

Nexus 7000 Series 스위치의 하드웨어 포워딩 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[문제 해결](#)

[브레이크아웃 케이블 없이 F3 Series 모듈의 ELAM 문제 해결](#)

[브레이크아웃 케이블을 사용하여 F3 Series 모듈의 ELAM 문제 해결](#)

소개

이 문서에서는 Cisco Nexus 7000 Series 스위치용 F3 Series 모듈의 하드웨어 포워딩 문제를 해결하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

이 문서에 설명된 정보를 진행하기 전에 Cisco Nexus NX-OS(Operating System) 및 기본 Nexus 아키텍처에 대해 숙지할 것을 권장합니다.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Cisco Nexus 7000 Series 스위치(N7K)
- Cisco N7K F3 Series 모듈(N7K-F312FQ-25, 12포트 10/40기가비트 이더넷 모듈)
- Cisco NX-OS 버전 6.2.8a 이상

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

이 문서에서는 포워딩 테이블 또는 컨트롤 플레인의 소프트웨어 부분을 소진했을 때 하드웨어 트러블슈팅에 사용되는 일부 내장 툴에 주로 초점을 맞춥니다. 이러한 툴 중 하나는 ELAM(Embedded Logic Analyzer Module)입니다. ELAM은 단일 패킷을 캡처하고 전달 후 DBUS(Data BUS) 및

RBUS(Result BUS)에 인그레스 패킷이 어떻게 나타나는지 보여줍니다.

ASIC는 포워딩 파이프라인 내에 내장되어 있으며 성능 또는 컨트롤 플레인 리소스에 지장을 주지 않고 실시간으로 패킷을 캡처할 수 있습니다. 이를 통해 다음과 같은 질문에 답변할 수 있습니다.

- 패킷이 FE(Forwarding Engine)에 도달했습니까?
- 패킷이 수신되는 포트 및 VLAN은 무엇입니까?
- 패킷은 어떻게 표시됩니까(레이어 2(L2) 또는 레이어 4(L4))?
- 패킷은 어떻게 변경되며 어디로 전송됩니까?

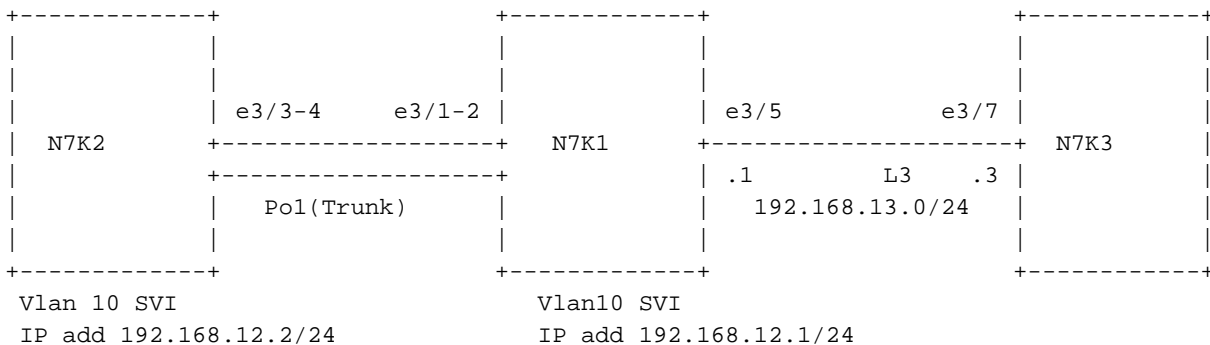
ELAM은 하드웨어 스위칭 플랫폼에서 작업하는 Cisco TAC(Technical Assistance Center) 엔지니어가 가장 일반적으로 사용하는 강력하고 세분화된 비침입적 툴입니다. 그러나 ELAM 툴은 한 번에 하나의 패킷만 캡처한다는 사실을 알아야 합니다. 즉, ELAM이 트리거된 후 수신되는 첫 번째 패킷입니다.

문제 해결

이 섹션에서는 브레이크아웃 케이블을 사용하지 않는 구축에서 F3 Series 모듈의 ELAM을 트러블 슈팅하는 방법과 브레이크아웃 케이블을 사용하는 구축에 대해 설명합니다.

브레이크아웃 케이블 없이 F3 Series 모듈의 ELAM 문제 해결

이 섹션 전체에서 예제에 사용되는 토폴로지입니다.



이 토폴로지에 대한 몇 가지 참고 사항은 다음과 같습니다.

- N7K는 NX-OS 버전 6.2.8a를 실행합니다.
- Ping은 N7K2 VLAN 10 인터페이스에서 192.168.12.1의 원격 IP 주소로 전송됩니다.
- ELAM은 N7K1에서 패킷을 캡처합니다.
- N7K-F312FQ-25가 사용되며 슬롯 3에 삽입되는 12포트 10/40기가비트 이더넷 모듈입니다. 시스템 문제를 해결하기 전에 기본 연결을 확인해야 합니다.

```

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
```

또한 Supervisor Engine(Sup)에서 MAC(Media Access Control) 주소 학습과 다음 홉의 모듈을 확인해야 합니다.

```
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

이 출력은 모듈/하드웨어에서 MAC 학습을 보여줍니다.그러나 인터페이스를 확인하려면 인덱스를 변환해야 합니다.

```
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---

VDC(Virtual Device Context) 번호(이 예에서는 3)를 얻고 모듈에서 직접 MAC를 확인하려면 다음 명령을 입력합니다.

```
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

포트 채널 1이 N7K1에서 N7K2로 사용될 때 N7K3에서 회신을 보내는 데 사용되는 링크와 N7K2에서 N7K2로 Sup의 트래픽을 전달하는 데 사용되는 포트 채널 1의 링크를 결정합니다.

```
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

N7K2(IP 주소 192.168.12.2)에서 ping을 보내고 인그레스 방향으로 N7K1의 패킷을 캡처하여 패킷이 N7K3(IP 주소 192.168.13.3)으로 전달되었는지 확인합니다.

Ping을 전송하기 전에 하드웨어 증강에 대한 정보를 알아야 합니다.다음 단계를 완료하여 추가 정보를 파악합니다.

1. 모듈을 연결합니다.

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

2. 측면 인스턴스를 식별합니다.플란커는 F3 Series 모듈의 SOC(Switch on Chip) ASIC입니다.각 플래너는 모듈의 2개의 외부 포트에 매핑됩니다(모듈 유형별로 정보가 변경되며 N7K-F312FQ-25에 따라 다름).

모듈에는 12개의 포트가 있으며, 각 ASIC는 전면 패널의 2개 포트에 매핑됩니다. 즉, 모듈에서 사용할 수 있는 6개의(0-5) 플란커 인스턴스가 있습니다(인스턴스 수는 0 기반임). 참고:시작하기 전에 네트워크 관리 권한이 있는지 확인합니다.N7K1의 포트 채널 1을 통해 N7K2에서 도착하는 패킷을 캡처할 때 각 인스턴스에 매핑된 포트(e3/1 및 e3/2)를 확인합니다.

```
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 |      0
2       |      0 |      0 |      0 |      0 |      0 |      0
3       |      1 |      1 |      1 |      1 |      1 |      0
4       |      1 |      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 |      4 |      0
10      |      4 |      4 |      4 |      4 |      4 |      0
11      |      5 |      5 |      5 |      5 |      5 |      0
12      |      5 |      5 |      5 |      5 |      5 |      0

!--- The L2LKP for both ports is 0, so both belong to instance 0.
```

+-----+
+-----+
3. 인스턴스를 선택하고 트리거를 설정하고 캡처를 시작합니다. 그러나 ELAM 트리거와 함께 사용할 수 있는 여러 옵션이 있음을 이해하는 것이 중요합니다.

```
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-----

이 두 옵션은 캡처에 DBUS(스위치에서 수신한 패킷)를 포함하려는 경우 중요합니다. 조회가 적용되지 않는 원시 패킷입니다. RBUS는 DBUS에 대한 하드웨어의 조회 결과를 표시합니다. 전체 ELAM 및 분석을 위해서는 RBUS와 DBUS를 모두 캡처해야 합니다.

다음 출력은 DBUS 옵션으로 캡처할 수 있는 패킷의 유형을 보여줍니다. 이 예에서는 IPv4(Internet Protocol Version 4) 패킷이 선택됩니다.

```
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
```

사용할 수 있는 몇 가지 추가 옵션은 다음과 같습니다.

```
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
```

이 예에서 **if** 핸들은 캡처에 대한 조건을 선택하기 위해 사용됩니다. 다음 출력에 표시되는 대부분의 옵션은 L2, L3 및 L4 헤더를 기반으로 합니다. 소스 및 대상 IP 주소도 캡처에 사용됩니다.

```
module-3(fln-l2-elam)# trigger dbus ipv4 ingress if ?
<CR>
acos          Acos
block-capture Capture 12 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
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

이 출력은 최종 트리거 옵션을 표시합니다.

```

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

```

참고:RBUS 컨피그레이션은 일반적으로 복잡하지 않으며 간단합니다.

4. 트리거를 확인하려면 **status** 명령을 입력하고 캡처 프로세스를 시작하고 N7K2에서 N7K3으로 ping을 시작합니다(192.168.12.1~192.168.13.3).

```

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. 상태가 Triggered(트리거됨)로 표시되면 RBUS와 DBUS의 시퀀스 번호가 동일한지 확인하여 동일한 패킷에 해당하는지 확인합니다. 이 예제에서는 0x55가 사용되지만 시퀀스 번호를 표시하는 열은 다릅니다.

```
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. DBUS 및 RBUS를 확인하려면 show dbus 및 show rbus 명령을 입력합니다. DBUS 명령 출력에서 소스 인덱스를 찾고 RBUS 명령 출력에서 대상 인덱스를 찾습니다.

```
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-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       dil-v5-delta-length : 0x0
dil-v5-delta-length-plus: 0x0    dil-v4-delta-length : 0x0
dil-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. Sup에서 대상 인덱스 및 소스 인덱스를 확인합니다.

```
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
```

이 출력은 패킷이 포트 채널 1(Po1)에서 수신되고 Eth3/5를 통해 전달되었음을 확인합니다.

8. 모듈의 LTL(Local Target Logic)에서 올바른 프로그래밍을 확인하십시오.

```
module-3# show system internal pixmc info ltl-cb ltl 0xa2c
ltl |ltl_type|if_index|lc_type| vdc |v4_fpoelv5_fpoel base_fpoel_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. 이그레스 시 ELAM 패킷을 캡처합니다.패킷을 캡처하려면 IP 주소192.168.13.3에서 192.168.12.2으로 ping 응답을 보냅니다. 포트 채널 1 인터페이스(e3/1-2)에서 이그레스 키워드로 캡처를 설정해야 합니다. 인터페이스는 앞서 설명한 대로 인스턴스 0에 속합니다.

```
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-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

```

표시된 것처럼 소스 및 대상 인덱스는 모두 DBUS의 일부입니다(인그레스 캡처에 표시된 것과 다름).

```

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
```

인그레스 ELAM 패킷 캡처 후 디코딩된 소스 및 목적지 IP 주소가 정확합니다. 그러나 반환 트래픽이 캡처되므로 인그레스 ELAM과 비교할 때 방향은 확실히 반대입니다.

- 10. VLAN 10이 트래픽을 포워딩할지 여부를 확인하려면 N7K1의 포트 채널 1의 모듈 3에 대한 CBL(Color-Based Logic)을 확인합니다. CBL은 물리적 인터페이스별 로직이므로 포트 채널 번호가 아닌 N7K1의 포트 채널 1의 멤버 인터페이스 번호를 입력해야 합니다. 다음 출력에서는 VLAN 10이 예상한 대로 이를 전달하는 것을 확인할 수 있습니다.

CBL은 하드웨어 내 포트의 STP(Spanning Tree Protocol) 상태를 확인하는 데 사용됩니다. Sup에서 VLAN에 대해 STP를 확인할 때 인터페이스에 포워딩이 표시될 수 있지만 모듈은 트래픽을 차단합니다. 참고: 멤버 인터페이스(e3/1 및 e3/2)에 대해 CBL을 개별적으로 확인해야 합니다.

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

참고:이전 명령은 포트 채널 1에 대한 것입니다(모듈 3은 e3/1에 있음).

```
module-3# show hardware internal mac port 2 table cbl vlan
```

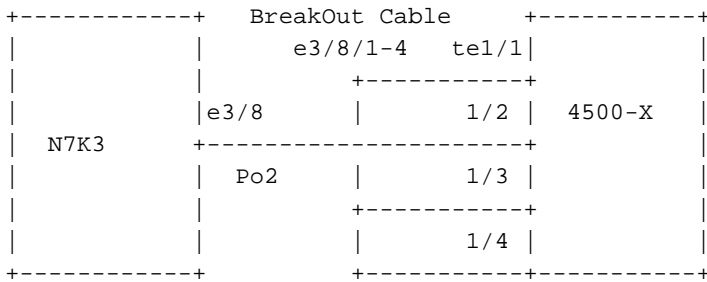
```
-----
|                                     INGRESS                                     |
| Disabled State | 0,2-9,11-4031,4036-4095 |
| Forwarding State | 1,10,4032-4035 |
| Blocked State |
| Learning State |
|-----|-----|
|                                     EGRESS                            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

참고:마찬가지로 이 명령은 CBL에서 포트 채널 2(e3/2)를 확인합니다.

브레이크아웃 케이블을 사용하여 F3 Series 모듈의 ELAM 문제 해결

브레이크아웃 케이블이 연결된 경우 F3 시리즈 모듈의 ELAM 절차는 일반 모듈 포트의 ELAM 절차와 다릅니다. 그러나 인덱스를 전면 패널 번호로 변환하려고 시도하는 동안 PIXM(Port Index Manager)의 확인과 관련하여 몇 가지 변경 사항이 있습니다. 이 경우 브레이크아웃 케이블에서 인터페이스가 수신됩니다.

다음은 이 섹션의 예제에 사용되는 토폴로지입니다.



```

Vlan20 SVI                               Vlan20 SVI
IP add 192.168.20.3                       IP add 192.168.20.1

```

이 예에서 브레이크아웃 케이블은 이더넷 인터페이스 3/8에 연결되어 40기가비트 포트를 4개의 10기가비트 포트에 분할합니다. 필요한 컨피그레이션은 이 섹션에서 참조할 수 있습니다.

```
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
-----
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

```

이전 출력에서 이더넷 인터페이스 3/7이 여전히 40기가비트 포트임을 확인할 수 있습니다. 그러나 이더넷 인터페이스 3/8은 4개의 10기가비트 포트에 나뉘어지며, 개별적으로 구성할 수 있습니다.

```
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
```

N7K3 SVI(Switched Virtual Interface) 20 IP 주소(192.168.20.3)에서 4500 SVI 20 IP 주소 (192.168.20.1)으로 패킷 캡처를 시작합니다. 패킷은 이그레스 시 N7K3에서 4500으로 캡처되고, 회신은 4500에서 N7K3으로 전송됩니다.

이전 섹션에서 설명한 대로 트리거를 적용하려면 플래너 인스턴스에 대한 지식이 있어야 합니다.이 출력은 모듈 3의 첨부 파일을 보여줍니다.

```
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

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

이 출력에서 포트 8은 플래너 인스턴스 3에 있습니다. 이제 인스턴스를 알고 있으므로 소스 및 대상 IP 주소를 통해 트리거를 배치할 수 있습니다.N7K3에서 4500으로 ping 요청을 캡처하므로 이그레스 ELAM이 됩니다.

```
module-3# elam asic flanker instance 3
```

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

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

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

N7K3에서 4500으로 ping이 시작됩니다.

```
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
```

ELAM 상태는 다음과 같습니다.

```
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
```

시퀀스 번호가 동일한지 확인합니다.

```
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
```

시퀀스 번호는 동일합니다. 이제 DBUS 및 RBUS 정보를 확인할 수 있습니다.

```
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-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	v1	: 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
```

흐름을 확인하기 위해 대상 및 소스 인덱스를 전면 패널 포트 로 변환합니다.

```
N7K3# show system internal pixm info lt1 0x400
0x0400 is in SUP In-band LTL range
```

이 출력은 소스 인덱스를 표시합니다. Sup에서 N7K3로 오는 핑거들 때문에 그것이 옳다는 것을 알고 있습니다. 다음 출력에는 N7K에서 VLAN 20을 허용하는 두 인터페이스 중 하나인 이그레스 인터페이스(e3/8/1)이 표시됩니다. 다른 인터페이스는 e3/8/4이며, STP로 인해 4500에서 차단됩니다.

```
N7K3# show system internal pixm info lt1 0x82
```

0x0082 is in DCE/FC pool

Member info

```

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

```

N7K에서 브레이크아웃 케이블로 생성된 포트의 CBL을 확인합니다.CBL을 확인하려면 새로 구성된 모든 포트에 대한 하드웨어 포트 번호가 있어야 합니다.

참고:스위치에 인터페이스 e3/8이 없습니다.새로 구성된 포트만 나타납니다.

```

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

```

브레이크아웃 케이블이 사용되고 e3/8 인터페이스가 스위치에 없으므로 하드웨어 포트 번호 변경을 가져오는 데 사용되는 계산입니다.분할을 지원하는 모든 모듈의 경우 하드웨어 포트 번호 지정이 다릅니다.먼저 포트가 브레이크아웃을 지원하는지 확인해야 합니다.

```

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

```

표시된 대로 포트 e3/7은 브레이크아웃을 지원하므로 대역폭이 4개의 10기가비트 포트에 분할될 수 있습니다.마찬가지로, 100기가비트 포트가 있는 다른 F3 Series 모듈은 각각 10기가비트로 10개의 포트에 분할하거나 초과 서브스크립션을 통해 40기가비트의 3개의 포트에 분할할 수 있습니다. 이는 모듈에 따라 다릅니다.

이 예에서 F3 Series 모듈에는 40기가비트 포트가 있으며 각 포트는 각각 4개의 포트에 나눌 수 있으므로, 각 포트의 하드웨어 포트 번호는 0부터 3, 4-7, 8-11...40-43, 44-47입니다.첫 번째 예제에 대해 포트에 브레이크아웃 케이블이 있는 경우 해당 하드웨어 포트 번호 지정은 0, 1, 2 및 3이 됩니다. 브레이크아웃 케이블이 없는 경우 해당 하드웨어 포트 번호 지정은 0(1, 2 및 3은 활성화되지 않음)이 됩니다.

상위 포트는 e3/8이므로 브레이크아웃 케이블 없이 사용할 경우 하드웨어 포트 번호는 28이고 브레이크아웃 케이블과 함께 사용할 경우 28, 29, 30 및 31입니다.이 명령 출력은 활성 하드웨어 포트 (0부터 시작)를 표시합니다.

```

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:
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

고장난 포트 하드웨어 포트 번호는 28이며, 현재는 4개로 분할됩니다(28-32). 이제 모듈 3을 연결하고 하드웨어에서 CBL을 확인할 수 있습니다.

```
N7K3# attach module 3
```

```
Attaching to module 3 ...
```

```
To exit type 'exit', to abort type '$.'
```

```
module-3#
```

F3 Series 모듈에서는 포트 번호를 1기반 배율에 따라 포맷해야 합니다.따라서 29, 30, 31 및 32를 입력해야 합니다.

```
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.
```

다음은 VLAN 전달 상태를 확인 및 확인하기 위해 이더넷 인터페이스 3/8/1에 대해 실행 중인 컨피그레이션입니다.

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

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

```
-----
```

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

```
-----
```

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

```
-----
```

다음은 VLAN 전달 상태를 확인 및 확인하기 위해 이더넷 인터페이스 3/8/2에 대해 실행 중인 컨피그레이션입니다.

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

```
module-3# show hardware internal mac port 30 table cbl vlan
```

```
-----
```

INGRESS	
Disabled State	0,2-29,31-39,41-4031,4036-4095
Forwarding State	30,40,4032-4035
Blocked State	1

```
-----
```



```

| Learning State |
-----
|
|                                     EGRESS
| Disabled State | 0,2-29,31-39,41-4031,4036-4095
| Forwarding State | 30,40,4032-4035
| Blocked State | 1
| Learning State |
-----

```

다음은 VLAN 전달 상태를 확인하고 확인하기 위해 이더넷 인터페이스 3/8/3에 대해 실행 중인 컨피그레이션입니다.

```

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

```

```

module-3# show hardware internal mac port 31 table cbl vlan

```

```

|                                     INGRESS
| Disabled State | 0,2-49,51-4031,4036-4095
| Forwarding State | 50,4032-4035
| Blocked State | 1
| Learning State |
-----
|                                     EGRESS
| Disabled State | 0,2-49,51-4031,4036-4095
| Forwarding State | 50,4032-4035
| Blocked State | 1
| Learning State |
-----

```

다음은 VLAN 포워딩 상태를 확인 및 확인하기 위해 이더넷 인터페이스 3/8/4에 대해 실행 중인 컨피그레이션입니다(구성된 모든 VLAN이 허용됨).

```

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

```

```

module-3# show hardware internal mac port 32 table cbl vlan

```

```

|                                     INGRESS
| Disabled State | 0,2-9,11-19,21-29,31-39,41-49,51-59,61-669,671-4031
| Disabled State | 4036-4095
| Forwarding State | 1,20,30,40,50,60,670,4032-4035
| Blocked State | 10
| Learning State |
-----
|                                     EGRESS
| Disabled State | 0,2-9,11-19,21-29,31-39,41-49,51-59,61-669,671-4031
| Disabled State | 4036-4095
| Forwarding State | 1,20,30,40,50,60,670,4032-4035
| Blocked State | 10
| Learning State |
-----

```

CBL은 올바른 VLAN이 전달되었음을 보여줍니다.

하드웨어 포트 번호를 얻기 위해 **show hardware internal error module <module number>** 명령을 사용할 수 있습니다. 이 명령은 **show interface x/y** 명령 출력에 나타나지 않는 내부 삭제를 확인해야 할 때 유용합니다. 예를 들면 다음과 같습니다.

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	00000000002160959	32 -
647	igr in upm: pkts with symbol/sequence error rcvd	00000000003722562	32 -
648	igr in upm: pkts with error	0000000000000002	32 -