

Dépannez les questions d'expédition de matériel sur des Commutateurs de gamme de Nexus 7000

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Dépannez](#)

[Dépannez ELAM sur des modules de gamme F3 sans câbles de séparation](#)

[Dépannez ELAM sur des modules de gamme F3 avec des câbles de séparation](#)

Introduction

Ce document décrit comment dépanner des questions d'expédition de matériel sur des modules de gamme F3 pour le Commutateurs de la gamme Cisco Nexus 7000.

Conditions préalables

Conditions requises

Cisco recommande que vous ayez une connaissance du système d'exploitation de Cisco Nexus (NX-OS) et l'architecture de base de Nexus avant que vous poursuiviez les informations qui sont décrites dans ce document.

[Composants utilisés](#)

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Commutateurs de la gamme Cisco Nexus 7000 (N7K)
- Modules de Gigabit Ethernet (N7K-F312FQ-25, 12-Port 10/40) de modules de gamme F3 de Cisco N7K
- Versions 6.2.8a et ultérieures de Cisco NX-OS

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est vivant, assurez-vous que vous comprenez l'impact potentiel de n'importe quelle commande.

Informations générales

Ce document se concentre principalement sur certains des outils intégrés qui sont utilisés pour le dépannage matériel quand vous avez épuisé votre pièce de logiciel de la table d'expédition ou de l'avion de contrôle. Un tel outil est le module inclus d'analyseur de logique (ELAM), qui est un circuit intégré spécifique (ASIC) qui capture un paquet simple et affiche comment le paquet d'entrée apparaît sur le BUS de données (D-BUS) et le BUS de résultat (RBUS) après transmission.

L'ASIC est encastré dans le pipeline d'expédition, et il peut capturer un paquet en temps réel sans interruptions à la représentation ou aux ressources du plan de contrôle. Ceci aide à répondre à des questions comme :

- Le paquet a-t-il atteint l'engine d'expédition (technicien) ?
- Sur quels port et VLAN le paquet est-il reçu ?
- Comment le paquet apparaît-il (couche 2 (L2) ou poser 4 données (L4)) ?
- Comment le paquet est-il modifié, et d'où est-il envoyé ?

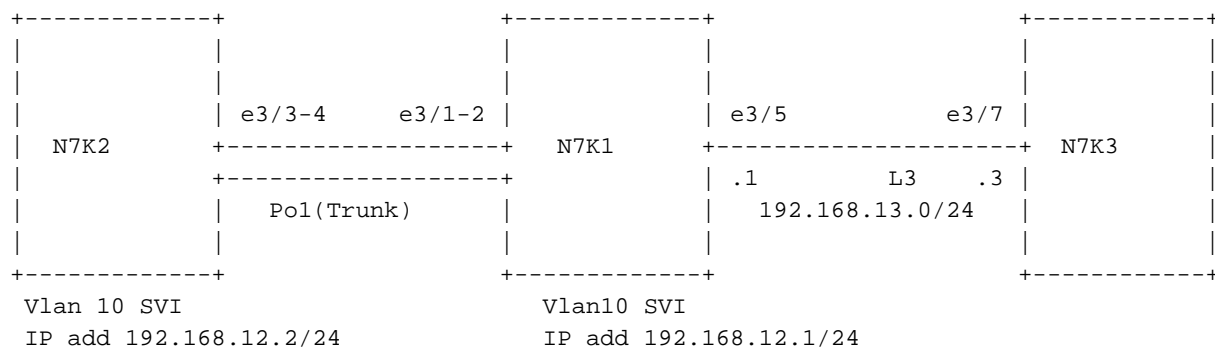
L'ELAM est un outil puissant, granulaire, et non intrusif qui est le plus utilisé généralement par les ingénieurs du centre d'assistance technique Cisco (TAC) qui travaillent aux Plateformes de commutation de matériel. Cependant, il est important de savoir que l'outil ELAM capture seulement un paquet au temps. C'est-à-dire, le premier paquet qui est reçu après que l'ELAM soit déclenché.

Dépannez

Cette section décrit comment dépanner ELAM sur un module de gamme F3 dans les déploiements qui ne comportent pas l'utilisation d'un câble de séparation, aussi bien que les déploiements qui utilisent des câbles de séparation.

Dépannez ELAM sur des modules de gamme F3 sans câbles de séparation

C'est la topologie qui est utilisée pour les exemples dans toute cette section :



Voici quelques notes au sujet de cette topologie :

- La version 6.2.8a du passage NX-OS N7Ks.

- Des pings sont envoyés de l'interface N7K2 VLAN 10 à une adresse IP distante de 192.168.12.1.
- L'ELAM capture des paquets sur le N7K1.
- Un N7K-F312FQ-25 est utilisé, qui est un module des Gigabit Ethernet 12-Port 10/40 inséré dans l'emplacement 3.

Avant que vous commenciez à dépanner votre système, vous devriez confirmer la Connectivité de base :

```
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
```

Vous devriez également vérifier l'autoapprentissage d'adresse de Contrôle d'accès au support (MAC) sur l'engine de superviseur (petite gorgée) et le module pour le prochain saut :

```
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

Cette sortie affiche le MAC apprenant sur le module/matériel ; cependant, afin de connaître l'interface, vous devez convertir l'index :

```
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---

Sélectionnez ces commandes afin d'obtenir le nombre du contexte de périphérique virtuel (volts continu) (dans cet exemple, il est 3) et vérifie le MAC directement sur le module :

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

Déterminez le lien sur le Port canalisé 1 qui est utilisé afin d'expédier le trafic sur la petite gorgée de N7K2, aussi bien que le lien qui est utilisé afin d'envoyer une réponse de N7K3 quand le Port canalisé 1 est utilisé de N7K1 à N7K2 :

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
```

Envoyez un ping de N7K2 (adresse IP 192.168.12.2) et capturez les paquets sur N7K1 dans la direction d'entrée afin de confirmer que les paquets sont expédiés à N7K3 (adresse IP 192.168.13.3).

Avant que vous envoyiez le ping, vous devriez avoir la connaissance de l'habillage de matériel. Terminez-vous ces étapes afin de comprendre l'habillage :

1. Reliez le module :

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

2. Identifiez l'exemple de *flanker*. Le FLANKER est un commutateur sur la puce (SOC) ASIC pour le module de gamme F3. Chaque flanker est tracé à deux ports externes sur le module (les changements des informations par type de module et est spécifique au N7K-F312FQ-25).

Il y a 12 ports sur le module, et cartes chaque ASIC à deux ports sur le panneau avant, ainsi il signifie qu'il y a 6 exemples de flanker (de 0-5) disponibles sur le module (le compte d'exemple est basé sur zéro). Remarque: Assurez-vous que vous avez des privilèges d'administrateur de réseau avant que vous commenciez. Car vous capturez le paquet qui arrive de N7K2 par l'intermédiaire du Port canalisé 1 sur N7K1, recherchez les ports (e3/1 et e3/2) qui sont tracés à chaque exemple :

```
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	0
6	0	2	2	2	0
7	1	3	3	3	0
8	1	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. Sélectionnez l'exemple, placez le déclencheur, et commencez la capture. Il est important de comprendre, cependant, qu'il y a beaucoup d'options qui peuvent être utilisées avec le déclencheur ELAM :

```
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-----Ces deux options sont importantes si vous voulez inclure le D-BUS dans la capture (le paquet qui est reçu par le commutateur). C'est le paquet cru qui n'est pas soumis à une consultation. Le RBUS donne les résultats de recherche dans le matériel pour un D-BUS. Pour un ELAM et une analyse complète, vous devez capturer le RBUS et le D-BUS.

La prochaine sortie affiche les types de paquets que vous pouvez capturer avec l'option de D-BUS. Dans cet exemple, le paquet de la version 4 d'Internet Protocol (ipv4) est sélectionné :

```
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
```

Voici quelques options supplémentaires que vous pouvez choisir pour utiliser :

```
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 Dans cet exemple, si le traitement est utilisé afin de sélectionner une condition pour la capture. La plupart des options qu'affiché dans la prochaine sortie sont basé sur les en-têtes L2, L3, et L4. La source et les adresses IP de destination sont également utilisées pour la capture.

```

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
lb10-eos            Lb10 eos
lb10-exp            Lb10 exp
lb10-lb1            Lb10 lb1
lb10-ttl            Lb10 ttl
lb10-valid          Lb10 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
pd-tag-gt-2	Pdt-tag-gt-2
pd-tag0	Pdt-tag0
pd-tag1	Pdt-tag1
pd-valid	Pdt-valid
pd-value	Pdt-value
port-id	Port-id
rbh	Rbh
rdt	Rdt
recirc-shim-vxlan-src-peer-id	Recirc 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
v1	v1
vlan	Vlan
vn-p	Vn p
vn-valid	Vn-valid
vqi	Vqi
vqi-valid	Vqi-valid
vsl-num	Vsl-num

Cette sortie affiche l'option finale de déclencheur :

```

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

Remarque: La configuration RBUS n'est habituellement pas complexe et maintenue simple.

4. Afin de vérifier le déclencheur, sélectionner la commande d'état, commencer le processus de capture, et initier un ping de N7K2 à N7K3 (192.168.12.1 à 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. Si les expositions d'état **déclenchées**, vérifient alors si les RBUS et le D-BUS ont le même numéro de séquence afin de confirmer qu'ils sont pour le même paquet. Dans cet exemple, **0x55** est utilisé, mais la colonne qui affiche le numéro de séquence est différente :

```
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. Entrez dans le **d-bus d'exposition** et affichez que les commandes de **rbus** vérifiaient le **D-BUS** et le **RBUS**. Recherchez *l'index de source* dans la sortie de commande de **D-BUS** et *l'index de destination* dans la sortie de commande **RBUS** :

```
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          lb10-vld          : 0x0
lb10-eos           : 0x0          lb10-lbl          : 0x0
lb10-exp           : 0x0          lb10-ttl          : 0x0
lb11-exp           : 0x0          lb11-ttl          : 0x0
ipv4                : 0x0          ipv6                : 0x0
l4-protocol        : 0x1          df                  : 0x0
mf                  : 0x0          frag                : 0x0
ttl                 : 0xff         13-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-lq        : 0x0
status-is-lq        : 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. Vérifiez l'index de destination et l'index de source sur la petite gorgée :

```
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 Cette sortie confirme que le paquet a été reçu sur le Port canalisé 1 (Po1) et a été expédié par l'intermédiaire d'Eth3/5.

8. Vérifiez la logique locale de cible (LTL) sur le module pour la programmation correcte :

```
module-3# show system internal pixmc info ltl-cb ltl 0xa2c
```

ltl	ltl_type	if_index	lc_type	vdc	v4_fpo	v5_fpo	base_fpo_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. Capturez le paquet ELAM sur le de sortie. Afin de capturer le paquet, envoyez une réponse ping d'IP address192.168.13.3 à 192.168.12.2. Vous devez placer la capture avec le mot clé de **sortie** sur les interfaces du Port canalisé 1 (e3/1-2). Les interfaces appartiennent pour citer **0**, comme décrit précédemment.

```

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-12-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-12-elam)#

```

```

module-3(fln-12-elam)# show dbus | in seq
sequence-number      : 0x8d          vl          : 0x3

```

!--- The sequence number is the same.

```

module-3(fln-12-elam)# show rbus | in seq
vl                   : 0x0          sequence-number    : 0x8d

```

```

module-3(fln-12-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-lq     : 0x0
status-is-lq   : 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 Comme affiché, la source et les index de destination sont une pièce du D-BUS (différent qui affiché dans la capture d'entrée).

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-vsant      : 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

```

La source et les adresses IP de destination sont correctes, comme décodé après que la capture de paquet du d'entrée ELAM ; cependant, la direction est certainement vis-à-vis d'une fois comparée au d'entrée ELAM, comme trafic de retour est capturée.

- Vérifiez la logique basée sur du Colorado (CBL) pour le module 3 du Port canalisé 1 sur N7K1 afin de déterminer si le VLAN 10 trafiquent en avant à travers lui. Le CBL est une logique basée par interface par-physique, ainsi vous devriez introduire le nombre d'interface de membre du Port canalisé 1 sur N7K1, pas le nombre de Port canalisé. Dans la prochaine sortie, vous pouvez voir ce VLAN 10 en avant il comme prévu.

Le CBL est utilisé afin de déterminer l'état du Protocole Spanning Tree (STP) d'un port dans le matériel. Il est possible que l'interface affiche l'expédition quand vous vérifiez le STP pour un VLAN sur la petite gorgée, mais le module bloque le trafic. Remarque: Vous devez examiner le CBL individuellement pour assurer chacun des deux interfaces de membre (e3/1 et 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 | |
|-----|-----|

```

Remarque: La commande précédente est pour le Port canalisé 1 (le module 3 est sur 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 | |
|-----|-----|

```

Remarque: De même, cette commande vérifie le CBL pour le Port canalisé 2 (e3/2).

Dépannez ELAM sur des modules de gamme F3 avec des câbles de séparation

La procédure ELAM pour un module de gamme F3 quand un câble de séparation est connecté ne diffère pas des procédures ELAM sur un port régulier de module. Cependant, il y a quelques modifications en vue de la vérification du gestionnaire d'index de port (PIXM) pendant les tentatives de convertir l'index en nombre de panneau avant, dans ce cas les interfaces sont reçues du câble de séparation.

Voici la topologie qui est utilisée pour les exemples dans toute cette section :

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

Pour cet exemple, un câble de séparation est connecté à l'interface Ethernet 3/8, qui divise le port 40-Gigabit en quatre ports 10-Gigabit. La configuration exigée est donnée dans cette section pour la référence.

```
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
```

Dans la sortie précédente, vous pouvez voir que l'interface Ethernet 3/7 est toujours un port 40-Gigabit ; cependant, l'interface Ethernet 3/8 est maintenant cassée dans quatre ports 10-Gigabit, qui peuvent être configurés individuellement :

```
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

```

Commencez la capture de paquet de l'adresse IP 20 de l'interface virtuelle commutée par N7K3 (SVI) (192.168.20.3) 4500 à l'adresse IP SVI 20 (192.168.20.1). Le paquet sera capturé sur N7K3 sur le de sortie à 4500, et la réponse est envoyée de 4500 à N7K3.

Comme décrit dans la section précédente, vous devez avoir la connaissance de l'exemple de flanker afin d'appliquer le déclencheur. Cette sortie affiche la connexion du module 3 :

```

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

```

```

+-----+
+-----+

```

Dans cette sortie, le port 8 est sur l'exemple 3. de flanker maintenant que vous connaissez l'exemple, vous peut placer le déclencheur par l'intermédiaire de la source et des adresses IP de destination. Puisque vous capturerez la requête ping de N7K3 à 4500, ce sera un de sortie ELAM.

```

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

```

```

+-----+
+-----+

```

Le ping est initié de N7K3 à 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
```

Voici l'état ELAM :

```
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
```

Vérifiez que les numéros de séquence sont identiques :

```
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
```

Les numéros de séquence sont identiques. Maintenant vous pouvez vérifier les informations de D-BUS et RBUS :

```
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                lb10-vld         : 0x0
lb10-eos            : 0x0                lb10-lbl        : 0x0
lb10-exp            : 0x0                lb10-ttl        : 0x0
lb11-exp            : 0x0                lb11-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-lq  : 0x0
status-is-lq  : 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-vsant     : 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

```

Convertissez la destination et les index de source en ports de panneau avant afin de confirmer l'écoulement :

```

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

```

Cette sortie affiche l'index de source. Vous savez qu'il est correct en raison du ce de pings été livré à N7K3 de la petite gorgée. La prochaine sortie affiche l'interface de sortie (e3/8/1), qui est l'une des deux interfaces sur le N7K qui permet le VLAN 20. L'autre interface est e3/8/4, qui est bloqué sur 4500 dus au 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

```

Vérifiez le CBL pour les ports qui ont été créés avec le câble de séparation sur le N7K. Afin de vérifier le CBL, vous devez avoir les numéros de port de matériel pour tous les ports récemment formés.

Remarque: L'interface e3/8 n'existe pas sur le commutateur. Seulement les ports récemment formés apparaissent.

```

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

```

Puisque le câble de séparation est utilisé et l'interface e3/8 est inexistante sur le commutateur, le calcul qui est utilisé afin d'obtenir le numéro de port de matériel change. Pour n'importe quel module qui prend en charge la rubrique, la numérotation de port de matériel est différente. Vous devriez d'abord vérifier si un port prend en charge la rubrique :

```

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
```

Comme affiché, le port e3/7 prend en charge la rubrique, ainsi il signifie que sa bande passante peut être divisée en quatre ports 10-Gigabit. De même, d'autres modules de gamme F3 qui ont les ports 100-Gigabit peuvent être divisés en dix met en communication chacun avec 10-Gigabits, ou trois ports avec 40-Gigabits avec le surabonnement. Ceci dépend du module.

Puisque le module de gamme F3 dans cet exemple a les ports 40-Gigabit, et chaque port peut être divisé en quatre met en communication chacun, les numéros d'accès de matériel pour chaque port est 0-3, 4-7, 8-11...40-43, 44-47 sur une échelle basée sur zéro. Si vous avez un câble de séparation sur un port pour le premier exemple, alors sa numérotation de port de matériel sera 0, 1, 2 et 3. Si vous n'avez pas un câble de séparation, alors sa numérotation de port de matériel sera 0 (1, 2, et 3 ne seront pas en activité).

Car le port de parent est e3/8, son numéro de port de matériel sera 28 s'il est utilisé sans câble de séparation, et il sera 28, 29, 30, et 31 s'il est utilisé avec le câble de séparation. Cette sortie de commande affiche les ports de matériel actifs (zéro basé) :

```
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_ports:
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
```

Le numéro de port cassé de matériel de port est **28**, qui est maintenant coupé en quatre (28-32). Maintenant vous pouvez l'attach module 3 et vérifier le CBL dans le matériel :

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

Le module de gamme F3 s'attend au numéro d'accès à formater selon une échelle basée sur un. Pour cette raison, vous devriez écrire 29, 30, 31, et 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.

Voici la configuration en cours pour l'interface Ethernet 3/8/1 afin de vérifier et confirmer l'état d'expédition VLAN :

```
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 | |
-----

```

Voici la configuration en cours pour l'interface Ethernet 3/8/2 afin de vérifier et confirmer l'état d'expédition VLAN :

```
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 | |
-----

```

Voici la configuration en cours pour l'interface Ethernet 3/8/3 afin de vérifier et confirmer l'état d'expédition VLAN :

```
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  |
-----

```

Voici la configuration en cours pour l'interface Ethernet 3/8/4 afin de vérifier et confirmer l'état d'expédition VLAN (on permet tous les VLAN configurés) :

```
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  |
-----

```

Le CBL prouve que les VLAN corrects sont expédiés.

Vous pouvez employer la commande de **number** de **<module de module d'erreur interne de matériel d'exposition** afin d'obtenir le numéro de port de matériel. Cette commande est utile quand vous devez vérifier toutes les baisses internes qui n'apparaissent pas dans la sortie de commande de **l'interface x/y d'exposition**. Voici un exemple :

```
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 -