---|---|---|------|-----|-----|------|-------|
 0   0   0    0    0    0    0    0x00  0   0   1    0   0x000 0x000  0x000 0x00a2a


N7K2# show system  internal pixm info ltl 0x00a2a

!--- This is the index that was received in the previous output.

---SNIP---

PC_TYPE    PORT    LTL      RES_ID       LTL_FLAG      CB_FLAG      MEMB_CNT
------------------------------------------------------------------------------
Normal    Po1     0x0a2a    0x16000000   0x00000000    0x00000002   2
```

```
Member rbh rbh_cnt
Eth3/4   0x000000f0   0x04
Eth3/3   0x0000000f   0x04

---SNIP---
```

输入以下命令以获取虚拟设备环境(VDC)编号(在本例中为**3**)，并直接检查模块上的MAC:

```
N7K2# show vdc

---SNIP---

vdc_id  vdc_name  state   mac                type        lc
------  --------  -----   ----------         ---------   ---
3       N7K2      active  e4:c7:22:10:a1:43  Ethernet    f3


module-3#attach module 3
module-3# vdc 3

!--- This data is obtained from the previous command output.

module-3# show mac address-table address e4c7.2210.a142
Legend:
       * - primary entry, G - Gateway MAC, (R) - Routed MAC, (d) - dec
       Age - seconds since last seen,,+ - primary entry using vPC Peer-Link
       (T) - True, (F) - False, h - hex, d - decimal

VDC = 3

 FE  VLAN/BD MAC Address    Type     Age   Secure  NTFY   Ports/SWID.SSID.LID(d)
-----+------+-------------+--------+------+------+------+----------------------
*  1    10  e4c7.2210.a142  dynamic 360        F     F    Po1
```

确定端口通道1上用于从N7K2转发Sup上流量的链路，以及当端口通道1从N7K1到N7K2使用时用于从N7K3发送应答的链路：

```
N7K2# show port-channel load-balance forwarding-path interface port-channel 1 src-ip
 192.168.12.2 dst-ip 192.168.13.3 module 3
Module 3: Missing params will be substituted by 0's.
Load-balance Algorithm: src-dst ip
RBH: 0xd2       Outgoing port id: Ethernet3/3


N7K1# show port-channel load-balance forwarding-path interface port-channel 1 src-ip
192.168.13.3 dst-ip 192.168.12.2 module 3
Module 3: Missing params will be substituted by 0's.
 Load-balance Algorithm: src-dst ip
RBH: 0xd2       Outgoing port id: Ethernet3/1
```

从N7K2（IP地址192.168.12.2）发送ping命令，并在N7K1上沿入口方向捕获数据包，以确认数据包被转发到N7K3（IP地址192.168.13.3）。

在发送ping之前，您应该了解硬件累积。完成以下步骤以了解累积：

1. 附加模块：

```
N7K1# attach module 3
```

```
Attaching to module 3 ...
To exit type 'exit', to abort type '$.'
```

2. 确定侧*翼实*例。侧翼是F3系列模块的片上交换(SOC)ASIC。每个侧面模块映射到模块上的两个外部端口（信息会按模块类型更改，并特定于N7K-F312FQ-25）。

   模块上有12个端口，每个ASIC映射到前面板上的两个端口，这意味着模块上有6(0-5)个侧面实例（实例计数为零）。 **注意：**在开始之前，请确保您具有网络管理权限。捕获通过N7K1上的端口通道1从N7K2到达的数据包时，请查找映射到每个实例的端口（e3/1和e3/2）：

```
module-3# show hardware internal dev-port-map
--------------------------------------------------------------
CARD_TYPE:       12 port 40G
>Front Panel ports:12
--------------------------------------------------------------
 Device name             Dev role              Abbr num_inst:
--------------------------------------------------------------
>Flanker Eth Mac Driver DEV_ETHERNET_MAC       MAC_0  6
>Flanker Fwd Driver     DEV_LAYER_2_LOOKUP     L2LKP  6

!--- Check for the L2LKP number for ports 1 and 2.

>Flanker Xbar Driver    DEV_XBAR_INTF          XBAR_INTF 6
>Flanker Queue Driver   DEV_QUEUEING           QUEUE  6
>Sacramento Xbar ASIC   DEV_SWITCH_FABRIC      SWICHF 1
>Flanker L3 Driver      DEV_LAYER_3_LOOKUP     L3LKP  6
>EDC                    DEV_PHY                PHYS   2
+-----------------------------------------------------------------------+
+----------------+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++------------+
+-----------------------------------------------------------------------+
FP port |  PHYS | MAC_0 | L2LKP | L3LKP | QUEUE |SWICHF
   1               0       0       0       0       0

!--- The L2KLP for both ports is 0, so both belong to instance 0.

  2               0       0       0       0       0
 3               1       1       1       1       0
  4               1       1       1       1       0
  5       0       2       2       2       2       0
  6       0       2       2       2       2       0
  7       1       3       3       3       3       0
  8       1       3       3       3       3       0
  9               4       4       4       4       0
  10              4       4       4       4       0
  11              5       5       5       5       0
  12              5       5       5       5       0
+-----------------------------------------------------------------------+
+-----------------------------------------------------------------------+
```

3. 选择实例，设置触发器，然后开始捕获。但是，必须了解ELAM触发器可以使用许多选项：

```
module-3# elam asic flanker instance 0
module-3(fln-elam)# layer2
module-3(fln-l2-elam)# trigger ?
 dbus  Pre L2 BUS
 rbus  Post L2 BUS

-----SNIP-----
```

   如果要将DBUS包括在捕获中（交换机接收的数据包），这两个选项非常重要。 这是未经查找

的原始数据包。RBUS显示DBUS硬件中的查找结果。要获得完整的ELAM和分析，必须同时捕获RBUS和DBUS。

下一个输出显示可使用DBUS选项捕获的数据包的类型。在本示例中，选择Internet协议版本4(IPv4)数据包：

```
module-3(fln-l2-elam)# trigger dbus ?
 arp    ARP Frame Format
 fc     Fc hdr Frame Format
 ipv4   IPV4 Frame Format
 ipv6   IPV6 Frame Format
 mpls   MPLS
 other  L2 hdr Frame Format
 pup    PUP Frame Format
 rarp   RARP Frame Format
 valid  On valid packet
```

以下是一些可供您选择使用的其他选项：

```
module-3(fln-l2-elam)# trigger dbus ipv4 ?
 egress                Egress packets

!--- Capture packets in egress (outbound from the port).

 if                    If Trigger Condition
 ingress               Ingress packets

!--- Capture packets in ingress (inbound to the port).

 multicast             Multicast packet
 multicast-replication  Multicast replication
```

在本示例中，**if句柄**用于为捕获选择条件。下一个输出中显示的大多数选项都基于L2、L3和L4报头。源IP地址和目的IP地址也用于捕获。

```
module-3(fln-l2-elam)# trigger dbus ipv4 ingress if ?
 <CR>
 acos                           Acos
 block-capture                  Capture l2 blocks
 bpdu                           Bpdu
 bundle-port                    Bundle-port
 ccc                            Ccc
 copp                           Copp
 da-type                        Da-type
 de-cfi                         De cfi
 destination-index              Destination-index
 destination-ipv4-address       destination ipv4 address
 destination-mac-address        Destination-mac-address
 destination-vif                Destination-vif
 df                             df
 dfst                           Dfst
 dft                            Dft
 disable-index-learn            Disable-index-learn
 disable-new-learn              Disable-new-learn
 dont-forward                   Dont-forward
 dont-learn                     Dont-learn
 dtag-ftag                      Dtag-ftag
 dtag-ttl                       Dtag-ttl
```

dti-type-vpnid	Dti type vpnid
error	Error
erspan-kpa-valid	Erspan kpa valid
ff	Ff
frag	frag
header-type	Header type
ib-length-bundle	Ib length bundle
ids-check-fail	Ids-check-fail
ignore-acli	Ignore-acli
ignore-aclo	Ignore-aclo
ignore-qosi	Ignore-qosi
ignore-qoso	Ignore-qoso
inband-flow-creation-deletion	Inband-flow-creation-deletion
index-direct	Index-direct
inner-cos	Inner-cos
inner-de-valid	Inner de valid
inner-drop-eligibility	Inner-drop-eligibility
ip-da-multicast	Ip-da-multicast
ip-multicast	Ip-multicast
ip-multicast-control	Ip-multicast-control
ipv6	Ipv6
l2	L2
l2-frame-type	L2-frame-type
l2-length-check	L2 length check
l2lu-mode	L2lu-mode
l3-packet-length	l3 packet length
l4-protocol	l4 protocol
label-count	Label count
last-ethertype	Last-ethertype
lbl0-eos	Lbl0 eos
lbl0-exp	Lbl0 exp
lbl0-lbl	Lbl0 lbl
lbl0-ttl	Lbl0 ttl
lbl0-valid	Lbl0 valid
lbl1-exp	Lbl1 exp
lbl1-ttl	Lbl1 ttl
mac-in-mac-valid	Mac-in-mac-valid
mc	Mc
md-acos	Md acos
md-destination-table-index	Md destination table index
md-fwd-only	Md fwd only
md-lif	Md lif
md-mark-enable	Md mark enable
md-multicast-bridge-disable	Md multicast bridge disable
md-preserve-acos	Md preserve acos
md-qos-group-id	Md qos group id
md-replication-packet	Md replication packet
md-router-mac	Md router mac
md-ttl-err	Md-ttl-err
md-version	Md version
mf	mf
mim-destination-mac-address	Mim-destination-mac-address
mim-source-mac-address	Mim-source-mac-address
mlh-type	Mlh-type
no-stats	No-stats
notify-index-learn	Notify-index-learn
notify-new-learn	Notify-new-learn
null-label-exp	Null label exp
null-label-ttl	Null label ttl
null-label-valid	Null label valid
option	option
outer-cos	Outer-cos
outer-drop-eligibility	Outer-drop-eligibility
ovl-mlh-bndl	Ovl mlh bndl

ovl-ulh-bndl	Ovl ulh bndl
ovl-ulh-bndl-1	Ovl-ulh-bndl-1
ovl-ulh-bndl-2	Ovl-ulh-bndl-2
packet-length	Packet-length
packet-type	Packet type
pdt-tag-gt-2	Pdt-tag-gt-2
pdt-tag0	Pdt-tag0
pdt-tag1	Pdt-tag1
pdt-valid	Pdt-valid
pdt-value	Pdt-value
port-id	Port-id
rbh	Rbh
rdt	Rdt
recir-shim-vxlan-src-peer-id	Recir shim vxlan src peer id
recirc-acos	Recirc acos
recirc-bypass-ife	Recirc bypass ife
recirc-bypass-l2	Recirc bypass l2
recirc-destination-table-index	Recirc destination table index
recirc-forward-only	Recirc forward only
recirc-l2-tunnel-encap	Recirc l2 tunnel encap
recirc-lif	Recirc lif
recirc-ls-hash	Recirc ls hash
recirc-mark-enable	Recirc mark enable
recirc-multicast-bridge-disable	Recirc multicast bridge disable
recirc-preserve-acos	Recirc preserve acos
recirc-preserve-ls-hash	Recirc preserve ls hash
recirc-preserve-rbh	Recirc preserve rbh
recirc-qos-group-id	Recirc qos group id
recirc-replication-packet	Recirc replication packet
recirc-router-mac	Recirc router mac
recirc-ttl-err	Recirc ttl err
recirc-valid	Recirc-valid
recirc-version	Recirc version
redirect	Redirect
repl-bypass-ife	Repl bypass ife
repl-bypass-l2	Repl bypass l2
repl-disable-local-bridge	Repl disable local bridge
repl-fwd-only	Repl fwd only
repl-l2-tunnel-encap	Repl l2 tunnel encap
repl-l2-tunnel-info	Repl l2 tunnel info
repl-lif	Repl lif
repl-mark-enable	Repl mark enable
repl-met-lif	Repl met lif
repl-ml3	Repl ml3
repl-preserve-acos	Repl preserve acos
repl-preserve-rbh	Repl preserve rbh
repl-qos-group-id	Repl qos group id
repl-replication-packet	Repl replication packet
repl-router-mac	Repl router mac
repl-ttl-err	Repl ttl err
repl-version	Repl version
rf	Rf
second-inner-cos	Second inner cos
segment-id	Segment id
segment-id-valid	Segment id valid
sequence-number	Sequence-number
sg-tag	Sg-tag
shim-valid	Shim valid
source-index	Source-index
source-ipv4-address	source ipv4 address
source-mac-address	Source-mac-address
source-vif	Source-vif
status-ce-1q	Status-ce-1q
status-is-1q	Status-is-1q

```
 sup-eid                          Sup-eid
 tos                              tos
 traceroute                       Traceroute
 trig                             Any of previous elam triggered
 trill-encap                      Trill-encap
 ttl                              ttl
 tunnel-bundle                    Tunnel bundle
 tunnel-type                      Tunnel type
 ulh-type                         Ulh-type
 valid                            VALID
 vl                               Vl
 vlan                             Vlan
 vn-p                             Vn p
 vn-valid                         Vn-valid
 vqi                              Vqi
 vqi-valid                        Vqi-valid
 vsl-num                          Vsl-num
```

此输出显示最终触发器选项：

```
module-3# elam asic flanker instance 0
module-3(fln-elam)# layer2
module-3(fln-l2-elam)# trigger dbus ipv4 ingress if source-ipv4-address 192.168.12.2
destination-ipv4-address 192.168.13.3
module-3(fln-l2-elam)# trigger rbus ingress if trig
```

注意：RBUS配置通常不复杂且保持简单。

4. 要检查触发器，请输入**status**命令，启动捕获进程，并启动从N7K2到N7K3（192.168.12.1到192.168.13.3）的ping:

```
module-3(fln-l2-elam)# stat
ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if
source-ipv4-address 192.168.12.2 destination-ipv4-address 192.168.13.3
L2 DBUS: Configured
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
L2 RBUS: Configured


module-3(fln-l2-elam)# start
module-3(fln-l2-elam)# status

!--- The status shows as Armed because the process has begun.

ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if
source-ipv4-address 192.168.12.2 destination-ipv4-address 192.168.13.1
L2 DBUS: Armed
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
L2 RBUS: Armed
module-3(fln-l2-elam)#


module-3(fln-l2-elam)# status

!--- If the packet is captured, the status shows Triggered.

ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if
source-ipv4-address 192.168.12.2 destination-ipv4-address 192.168.13.3
L2 DBUS: Triggered
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
L2 RBUS: Triggered
module-3(fln-l2-elam)#
```

5. 如果状态显示**Triggered**，则验证RBUS和DBUS是否具有相同的序列号，以确认它们是否用于同一数据包。在本例中，**使用0x55**，但显示序列号的列不同：

```
module-3(fln-l2-elam)# show dbus | in seq
sequence-number     : 0x6b            vl    : 0x0

!--- The sequence number is the same (0x6b).

module-3(fln-l2-elam)# show rbus | in seq
l2-rbus-trigger     : 0x1             sequence-number     : 0x6b
```

6. 输入**show dbus**和**show rbus**命令以验证DBUS和RBUS。在DBUS命令*输出*中查找源索引，在RBUS命令*输出*中查找目标索引：

```
module-3(fln-l2-elam)# show dbus
cp = 0x1007db4c, buf = 0x1007db4c, end = 0x10089e9c
------------------------------------------------------------------
Flanker Instance 00 - Capture Buffer On L2 DBUS:

Status(0x0102), TriggerWord(0x000), SampleStored(0x005),CaptureBufferPointer(0x005)

is_l2_egress: 0x0000, data_size: 0x023
[000]: 5902a000 08010000 00000000 0cc01400 00145800 00000000 01800100 00000000
00000000 00000000 003931c8 842850b9 31c88428 50c00000 01ac0000 00000000 00000000
00000000 00000000 00000000 00000005 80005000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00605406 01605406 8180008f f0054608 00000000

Printing packet 0
------------------------------------------------------------------
                      L2 DBUS PRS MLH IPV4
------------------------------------------------------------------
label-count         : 0x0             mc                  : 0x0
null-label-valid    : 0x0             null-label-exp      : 0x0
null-label-ttl      : 0x0             lbl0-vld            : 0x0
lbl0-eos            : 0x0             lbl0-lbl            : 0x0
lbl0-exp            : 0x0             lbl0-ttl            : 0x0
lbl1-exp            : 0x0             lbl1-ttl            : 0x0
ipv4                : 0x0             ipv6                : 0x0
l4-protocol         : 0x1             df                  : 0x0
mf                  : 0x0             frag                : 0x0
ttl                 : 0xff            l3-packet-length    : 0x54
option              : 0x0             tos                 : 0x0
sup-eid             : 0x0             header-type         : 0x1
error               : 0x0             redirect            : 0x0
port-id             : 0x0             last-ethertype      : 0x800
l2-frame-type       : 0x0             da-type             : 0x0
packet-type         : 0x0             l2-length-check     : 0x0
ip-da-multicast     : 0x0             ip-multicast        : 0x0
ip-multicast-control: 0x0             ids-check-fail      : 0x0
traceroute          : 0x0             outer-cos           : 0x0
inner-cos           : 0x0             vqi-valid           : 0x0
vqi                 : 0x0             packet-length       : 0x66
vlan                : 0xa             destination-index   : 0x0
source-index        : 0xa2c           bundle-port         : 0x0
acos                : 0x0             outer-drop-eligibility: 0x0
inner-drop-eligibility: 0x0             sg-tag              : 0x0
rbh                 : 0x0             vsl-num             : 0x0
inband-flow-creation-deletion: 0x0             ignore-qoso         : 0x0
ignore-qosi         : 0x0             ignore-aclo         : 0x0
ignore-acli         : 0x0             index-direct        : 0x0
```

```
no-stats            : 0x0            dont-forward        : 0x0
notify-index-learn  : 0x1            notify-new-learn    : 0x1
disable-new-learn   : 0x0            disable-index-learn : 0x0
dont-learn          : 0x0            bpdu                : 0x0
ff                  : 0x0            rf                  : 0x0
ccc                 : 0x0            l2                  : 0x0
rdt                 : 0x0            dft                 : 0x0
dfst                : 0x0            status-ce-1q        : 0x0
status-is-1q        : 0x1            trill-encap         : 0x0
mim-valid           : 0x0            dtag-ttl            : 0x0
dtag-ftag           : 0x0            valid               : 0x1
erspan-kpa-valid    : 0x0            recir-shim-vxlan-src-peer-id: 0x0
vn-valid            : 0x0            source-vif          : 0x0
destination-vif     : 0x0            vn-p                : 0x0
sequence-number     : 0x6b           vl                  : 0x0
inner-de-valid      : 0x0            de-cfi              : 0x0
second-inner-cos    : 0x0            tunnel-type         : 0x0
shim-valid          : 0x0
segment-id-valid    : 0x0            copp                : 0x0
dti-type-vpnid      : 0x0            segment-id          : 0x0
ib-length-bundle    : 0x58000        mlh-type            : 0x5
ulh-type            : 0x6
source-ipv4-address: 192.168.12.2
destination-ipv4-address: 192.168.13.3
mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address : 0000.0000.0000
destination-mac-address : e4c7.2210.a142
source-mac-address : e4c7.2210.a143


module-3(fln-l2-elam)# show rbus
cp = 0x100a2548, buf = 0x100a2548, end = 0x100ae898
------------------------------------------------------------------
Flanker Instance 00 - Capture Buffer On L2 RBUS:

Status(0x0102), TriggerWord(0x000), SampleStored(0x005),CaptureBufferPointer(0x005)

is_l2_egress: 0x0000, data_size: 0x018
[000]: 0059d930 0000000c c0000000 03580000 00000000 00000000 0000001f 57b00021
fdfc0000 00000000 02000000 14001402 8b000105 00000000 68200000 00000000 00000000
00000400 00008000 005b0000 00fe0e4c 7220850a 210000a0 000000b6

Printing packet 0

------------------------------------------------------------------
                      L2 RBUS INGRESS CONTENT
------------------------------------------------------------------
pad                 : 0x16764        valid               : 0x1
l2-rbus-trigger     : 0x1            sequence-number     : 0x6b
rit-ipv4-id         : 0x0            ipv4-tunnel-encap   : 0x0
rit-mpls-rw         : 0x0            ml2-ptr             : 0x0
ml3-ptr             : 0x0            mark                : 0x0
result-cap3         : 0x0            di1-v5-delta-length : 0x0
di1-v5-delta-length-plus: 0x0            di1-v4-delta-length : 0x0
di1-v4-delta-length-plus: 0x0            di2-delta-length    : 0x0
di2-delta-length-plus: 0x0            ml2-delta-length    : 0x0
ml2-delta-length-plus: 0x0            ml3-delta-length    : 0x0
ml3-delta-length-plus: 0x0            s-vector            : 0x0
lcpu-ff-valid       : 0x0            sup-di-vqi          : 0x0
erspan-term-index-dir: 0x0            erspan-buffer-check : 0x0
l2-tunnel-decapped  : 0x0            l3-delta-length     : 0x0
rit-crc16-valid     : 0x1            rit-crc16           : 0xf57b
vntag-p             : 0x0            frr-recirc          : 0x0
ingress-lif         : 0x1            earl-proxy-vld      : 0x0
```

```
md-di-vld           : 0x0            rc                  : 0x0
segment-id-valid    : 0x0            ttl-out             : 0xfe
ttl-mid             : 0xfe           tos-out             : 0x0
tos-in              : 0x0            orig-vlan1          : 0x0
vlan1               : 0x0            source-peer-id      : 0x0
final-ignore-qoso   : 0x0            port-id             : 0x0
cr-type             : 0x1            pup-packet          : 0x0
bpdu                : 0x0            vdc                 : 0x0
traceroute          : 0x0            de                  : 0x0
cos                 : 0x0            inner-drop-eligibility: 0x0
inner-cos           : 0x0            acos                : 0x0
di-ltl-index        : 0x50           l3-multicast-di     : 0x50
source-index        : 0xa2c          vlan                : 0x0
index-direct        : 0x0            di1-valid           : 0x1
vqi                 : 0x50           di2-valid           : 0x0
v5-fpoe-idx         : 0x0            di2-fpoe-idx        : 0x0
l3-multicast-v5     : 0x0            dft                 : 0x0
dfst                : 0x0            l3-learning-ff      : 0x0
result-rbh          : 0xd0           di2-cr-type         : 0x0
result-2            : 0x1            dtag-ftag           : 0x0
dtag-ttl            : 0x20           mac-in-mac-op       : 0x0
dvif                : 0x0            result-cap1         : 0x0
result-cap2         : 0x0            erspan-term         : 0x0
erspan-decap        : 0x0            dont-learn          : 0x0
routed-frame        : 0x1            copy-cause          : 0x0
l2-copy-cause       : 0x0            l3-rit-ptr          : 0x5b
sg-tag              : 0x0            trill-nh-id         : 0x0
ttl-in              : 0xfe           fc-up               : 0x0
up-did              : 0x0            did                 : 0xe4c722
up-sid              : 0x0            sid                 : 0x10a144
shim-l2-tunnel-encap: 0x0            shim-ls-hash        : 0x8
shim-rc             : 0x0            shim-lif            : 0x1
shim-replication-pkt: 0x0            shim-router-mac     : 0x1
shim-mark-enable    : 0x0            shim-qos-group-id   : 0x0
shim-destination-table-index: 0x5b           shim-acos-preserve  : 0x0
mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address : 0000.0000.0000
```

7. 在Sup:

```
N7K1# show system  internal pixm info ltl 0xa2c

PC_TYPE    PORT    LTL      RES_ID       LTL_FLAG      CB_FLAG      MEMB_CNT
------------------------------------------------------------------------------
Normal    Po1     0x0a2c    0x16000000   0x00000000    0x00000002   2

Member rbh rbh_cnt
Eth3/2   0x000000f0   0x04
Eth3/1   0x0000000f   0x04

CBL Check States: Ingress: Enabled; Egress: Enabled

VLAN| BD| BD-St              | CBL St & Direction:
--------------------------------------------------
1 | 0x15 | INCLUDE_IF_IN_BD  | FORWARDING (Both)
10 | 0x19 | INCLUDE_IF_IN_BD  | FORWARDING (Both)


Member info
------------------
Type            LTL
---------------------------------
PORT_CHANNEL   Po1
```

```
FLOOD_W_FPOE   0x8019
FLOOD_W_FPOE   0x8015


N7K1# show system  internal pixm info ltl 0x50
0x0050 is in DCE/FC pool

Member info
------------------
Type            LTL
---------------------------------
PHY_PORT      Eth3/5
```

此输出确认数据包是在端口通道1(Po1)上接收并通过Eth3/5转发的。

8. 验证模块上的本地目标逻辑(LTL)以获取正确的编程：

```
 module-3# show system  internal pixmc info ltl-cb ltl 0xa2c
 ltl |ltl_type|if_index|lc_type| vdc |v4_fpoe|v5_fpoe| base_fpoe_idx | flag
0x0a2c | 4    | Po1    |   2   |  2  | 0x00  |  0x00 |    0x0000     | 0x0
, local ports:
VDCs the entry is part of:

LTL HW programming info
.....................
-----------------------------------------------------
|Index | ec |drop|span_vec|SOM|ucr_fab|
|-----------------------------------------------------

|[ a2c]|   1|   0|      0|  0|      0|
|  RBH    |   VQI |   PS(INST:LPOE)
|---------------------------------
     0,       40      0 :  1
     1,       40      0 :  1
     2,       40      0 :  1
     3,       40      0 :  1
     4,       44      0 : 10
     5,       44      0 : 10
     6,       44      0 : 10
     7,       44      0 : 10
     8,        0      0 :  1
     9,        0      0 :  1
     a,        0      0 :  1
     b,        0      0 :  1
     c,        0      0 : 10
     d,        0      0 : 10
     e,        0      0 : 10
     f,        0      0 : 10


module-3# show system  internal pixmc info ltl-cb ltl 0x50
ltl |ltl_type|if_index|lc_type| vdc |v4_fpoe | v5_fpoe| base_fpoe_idx | flag
0x0050 |  5  |Eth3/5  |   2   |  2  |  0x00  |  0x00  |    0x0000     | 0x0
, local ports:
VDCs the entry is part of:

LTL HW programming info
.....................
-----------------------------------------------------
|Index | ec |drop|span_vec|SOM|ucr_fab|
|-----------------------------------------------------
```

```
|[  50]|   1|   0|      0|  0|      0|
|  RBH    |  VQI |       PS
|---------------------------
ALL RBH|      50 |     2  :    1
```

9. 在出口时捕获ELAM数据包。要捕获数据包，请从IP地址192.168.13.3发送ping应答到192.168.12.2。必须在端口通道1接口(e3/1-2)上使用**egress**关键字设置捕获。 如前所述，接口属于实例0。

```
N7K1# att mo 3
Attaching to module 3 ...
To exit type 'exit', to abort type '$.'
module-3# el asic flanker instance 0
module-3(fln-elam)# layer2
module-3(fln-l2-elam)# trigger dbus ipv4 egress if source-ipv4-address 192.168.13.3
destination-ipv4-address 192.168.12.2
module-3(fln-l2-elam)# trigger rbus egress if trig


module-3(fln-l2-elam)# status
ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.13.3 destination-ipv4-address 192.168.12.2
L2 DBUS: Configured
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus egress if trig
L2 RBUS: Configured


module-3(fln-l2-elam)# start
module-3(fln-l2-elam)# status
ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.13.3 destination-ipv4-address 192.168.12.2
L2 DBUS: Armed
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus egress if trig
L2 RBUS: Armed


module-3(fln-l2-elam)# status
ELAM Slot 3 instance 0: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.13.3 destination-ipv4-address 192.168.12.2
L2 DBUS: Triggered
ELAM Slot 3 instance 0: L2 RBUS Configuration: trigger rbus egress if trig
L2 RBUS: Triggered
module-3(fln-l2-elam)#


module-3(fln-l2-elam)# show dbus | in seq
sequence-number     : 0x8d            vl      : 0x3

!--- The sequence number is the same.

module-3(fln-l2-elam)# show rbus | in seq
vl                  : 0x0       sequence-number     : 0x8d


module-3(fln-l2-elam)# show dbus
cp = 0x1007db4c, buf = 0x1007db4c, end = 0x10089e9c
----------------------------------------------------------------
Flanker Instance 00 - Capture Buffer On L2 DBUS:

Status(0x0102), TriggerWord(0x000), SampleStored(0x005),CaptureBufferPointer(0x005)
```

```
is_l2_egress: 0x0000, data_size: 0x023
[000]: 48c22000 08210000 40020800 0cc01414 5800a000 00001a40 01030000 00000000
00000000 00000000 003931c8 842850f9 31c88428 50800000 02358000 00000000 00000000
00000000 00000000 00000000 00000000 00005000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00605406 81e05406 0100008f e0054600 00000000

Printing packet 0

------------------------------------------------------------------------
                      L2 DBUS PRS MLH IPV4
------------------------------------------------------------------------
label-count         : 0x0             mc                  : 0x0
null-label-valid    : 0x0             null-label-exp      : 0x0
null-label-ttl      : 0x0             lbl0-vld            : 0x0
lbl0-eos            : 0x0             lbl0-lbl            : 0x0
lbl0-exp            : 0x0             lbl0-ttl            : 0x0
lbl1-exp            : 0x0             lbl1-ttl            : 0x0
ipv4                : 0x0             ipv6                : 0x0
l4-protocol         : 0x1             df                  : 0x0
mf                  : 0x0             frag                : 0x0
ttl                 : 0xfe            l3-packet-length    : 0x54
option              : 0x0             tos                 : 0x0
sup-eid             : 0x0             header-type         : 0x1
error               : 0x0             redirect            : 0x0
port-id             : 0x1             last-ethertype      : 0x800
l2-frame-type       : 0x0             da-type             : 0x0
packet-type         : 0x1             l2-length-check     : 0x0
ip-da-multicast     : 0x0             ip-multicast        : 0x0
ip-multicast-control: 0x0             ids-check-fail      : 0x0
traceroute          : 0x0             outer-cos           : 0x0
inner-cos           : 0x0             vqi-valid           : 0x1
vqi                 : 0x40            packet-length       : 0x66
vlan                : 0xa             destination-index   : 0xa2c
source-index        : 0x50            bundle-port         : 0x0
acos                : 0x0             outer-drop-eligibility: 0x0
inner-drop-eligibility: 0x0             sg-tag              : 0x0
rbh                 : 0xd2            vsl-num             : 0x0
inband-flow-creation-deletion: 0x0             ignore-qoso         : 0x0
ignore-qosi         : 0x0             ignore-aclo         : 0x0
ignore-acli         : 0x0             index-direct        : 0x0
no-stats            : 0x0             dont-forward        : 0x0
notify-index-learn  : 0x1             notify-new-learn    : 0x0
disable-new-learn   : 0x0             disable-index-learn : 0x0
dont-learn          : 0x0             bpdu                : 0x0
ff                  : 0x0             rf                  : 0x1
ccc                 : 0x4             l2                  : 0x0
rdt                 : 0x0             dft                 : 0x0
dfst                : 0x0             status-ce-1q        : 0x0
status-is-1q        : 0x0             trill-encap         : 0x0
mim-valid           : 0x0             dtag-ttl            : 0x0
dtag-ftag           : 0x0             valid               : 0x1
erspan-kpa-valid    : 0x0             recir-shim-vxlan-src-peer-id: 0x0
vn-valid            : 0x0             source-vif          : 0x0
destination-vif     : 0x0             vn-p                : 0x0
sequence-number     : 0x8d            vl                  : 0x3
inner-de-valid      : 0x0             de-cfi              : 0x0
second-inner-cos    : 0x0             tunnel-type         : 0x0
shim-valid          : 0x0
segment-id-valid    : 0x0             copp                : 0x0
dti-type-vpnid      : 0x0             segment-id          : 0x0
ib-length-bundle    : 0x0             mlh-type            : 0x5
ulh-type            : 0x6
source-ipv4-address: 192.168.13.3
destination-ipv4-address: 192.168.12.2
```

```
mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address : 0000.0000.0000
destination-mac-address : e4c7.2210.a143
source-mac-address : e4c7.2210.a142
```

如图所示，源索引和目标索引都是DBUS的一部分（与入口捕获中显示的不同）。

```
module-3(fln-l2-elam)# show rbus
cp = 0x100a2548, buf = 0x100a2548, end = 0x100ae898
------------------------------------------------------------------
Flanker Instance 00 - Capture Buffer On L2 RBUS:

Status(0x1102), TriggerWord(0x000), SampleStored(0x008),CaptureBufferPointer(0x000)

is_l2_egress: 0x0001, data_size: 0x018
[000]: 0048ea00 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 0c000000 00000000 04014008 00005000 00000000
00000726 3910850a 1b931c88 42850800 00000000 00000000 0000008d

Printing packet 0

------------------------------------------------------------------
                      L2 RBUS EGRESS CONTENT
------------------------------------------------------------------
pad                 : 0x0            valid               : 0x1
trig                : 0x1            reserved            : 0x0
vn-tag-p            : 0x0            cbl-vlan-valid      : 0x0
vft-hop-count       : 0x0            vft-vsan            : 0x0
vft-up              : 0x0            vft-valid           : 0x0
copp                : 0x0            segment-id-valid    : 0x0
segment-id-23       : 0x0            vsl-num             : 0x0
inner-cos           : 0x0            inner-drop-eligibility: 0x0
cos                 : 0x0            drop-eligibility    : 0x0
dce-mode            : 0x0            flood-to-bd         : 0x0
pt-bit-en           : 0x1            cpu-port            : 0x0
vlan-id             : 0xa            ip-tos              : 0x0
result-rbh          : 0x1            met-ptr             : 0x2000
packet-type         : 0x1            sg-tag              : 0x0
dtag-ftag           : 0x0            vdc                 : 0x0
vn-tag-src-vif      : 0x0            vn-tag-dst-vif      : 0x0
vn-tag-l            : 0x0            dc3-tr              : 0x0
vl                  : 0x0            sequence-number     : 0x8d
destination-mac-valid: 0x0
source-mac-valid: 0x0
mim-destination-mac-address : 0000.0000.0000
destination-mac-address : e4c7.2210.a143
source-mac-address : e4c7.2210.a142
mim-source-mac-address : 0000.0000.0000
```

源IP地址和目的IP地址正确，在入口ELAM数据包捕获后解码；但是，与入口ELAM相比，方向肯定相反，因为返回流量被捕获。

10. 检查N7K1上端口通道1模块3的基于颜色的逻辑(CBL)，以确定VLAN 10是否通过它转发流量。CBL是基于每个物理接口的逻辑，因此您应输入N7K1上端口通道1的成员接口编号，而不是端口通道编号。在下一个输出中，您可以看到VLAN 10按预期转发它。

    CBL用于确定硬件中端口的生成树协议(STP)状态。当您检查SP上VLAN的STP时，接口可能显示转发，但模块会阻止流量。 **注意：**必须分别检查两个成员接口（e3/1和e3/2）的CBL。

    ```
    module-3# show hardware internal mac  port 1 table cbl vlan
     ------------------------------------------------------------------------------
     |                                  INGRESS                                   |
    ```

```
| Disabled State   | 0,2-9,11-4031,4036-4095                                      |
| Forwarding State | 1,10,4032-4035                                               |
| Blocked State    |                                                              |
| Learning State   |                                                              |
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
|                                EGRESS                                       |
| Disabled State   | 0,2-9,11-4031,4036-4095                                      |
| Forwarding State | 1,10,4032-4035                                               |
| Blocked State    |                                                              |
| Learning State   |                                                              |
-------------------------------------------------------------------------------
```

注意：以前的命令用于端口通道1(模块3在**e3/1**上)。

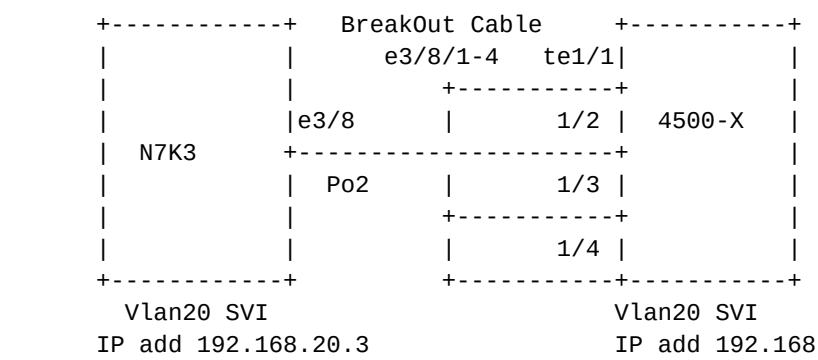
```
module-3# show hardware internal mac  port 2 table cbl vlan
-------------------------------------------------------------------------------
|                                INGRESS                                      |
| Disabled State   | 0,2-9,11-4031,4036-4095                                      |
| Forwarding State | 1,10,4032-4035                                               |
| Blocked State    |                                                              |
| Learning State   |                                                              |
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
|                                EGRESS                                       |
| Disabled State   | 0,2-9,11-4031,4036-4095                                      |
| Forwarding State | 1,10,4032-4035                                               |
| Blocked State    |                                                              |
| Learning State   |                                                              |
-------------------------------------------------------------------------------
```

注意：同样，此命令会检查端口通道2(**e3/2**)的CBL。

使用分支电缆对F3系列模块上的ELAM进行故障排除

连接分支电缆时F3系列模块的ELAM程序与常规模块端口上的ELAM程序无异。但是，在尝试将索引转换为前面板编号时，端口索引管理器(PIXM)的验证有一些变化，在这种情况下，接口会从分支电缆接收。

以下是本节中用于示例的拓扑：



在本例中，分支电缆连接到以太网接口*3/8*，这将40 Gb端口分为四个10 Gb端口。本部分提供了所需的配置以供参考。

```
N7K3(config)# interface breakout module 3 port 8 map 10g-4x
```

```
N7K3(config)# show interface brief
---SNIP---
```

```
--------------------------------------------------------------
Ethernet      VLAN    Type Mode   Status  Reason  Speed   Port
Interface                                          Ch #
--------------------------------------------------------------
Eth3/7        --      eth  routed up      none    40G(D) --
Eth3/8/1      1       eth  trunk  up      none    10G(D) 2

!--- From 3/8/1 to 3/8/4.

Eth3/8/2      1       eth  trunk  up      none    10G(D) 2
Eth3/8/3      1       eth  trunk  up      none    10G(D) 2
Eth3/8/4      1       eth  trunk  up      none    10G(D) 2
```

在上一输出中，您可以看到以太网接*口3/7*仍是一个40 Gb端口；但是，以太网*接口3/*8现在分为四个万兆端口，可单独配置：

```
N7K3# show run interface e3/8/1 - 4

!Command: show running-config interface Ethernet3/8/1-4
!Time: Mon May  4 01:46:28 2015

version 6.2(8a)

interface Ethernet3/8/1
 switchport
 switchport mode trunk
 switchport trunk allowed vlan 10,20
 no shutdown

interface Ethernet3/8/2
 switchport
 switchport mode trunk
 switchport trunk allowed vlan 30,40
 no shutdown

interface Ethernet3/8/3
 switchport
 switchport mode trunk
 switchport trunk allowed vlan 50
 no shutdown

interface Ethernet3/8/4
 switchport
 switchport mode trunk
 no shutdown
```

从N7K3交换虚拟接口(SVI)20 IP地址(192.168.20.3)开始数据包捕获到4500 SVI 20 IP地址(192.168.20.1)。 数据包将在N7K3上从4500出口捕获，应答从4500发送到N7K3。

如上节所述，您必须了解flanker实例才能应用触发器。此输出显示模块3的附件：

```
N7K3# attach module 3
Attaching to module 3 ...
To exit type 'exit', to abort type '$.'


module-3# show hardware internal dev
dev-port-map   dev-version
module-3# show hardware internal dev-port-map
--------------------------------------------------------------
```

```
CARD_TYPE:       12 port 40G
>Front Panel ports:12
--------------------------------------------------------------
 Device name             Dev role              Abbr num_inst:
--------------------------------------------------------------
>Flanker Eth Mac Driver DEV_ETHERNET_MAC       MAC_0  6
>Flanker Fwd Driver     DEV_LAYER_2_LOOKUP     L2LKP  6
>Flanker Xbar Driver    DEV_XBAR_INTF          XBAR_INTF 6
>Flanker Queue Driver   DEV_QUEUEING          QUEUE  6
>Sacramento Xbar ASIC   DEV_SWITCH_FABRIC      SWICHF 1
>Flanker L3 Driver      DEV_LAYER_3_LOOKUP     L3LKP  6
>EDC                    DEV_PHY                PHYS   2
+---------------------------------------------------------------------------+
+----------------+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++------------+
+---------------------------------------------------------------------------+
FP port |  PHYS | MAC_0 | L2LKP | L3LKP | QUEUE |SWICHF
  1               0       0       0       0       0
  2               0       0       0       0       0
  3               1       1       1       1       0
  4               1       1       1       1       0
  5       0       2       2       2       2       0
  6       0       2       2       2       2       0
  7       1       3       3       3       3       0
  8       1       3       3       3       3       0

!--- The port 8 L2LKP column shows a value of 3.

  9               4       4       4       4       0
  10              4       4       4       4       0
  11              5       5       5       5       0
  12              5       5       5       5       0
+---------------------------------------------------------------------------+
+---------------------------------------------------------------------------+
```

在此输出中，端口8位于Flanker实例*3*上。现在您知道该实例，可以通过源和目标IP地址放置触发器。由于您将捕获从N7K3到4500的ping请求，因此它将是出口ELAM。

```
module-3# elam asic flanker instance 3
module-3(fln-elam)# layer2
module-3(fln-l2-elam)# trigger dbus  ipv4 egress if source-ipv4-address 192.168.20.3
destination-ipv4-address 192.168.20.1
module-3(fln-l2-elam)# trigger  rbus egress if trig


module-3(fln-l2-elam)# status
ELAM Slot 3 instance 3: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.20.3 destination-ipv4-address 192.168.20.1
L2 DBUS: Configured
ELAM Slot 3 instance 3: L2 RBUS Configuration: trigger rbus egress if trig
L2 RBUS: Configured


module-3(fln-l2-elam)# start
module-3(fln-l2-elam)# status
ELAM Slot 3 instance 3: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.20.3 destination-ipv4-address 192.168.20.1
L2 DBUS: Armed
ELAM Slot 3 instance 3: L2 RBUS Configuration: trigger rbus egress if trig
L2 RBUS: Armed
```

从N7K3向4500发起ping:

```
N7K3# ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1): 56 data bytes
36 bytes from 192.168.20.3: Destination Host Unreachable
Request 0 timed out
64 bytes from 192.168.20.1: icmp_seq=1 ttl=254 time=6.49 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=254 time=6.518 ms
64 bytes from 192.168.20.1: icmp_seq=3 ttl=254 time=7.936 ms
64 bytes from 192.168.20.1: icmp_seq=4 ttl=254 time=7.945 ms

--- 192.168.20.1 ping statistics ---
5 packets transmitted, 4 packets received, 20.00% packet loss
round-trip min/avg/max = 6.49/7.222/7.945 ms
```

以下是ELAM状态：

```
module-3(fln-l2-elam)# status
ELAM Slot 3 instance 3: L2 DBUS Configuration: trigger dbus ipv4 egress if
source-ipv4-address 192.168.20.3 destination-ipv4-address 192.168.20.1
L2 DBUS: Triggered
ELAM Slot 3 instance 3: L2 RBUS Configuration: trigger rbus egress if trig
L2 RBUS: Triggered
```

验证序列号是否相同：

```
module-3(fln-l2-elam)# show dbus | in seq
sequence-number     : 0x27            vl                  : 0x3
module-3(fln-l2-elam)# show rbus | in seq
vl                  : 0x0             sequence-number     : 0x27
```

序列号相同。现在，您可以检查DBUS和RBUS信息：

```
module-3(fln-l2-elam)# show dbus
cp = 0x1011033c, buf = 0x1011033c, end = 0x1011c68c
----------------------------------------------------------------------
Flanker Instance 03 - Capture Buffer On L2 DBUS:

Status(0x0102), TriggerWord(0x000), SampleStored(0x004),CaptureBufferPointer(0x004)

is_l2_egress: 0x0000, data_size: 0x023
[000]: 4c1ea000 20a10000 40021040 0cc02801 04080000 00000000 08100000 00000000
00000000 00000000 003c1fc1 8732dff9 31c88428 51000000 009d8000 00000000 00000000
00000000 00000000 00000000 00000000 00005000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 0060540a 01e0540a 0080008f f0054608 00000000

Printing packet 0

----------------------------------------------------------------------
                      L2 DBUS PRS MLH IPV4
----------------------------------------------------------------------
label-count         : 0x0             mc                  : 0x0
null-label-valid    : 0x0             null-label-exp      : 0x0
null-label-ttl      : 0x0             lbl0-vld            : 0x0
lbl0-eos            : 0x0             lbl0-lbl            : 0x0
lbl0-exp            : 0x0             lbl0-ttl            : 0x0
lbl1-exp            : 0x0             lbl1-ttl            : 0x0
ipv4                : 0x0             ipv6                : 0x0
l4-protocol         : 0x1             df                  : 0x0
mf                  : 0x0             frag                : 0x0
ttl                 : 0xff            l3-packet-length    : 0x54
option              : 0x0             tos                 : 0x0
sup-eid             : 0x1             header-type         : 0x0
```

```
error               : 0x0            redirect            : 0x0
port-id             : 0x5            last-ethertype      : 0x800
l2-frame-type       : 0x0            da-type             : 0x0
packet-type         : 0x1            l2-length-check     : 0x0
ip-da-multicast     : 0x0            ip-multicast        : 0x0
ip-multicast-control: 0x0            ids-check-fail      : 0x0
traceroute          : 0x0            outer-cos           : 0x0
inner-cos           : 0x0            vqi-valid           : 0x1
vqi                 : 0x82           packet-length       : 0x66
vlan                : 0x14           destination-index   : 0x82
source-index        : 0x400          bundle-port         : 0x0
acos                : 0x0            outer-drop-eligibility: 0x0
inner-drop-eligibility: 0x0            sg-tag              : 0x0
rbh                 : 0x0            vsl-num             : 0x0
inband-flow-creation-deletion: 0x0            ignore-qoso         : 0x0
ignore-qosi         : 0x0            ignore-aclo         : 0x0
ignore-acli         : 0x0            index-direct        : 0x1
no-stats            : 0x0            dont-forward        : 0x0
notify-index-learn  : 0x0            notify-new-learn    : 0x0
disable-new-learn   : 0x0            disable-index-learn : 0x0
dont-learn          : 0x1            bpdu                : 0x0
ff                  : 0x0            rf                  : 0x0
ccc                 : 0x0            l2                  : 0x0
rdt                 : 0x0            dft                 : 0x0
dfst                : 0x0            status-ce-1q        : 0x0
status-is-1q        : 0x0            trill-encap         : 0x0
mim-valid           : 0x0            dtag-ttl            : 0x0
dtag-ftag           : 0x0            valid               : 0x1
erspan-kpa-valid    : 0x0            recir-shim-vxlan-src-peer-id: 0x0
vn-valid            : 0x0            source-vif          : 0x0
destination-vif     : 0x0            vn-p                : 0x0
sequence-number     : 0x27           vl                  : 0x3
inner-de-valid      : 0x0            de-cfi              : 0x0
second-inner-cos    : 0x0            tunnel-type         : 0x0
shim-valid          : 0x0
segment-id-valid    : 0x0            copp                : 0x0
dti-type-vpnid      : 0x0            segment-id          : 0x0
ib-length-bundle    : 0x0            mlh-type            : 0x5
ulh-type            : 0x6
source-ipv4-address: 192.168.20.3
destination-ipv4-address: 192.168.20.1
mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address : 0000.0000.0000
destination-mac-address : f07f.061c.cb7f
source-mac-address : e4c7.2210.a144


module-3(fln-l2-elam)#
module-3(fln-l2-elam)#
module-3(fln-l2-elam)#
module-3(fln-l2-elam)# show rbus
cp = 0x10134d38, buf = 0x10134d38, end = 0x10141088
------------------------------------------------------------------
Flanker Instance 03 - Capture Buffer On L2 RBUS:

Status(0x1102), TriggerWord(0x000), SampleStored(0x008),CaptureBufferPointer(0x000)

is_l2_egress: 0x0001, data_size: 0x018
[000]: 004c4780 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 0c001000 00000000 80028010 00009000 00000000
00000783 f830e65b fb931c88 42851000 00000000 00000000 00000027

Printing packet 0
```

```
------------------------------------------------------------------
                     L2 RBUS EGRESS CONTENT
------------------------------------------------------------------
pad                 : 0x0             valid               : 0x1
trig                : 0x1             reserved            : 0x0
vn-tag-p            : 0x1             cbl-vlan-valid      : 0x0
vft-hop-count       : 0x0             vft-vsan            : 0x0
vft-up              : 0x0             vft-valid           : 0x0
copp                : 0x0             segment-id-valid    : 0x0
segment-id-23       : 0x0             vsl-num             : 0x0
inner-cos           : 0x0             inner-drop-eligibility: 0x0
cos                 : 0x0             drop-eligibility    : 0x0
dce-mode            : 0x0             flood-to-bd         : 0x0
pt-bit-en           : 0x20            cpu-port            : 0x0
vlan-id             : 0x14            ip-tos              : 0x0
result-rbh          : 0x2             met-ptr             : 0x4000
packet-type         : 0x1             sg-tag              : 0x0
dtag-ftag           : 0x0             vdc                 : 0x0
vn-tag-src-vif      : 0x0             vn-tag-dst-vif      : 0x0
vn-tag-l            : 0x0             dc3-tr              : 0x0
vl                  : 0x0             sequence-number     : 0x27
destination-mac-valid: 0x0
source-mac-valid: 0x0
mim-destination-mac-address : 0000.0000.0000
destination-mac-address : f07f.061c.cb7f
source-mac-address : e4c7.2210.a144
mim-source-mac-address : 0000.0000.0000
```

将目标索引和源索引转换为前面板端口以确认流：

```
N7K3# show system  internal pixm info ltl 0x400
0x0400 is in SUP In-band LTL range
```

此输出显示源索引。您知道这是正确的，因为从Sup到N7K3的ping操作。下一个输出显示出出口接口(e3/8/1)，它是N7K上允许VLAN 20的两个接口之一。另一个接口是*e3/8/4*，由于STP而在4500上被阻塞。

```
N7K3# show system  internal pixm info ltl 0x82
0x0082 is in DCE/FC pool

Member info
------------------
Type            LTL
--------------------------------
PHY_PORT       Eth3/8/1
FLOOD_W_FPOE   0x8039
FLOOD_W_FPOE   0x803f
```

验证N7K上使用分支电缆创建的端口的电缆。要检查CBL，您必须拥有所有新形成端口的硬件端口号。

注意：交换机上不存在接口e3/8。仅显示新形成的端口。

```
N7K3# show  interface e3/8
                        ^
% Incomplete command at '^' marker.
N7K3#
```

由于使用了分支电缆，并且交换机上不存在e3/8接口，因此用于获取硬件端口号的计算会发生变化。对于任何支持分支的模块，硬件端口编号都不同。您应首先检查端口是否支持分支：

```
N7K3# show int e3/7 capabilities
Ethernet3/7
 Model:                 N7K-F312FQ-25
 Type (SFP capable):    QSFP-40G-CR4
 Speed:                 10000,40000
 Duplex:                full
 ---SNIP---
 PFC capable:           yes
  Breakout capable:      yes
```

如图所示，端*口e3/7*支持分支，这意味着其带宽可分为四个万兆端口。同样，其他具有100千兆端口的F3系列模块可分为10个每个10千兆端口或3个40千兆超订用端口。具体情况取决于模块。

由于本示例中的F3系列模块有40千兆端口，并且每个端口可分为四个端口，因此每个端口的硬件端口号是0-3、4-7、8-11...40-43、44-47（以零为基数）。如果第一个示例的端口上有分支电缆，则其硬件端口编号为0、1、2和3。如果没有分支电缆，则其硬件端口编号为0（1、2和3将不活动）。

由于父端口是e3/8，如果不使用分支电缆，则其硬件端口号为28；如果与分支电缆一起使用，则其硬件端口号为28、29、30和31。此命令输出显示活动硬件端口（基于零）：

```
N7K3# show system internal ifindex info mod 3

Init DB dump follows:
module_num_bitmask = 0x3ffff
Slot:3, Proc:1, breakout_factor:0, sw_card_id:0, active_cfg_ports:, broken_fp_po
rts:
Slot:3, Proc:2, breakout_factor:4, sw_card_id:155, active_cfg_ports:0,4,8,12,16,
20,24,28-32,36,40,44, broken_fp_ports:28

Lookup DB dump follows:
Slot:3, breakout_factor:4
```

损坏的端口硬件端口号为28，现在分为四个(28-32)。 现在，您可以连接模块3并检查硬件中的电缆：

```
N7K3# attach  module  3
Attaching to module 3 ...
To exit type 'exit', to abort type '$.'
module-3#
```

F3系列模块希望端口号按照基于一的比例进行格式化。因此，您应输入29、30、31和32:

```
module-3# show  hardware  internal mac  port  ?
 <1-96>  Port number (1-based)

!--- This is context sensitive, so it helps to say the port number is 1-based.
```

以下是以太网接口3/8/*1的运行配*置，以便检查和确认VLAN转发状态：

```
interface Ethernet3/8/1
switchport
switchport mode trunk
switchport trunk allowed vlan 10,20
no shutdown

module-3# show  hardware  internal mac  port 29  table  cbl vlan
 --------------------------------------------------------------------------------
 |                                 INGRESS                                      |
```

```
 | Disabled State   | 0,2-9,11-19,21-4031,4036-4095                              |
 | Forwarding State | 10,20,4032-4035                                            |
 | Blocked State    | 1                                                          |
 | Learning State   |                                                            |
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 |                                EGRESS                                     |
 | Disabled State   | 0,2-9,11-19,21-4031,4036-4095                              |
 | Forwarding State | 10,20,4032-4035                                            |
 | Blocked State    | 1                                                          |
 | Learning State   |                                                            |
 ---------------------------------------------------------------------------
```

以下是以太网接口3/8/*2的运行配*置，以便检查和确认VLAN转发状态：

```
interface Ethernet3/8/2
switchport
switchport mode trunk
switchport trunk allowed vlan 30,40
no shutdown

module-3# show  hardware  internal mac  port 30  table  cbl vlan
 ---------------------------------------------------------------------------
 |                                INGRESS                                    |
 | Disabled State   | 0,2-29,31-39,41-4031,4036-4095                             |
 | Forwarding State | 30,40,4032-4035                                            |
 | Blocked State    | 1                                                          |
 | Learning State   |                                                            |
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 |                                EGRESS                                     |
 | Disabled State   | 0,2-29,31-39,41-4031,4036-4095                             |
 | Forwarding State | 30,40,4032-4035                                            |
 | Blocked State    | 1                                                          |
 | Learning State   |                                                            |
 ---------------------------------------------------------------------------
```

以下是以太网接口3/8/*3的运行配*置，以便检查和确认VLAN转发状态：

```
interface Ethernet3/8/3
switchport
switchport mode trunk
switchport trunk allowed vlan 50
no shutdown

module-3# show  hardware  internal mac  port 31  table  cbl vlan
 ---------------------------------------------------------------------------
 |                                INGRESS                                    |
 | Disabled State   | 0,2-49,51-4031,4036-4095                                   |
 | Forwarding State | 50,4032-4035                                               |
 | Blocked State    | 1                                                          |
 | Learning State   |                                                            |
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
 |                                EGRESS                                     |
 | Disabled State   | 0,2-49,51-4031,4036-4095                                   |
 | Forwarding State | 50,4032-4035                                               |
 | Blocked State    | 1                                                          |
 | Learning State   |                                                            |
 ---------------------------------------------------------------------------
```

以下是以太网接口3/*8/4的运*行配置，用于检查和确认VLAN转发状态（允许所有已配置的VLAN）：

```
interface Ethernet3/8/4
switchport
switchport mode trunk
no shutdown
```

```
module-3# show  hardware  internal mac  port 32  table  cbl vlan
 ------------------------------------------------------------------------------
 |                                  INGRESS                                   |
 | Disabled State   | 0,2-9,11-19,21-29,31-39,41-49,51-59,61-669,671-4031       |
 | Disabled State   | 4036-4095                                                 |
 | Forwarding State | 1,20,30,40,50,60,670,4032-4035                            |
 | Blocked State    | 10                                                        |
 | Learning State   |                                                           |
 ------------------------------------------------------------------------------
 ------------------------------------------------------------------------------
 |                                  EGRESS                                    |
 | Disabled State   | 0,2-9,11-19,21-29,31-39,41-49,51-59,61-669,671-4031       |
 | Disabled State   | 4036-4095                                                 |
 | Forwarding State | 1,20,30,40,50,60,670,4032-4035                            |
 | Blocked State    | 10                                                        |
 | Learning State   |                                                           |
 ------------------------------------------------------------------------------
```

CBL显示转发了正确的VLAN。

可以使用show hardware internal error module <module number>命令来获取硬件端口号。当您必须检查show interface x/y命令输出中未显示的任何内部丢**弃时，此命令**非常有用。示例如下：

```
N7K2# show hardware internal errors module 3
---SNIP---
Instance:1
Cntr  Name                                          Value             Ports
----- ----                                          -----             -----
 3836 igr rx pl: cbl drops                          0000000000000001  10 -
 4636 igr rx pl: cbl drops                          0000000000000001  14 -


Instance:2
Cntr  Name                                          Value             Ports
----- ----                                          -----             -----
 423 igr in upm: pkts with symbol/sequence error rcvd 0000000000000478  18 -
 455 igr in upm: pkts with symbol/sequence error rcvd 0000000000000478  17 -
 487 igr in upm: pkts with symbol/sequence error rcvd 0000000000000478  19 -
 519 igr in upm: pkts with symbol/sequence error rcvd 0000000000000478  20 -


Instance:3
Cntr  Name                                          Value             Ports
----- ----                                          -----             -----
 423 igr in upm: pkts with symbol/sequence error rcvd 0000000000000745  26 -
 455 igr in upm: pkts with symbol/sequence error rcvd 0000000000000745  25 -
 487 igr in upm: pkts with symbol/sequence error rcvd 0000000000000745  27 -
 519 igr in upm: pkts with symbol/sequence error rcvd 0000000000000745  28 -
 550 igr in upm: pkts rcvd, with RCODE violation   0000359810913821  30 -
 551 igr in upm: pkts with symbol/sequence error rcvd 0000425092490108  30 -
 552 igr in upm: pkts with error                   0000000000176136  30 -
 582 igr in upm: pkts rcvd, with RCODE violation   0000000000292641  29 -
 583 igr in upm: pkts with symbol/sequence error rcvd 0000000000114014  29 -
 614 igr in upm: pkts rcvd, with RCODE violation   0000133362265995  31 -
 615 igr in upm: pkts with symbol/sequence error rcvd 0000146701474013  31 -
 616 igr in upm: pkts with error                   0000000000157479  31 -
```

```
646 igr in upm: pkts rcvd, with RCODE violation   0000000002160959  32 -
647 igr in upm: pkts with symbol/sequence error rcvd 0000000003722562  32 -
648 igr in upm: pkts with error                   0000000000000002  32 -
```