排除故障在连结7000系列交换机的硬件转发问题

目录

简介

本文描述如何排除故障在F3系列模块的硬件转发问题Cisco Nexus 7000系列交换机的。

先决条件

要求

思科建议您有一熟悉用思科连结操作系统(NX-OS)和基本连结体系结构，在您继续进行在本文描述的信息前。

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)
- Cisco N7K F3系列模块(N7K-F312FQ-25， 12波尔特10/40千兆以太网模块)
- Cisco NX-OS版本6.2.8a和以上

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。如果您的网络实际，请保证您了解所有命令潜在影响。

背景信息

在使用硬件故障排除的工具着重主要一些建立的-本文，当您用尽您的转发表或控制层面的软件部件时。一个这样工具是嵌入式逻辑分析器模块(伊拉姆)，是Application-specific integrated circuit (ASIC)获取单个数据包并且显示入口数据包如何出现在数据总线(DBUS)和结果BUS (RBUS)在转发以后。

ASIC在转发渠道内是嵌入式的，并且在实时能获取一数据包，不用中断到性能或控制面板资源。这帮助应答问题例如：

- 数据包是否到达了转发引擎(FE) ？
- 在什么端口和VLAN数据包接收？
- 数据包如何出现(Layer2 (L2)或Layer4 (L4)数据) ？
- 数据包如何被修改，并且在哪里发送？

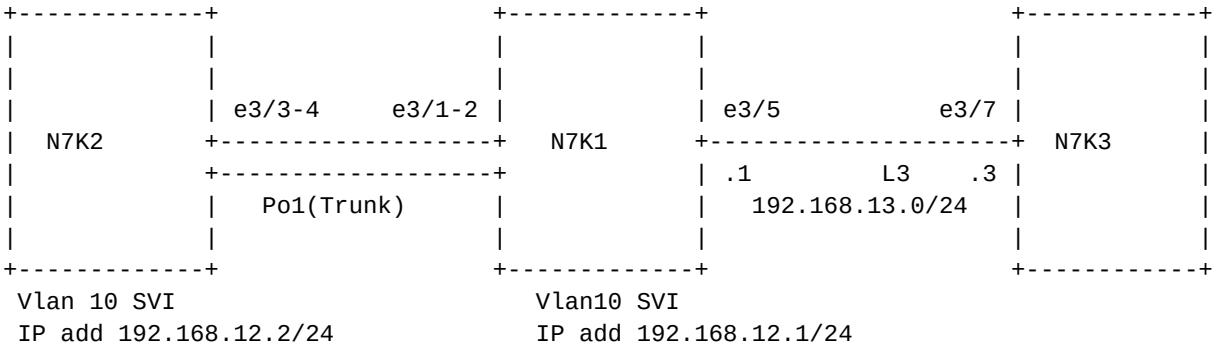
伊拉姆是由Cisco技术支持中心(TAC)工程师是最常用的在硬件交换平台工作的一个强大，粒状和不打扰的工具。 然而，知道是重要的伊拉姆工具只获取一数据包在时间。即接收的第一数据包，在伊拉姆被触发后。

故障排除

此部分描述如何排除故障在一个F3系列模块的ELAM在不介入使用分支电缆的部署，以及使用分支电缆的部署。

排除故障在F3系列模块的ELAM，不用分支电缆

这是使用在此部分中的示例的拓扑：



这是关于此拓扑的一些笔记：

- N7Ks运行NX-OS版本6.2.8a。
- Ping从N7K2 VLAN10接口被发送到192.168.12.1远程IP地址。
- 在N7K1的伊拉姆捕获数据包。
- 使用N7K-F312FQ-25，是12波尔特10/40千兆以太网模块插入到Slot3。

在您开始排除故障您的系统前，您应该确认基本连通性：

```
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
```

应该也验证MAC控制(MAC)地址学习在Supervisor引擎(Sup)和下一跳的模块的您：

```
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
```

此输出显示在模块/硬件的MAC学习;然而，为了认识接口，您必须转换索引：

```
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---
```

输入这些命令为了得到虚拟设备上下文(VDC)编号(在本例中，是**3**)并且检查MAC直接地在模块：

```
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
```

确定在使用为了转发在Sup的流量从N7K2的端口通道1的链路，以及使用为了发送从N7K3的一回复的链路，当端口通道1从N7K1使用到N7K2时：

```
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
```

发送从N7K2的一ping (IP地址192.168.12.2)并且获取在N7K1的数据包在入口方向为了确认数据包转发对N7K3 (IP地址192.168.13.3)。

在您发送ping前，您应该有硬件积累的知识。完成这些步骤为了了解积累：

1. 附加模块：

   ```
   N7K1# attach module 3
   Attaching to module 3 ...
   To exit type 'exit', to abort type '$.'
   ```

2. 识别*侧面部队*实例。侧面部队是在芯片(SOC) F3系列模块的ASIC的一交换机。每个侧面部队被映射到模块的两个外部端口(信息更改每种模块类型和是特定对N7K-F312FQ-25)。

   有模块的12个端口，和每ASIC地图对前面板的两个端口，因此意味着有6个(0-5)侧面部队实例可用在模块(实例计数基于零的)。 **注意**：保证您有网络管理权限，在您开始前。因为您获取从

N7K2到达通过在N7K1的端口通道1的数据包，请寻找被映射对每个实例的端口(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，并且开始捕获。了解，然而是重要的，有能用伊拉姆触发使用的许多选项：

```
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-----这两个选项是重要，如果在捕获(由交换机接收)的数据包要包括DBUS。这是对查找没有被服从的原始数据包。RBUS在硬件里显示查找结果DBUS的。对于一完整伊拉姆和分析，您必须捕获RBUS和DBUS。

下输出显示您能获取与DBUS选项数据包的种类。在本例中，互联网协议版本4 (IPv4)数据包选择：

```
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
```

这是您能选择使用的一些其它选项：

```
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
```

在本例中，**如果**把柄用于为了选择捕获的一个条件。显示在下输出中根据L2、L3和L4报头的大多选项。源和目的IP地址也使用捕获。

```
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配置通常不是复杂和保持简单化。

4. 为了检查触发，输入**status命令**，开始捕获进程和启动一ping从N7K2到N7K3 (192.168.12.1到192.168.13.3) ：

```
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**触发**，则请验证RBUS和DBUS是否有同一个序号为了确认他们是为同一数据包。在本例中，使用**0x55**，但是显示序号的列不同的：

```
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**显示d总线**并且**显示rbus**命令为了验证DBUS和RBUS。寻找*资料来源索引*在DBUS命令输出中和*目的地索引*在RBUS命令输出中：

```
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. 检查目的地索引和资料来源索引在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
```

此输出确认数据包在端口通道1 (Po1)接收和通过Eth3/5转发。

8. 验证在模块的Local Target Logic (LTL)正确编程的：

```
 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. 获取伊拉姆数据包在出口。为了获取数据包，请发送从IP address192.168.13.3的一ping回复对192.168.12.2。您必须设置与**出口**关键字的捕获在端口通道1接口(e3/1-2)。接口属于实例**0**，如前所述。

```
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
```

如显示，两个源和目的索引是DBUS (在入口捕获显示的不同的)部分。

```
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 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源和目的IP地址是正确，如解码在入口伊拉姆数据包捕获以后;然而，方向确实在对面，当与入口伊拉姆比较，作为回程数据流捕获。

10. 检查科罗拉多-基于逻辑(CBL)模块的3在N7K1的端口通道1为了确定VLAN10是否转发在它间的流量。CBL是一个每个物理接口基于逻辑，因此您应该输入成员接口编号在N7K1的端口通道1，不是端口通道频道数。在下输出中，您能看到VLAN10转发它正如所料。

    CBL用于为了确定端口的生成树协议状态在硬件内的。很可能，接口显示转发，当您检查STP在Sup时的VLAN，但是模块阻塞流量。 **注意**：您必须单个检查CBL两个成员接口(e3/1和e3/2)。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：前面的命令是为端口通道1 (模块3在e3/1)。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：同样地，此命令检查CBL端口通道2 (**e3/2**)。

排除故障在F3系列模块的ELAM用分支电缆

一个F3系列模块的ELAM步骤，当分支电缆连接时与在一个正常模块端口的ELAM步骤没有所不同。然而，有一些更改关于端口索引管理器(PIXM)的验证在尝试转换索引期间到前面板编号，在接口从分支电缆情况下接收。

这是使用在此部分中的示例的拓扑：

```
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   |                                                         |
 ------------------------------------------------------------------------------
```

对于此示例，分支电缆连接对以太网接口*3/8*，分成40千兆位端口四个10 Gigabit端口。必需的配置在此部分提供供参考。

```
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
```

在上一个输出中，您能看到以太网接口*3/7*仍然是40千兆位端口;然而，以太网接口*3/8*当前是残破的到四个10 Gigabit端口，可以单个配置：

```
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
```

开始从N7K3交换的虚拟接口(SVI) 20 IP地址(192.168.20.3)的数据包捕获对4500 SVI 20 IP地址(192.168.20.1)。数据包在出口的N7K3将捕获到4500，并且回复从4500被发送到N7K3。

正如前面部分所描述，您必须有侧面部队实例的知识为了应用触发。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在侧面部队实例*3.*，即然您认识实例，您能通过源和目的IP地址放置触发。由于您将捕获从N7K3的ping请求到4500，它将是出口伊拉姆。

```
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
+------------------------------------------------------------------------+
+------------------------------------------------------------------------+
```

ping从N7K3启动到4500 ：

```
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
```
这是伊拉姆状态：

```
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
```
验证序号是相同的：

```
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
```
序号是相同的。现在您能检查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
```

转换目的地和资料来源索引到前面板端口为了确认流：

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

此输出显示资料来源索引。您知道它是正确由于来自到N7K3 Sup的ping。下输出显示出口接口(e3/8/1)，是在N7K的两个接口之一允许VLAN 20。另一个接口是*e3/8/4*，在4500阻塞由于STP。

```
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
```

验证用在N7K的分支电缆创建的端口的CBL。为了检查CBL，您必须有所有的硬件端口端口号新形成的端口。

注意：接口e3/8在交换机不存在。仅新形成的端口出现。

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

由于使用分支电缆，并且e3/8接口是不存在的在交换机，使用为了获取硬件端口端口号的计算更改。对于支持分组讨论(&B)的所有模块，硬件端口编号不同的。您应该首先证实端口支持是否分组讨论(&B) ：

```
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
```

如显示，端口*e3/7*支持分组讨论(&B)，因此意味着其带宽可以分成四个10 Gigabit端口。同样地，有100千兆位端口的其他F3系列模块可以分成有10吉比特的十个端口中的每一个或者有40吉比特的三个端口与超额预订。具体情况取决于模块。

因为在本例中的F3系列模块有40千兆位端口，并且每个端口可以分成四个端口中的每一个，每个端口的硬件端口端口号是0-3， 4-7， 8-11...40-43， 44-47在一基于零的等级。如果有在端口的分支电缆第一示例的，则其硬件端口编号将是0， 1， 2和3。如果没有分支电缆，则其硬件端口编号将

是0 (1， 2和3不会是活跃的)。

因为parent端口是e3/8，其硬件端口端口号将是28，如果使用，不用分支电缆，并且将是28， 29， 30和31，如果与分支电缆一起使用。此命令输出显示活动硬件端口(零基于) ：

```
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
```

残破的端口硬件端口端口号是28，当前拆分到四(28-32)。现在您在硬件里能附加模块3和检查CBL ：

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

F3系列模块预计端口号被格式化符合一基于一的缩放。为此，您应该输入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.
```

这是以太网接口的*3/8/1*运行的配置为了检查和确认VLAN转发状态：

```
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.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   |                                                         |
 ------------------------------------------------------------------------------
```

这是以太网接口的*3/8/2*运行的配置为了检查和确认VLAN转发状态：

```
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   |                                                         |
 ------------------------------------------------------------------------------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   |                                                         |
 ------------------------------------------------------------------------------
```

这是以太网接口的*3/8/3*运行的配置为了检查和确认VLAN转发状态：

```
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   |                                                         |
 ------------------------------------------------------------------------------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   |                                                         |
 ------------------------------------------------------------------------------
```

这是以太网接口的*3/8/4*运行的配置为了检查和确认VLAN转发状态(所有配置的VLAN允许)：

```
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   |                                                         |
 ------------------------------------------------------------------------------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显示正确VLAN转发。

您能使用**show hardware内部错误模块<module number>**命令为了获取硬件端口端口号。此命令是有用的，当您必须检查在**show interface x/y**命令输出中没出现的所有内部丢包时。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 -
```