

# Nexus 7000シリーズスイッチでVXLANカプセル化パケットをキャプチャするためにELAMを導入する

## 内容

---

[はじめに](#)

[背景説明](#)

[トポロジ](#)

[トリガーの設定](#)


[結果の解釈](#)

[関連情報](#)

---

## はじめに

このドキュメントでは、Nexus 7000シリーズスイッチでVXLANカプセル化パケットをキャプチャするためにEmbedded Logic Analyzer Module(ELAM)を導入する方法について説明します。

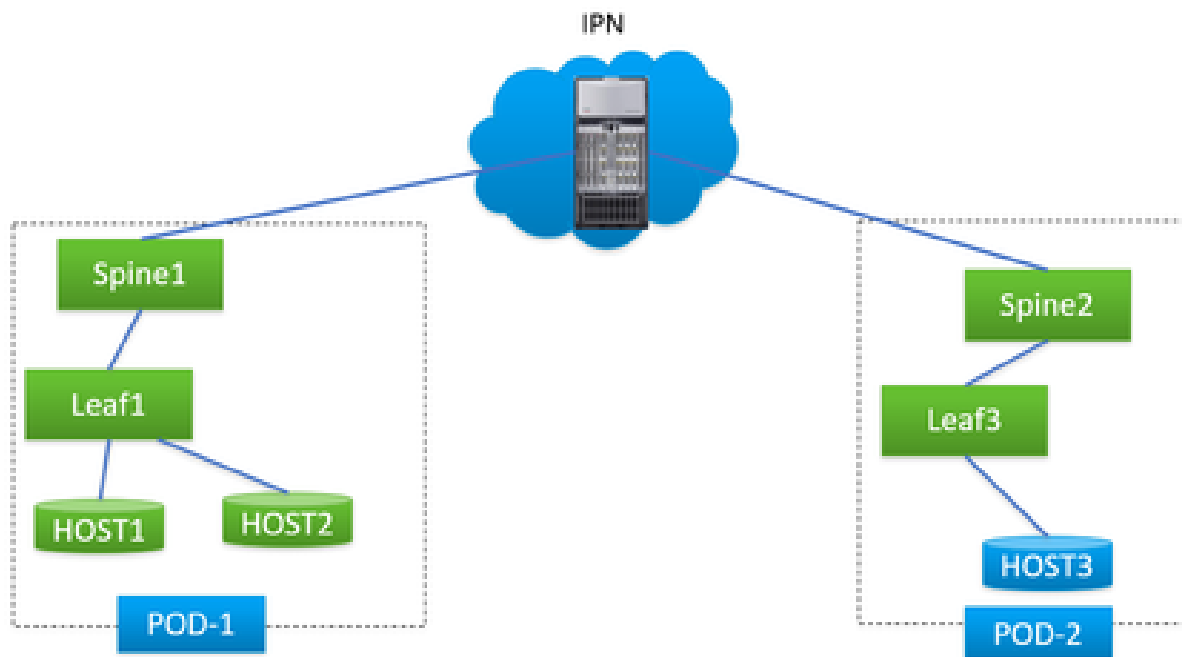
 ヒント:ELAMの概要については、『[ELAMの概要](#)』を参照してください。

---

## 背景説明

現在、多くのユーザがACI MPOD/MSITEの展開にIPN/ISN中継デバイスとしてN7Kを利用しています。ただし、N9Kと比較すると、N7Kには、豊富なOuter(I2(vntag)|I3|I4)-inner(I2|I3|I4)-iethの組み合わせに基づいてELAMトリガーを設定する堅牢な機能がありません。その結果、ELAMの観点から、特定のVXLANカプセル化パケットがIPNエッジでN7Kに到達しているかどうかを判断することが困難になります。このドキュメントでは、この課題に対処する方法の概要を説明します。

## トポロジ



このシナリオでは、単純なACI MPODトポロジを図示します。ここで、IPNはF3カードを搭載したN7Kです。HOST1とHOST2はpod1にあり、HOST3はpod2にあります。HOST1はHOST3と通信できるが、HOST2は通信できない。ACIエンジニアが実施したトラブルシューティングの結果、HOST2からHOST3へのパケットはpod1のspine1からN7Kに送信されたが、pod2のspine2には受信されなかったことが判明しました。これはACIスパインのELAMを通じて確認されたため、パケットがN7Kでドロップされた疑いがあります。

ACIスパインでのELAMの結果のみに基づいて、問題をN7Kに明確に関連付けることはできません。もちろん違います。出力spine1のELAMは、パケットをN7Kに送信したことを示していますが、これはパケットが物理的にN7Kに到達したことを保証するものではありません。これは、下位レイヤの問題により、ELAMサイクルの後でもパケットがドロップされる可能性があるためです。ただし、N7K側でこれらの特定の packets をELAMすると、問題に関係する正しいデバイスを正確に特定するのに役立ちます。

## トリガーの設定

「l3-packet-length」は、N7Kプラットフォーム上のほぼすべての世代のLCに対する有効なELAMトリガーです。したがって、これを利用してIPN N7KでELAM状態を確立してみましょう。このタスクには、次に示すように、指定されたパケット長のテストパケットを送信するようにホストを制御することが含まれます。

```
<#root>
```

```
#
```

```
ping 172.28.1.20 packet-size 777
```

```
PING 172.28.1.20 (172.28.1.20): 777 data bytes
```

```
785 bytes from 172.28.1.20: icmp_seq=0 ttl=252 time=1.246 ms
785 bytes from 172.28.1.20: icmp_seq=1 ttl=252 time=0.846 ms
785 bytes from 172.28.1.20: icmp_seq=2 ttl=252 time=0.84 ms
785 bytes from 172.28.1.20: icmp_seq=3 ttl=252 time=0.814 ms
785 bytes from 172.28.1.20: icmp_seq=4 ttl=252 time=0.817 ms
```

```
--- 172.28.1.20 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0.00% packet loss
```

```
round-trip min/avg/max = 0.814/0.912/1.246 ms
```

PINGユーティリティはあらゆるタイプのOSに統合されており、使用しているOSによってパラメータにわずかな違いしかありません。重要なポイントの1つは、OSでPINGを開始するときに指定するパケットサイズに注意を払うことです。この例では、777Bは純粋なデータ長を表しており、805Bの最終的なIP長を取得するために、さらに8B (ICMPヘッダー) と20B (IPヘッダー) が必要です。VXLANカプセル化 (追加の50 Bオーバーヘッド) の後、パケットは855 BでN7Kに到達すると予測できます。ELAMで設定してみましょう。

この例では、spine1に接続するインターフェイスはE7/1で、E7/4はspine2に接続します。

```
<#root>
```

```
#
```

```
show module 7
```

Mod	Ports	Module-Type	Model	Status
7	12	10/40 Gbps Ethernet Module	N7K-F312FQ-25	ok

```
module-7#
```

```
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	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

```

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

```

そのため、インスタンス0で設定する必要があります。

```
<#root>
module-7# elam asic flanker instance 0

module-7(fln-elam)# layer2

module-7(fln-l2-elam)#
trigger dbus ipv4 ingress if l3-packet-length 855

module-7(fln-l2-elam)#
trigger rbus ingress if trig

module-7(fln-l2-elam)# start
module-7(fln-l2-elam)# status

ELAM Slot 7 instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if l3-packet-length 855
L2 DBUS: Triggered

ELAM Slot 7 instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
L2 RBUS: Triggered
```

## 結果の解釈

```
<#root>
module-7(fln-l2-elam)#
show dbus

cp = 0x10084d00, buf = 0x10084d00, end = 0x10091050
-----

Flanker Instance 00 - Capture Buffer On L2 DBUS:

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

is\_l2\_egress: 0x0000, data\_size: 0x023

[000]: 14f4a000 08010000 00000000 6d200800 00006000 00000000 01800100 00000000 00000000 00000000 00003000  
0590 00990000 00000000 00000000 00000005 88405000 00000000 00000000 00000000 00000000 00000000 00000000  
a4 2dbeef00

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
14-protocol	: 0x11		
df	: 0x0		
mf	: 0x0	frag	: 0x0
t1	: 0x1f	13-packet-length	: 0x357
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
tr	: 0x0	outer-cos	: 0x0
inner-cos	: 0x0	vqi-valid	: 0x0
vqi	: 0x0	packet-length	: 0x369

vlan	: 0x4	destination-index	: 0x0
source-index	: 0x30		
bundle-port	: 0x0		
acos	: 0x0	outer-drop-eligibility:	0x0
inner-drop-eligibility:	0x0	sg-tag	: 0x0
rbh	: 0x0	vs1-num	: 0x0
inband-flow-creation-deletion:	0x0	ignore-qoso	: 0x0
ignore-qosi	: 0x0	ignore-aclo	: 0x0
ignore-acli	: 0x0	index-direct	: 0x0
no-stats	: 0x0	dont-forward	: 0x0
notify-index-learn	: 0x1	notify-new-learn	: 0x1
disable-new-learn	: 0x0	disable-index-learn	: 0x0
dont-learn	: 0x0	bpdu	: 0x0
ff	: 0x0	rf	: 0x0
ccc	: 0x0	l2	: 0x0
rdt	: 0x0	dft	: 0x0
dfst	: 0x0	status-ce-1q	: 0x0
status-is-1q	: 0x1	trill-encap	: 0x0
mim-valid	: 0x0	dtag-ttl	: 0x0
dtag-ftag	: 0x0	valid	: 0x1
erspan-kpa-valid	: 0x0	recir-shim-vxlan-src-peer-id:	0x0
vn-valid	: 0x0	source-vif	: 0x0
destination-vif	: 0x0	vn-p	: 0x0
sequence-number	: 0x60	v1	: 0x0
inner-de-valid	: 0x0	de-cfi	: 0x0
second-inner-cos	: 0x0	tunnel-type	: 0x2

-----  
UDP OTV/LISP TUNNEL BNDL  
-----

vlan-tag-valid:	0x0	segment-id-valid:	0x0
v1:	0x0	de:	0x0

```
sgt-valid: 0x0          inner-ip-ttl: 0x0
ip-da-multicast: 0x0
lisp-inst-id: 0x2c8004

lisp-flags: 0xc8       isis-mac-da-valid: 0x0
type: 0x0
shim-valid           : 0x0
segment-id-valid    : 0x0          copp                : 0x0
dti-type-vpnid     : 0x0          segment-id          : 0x0
ib-length-bundle   : 0x58840     m1h-type            : 0x5
ulh-type            : 0x4

source-ipv4-address: 10.0.200.64
```

```
destination-ipv4-address: 10.1.224.67
```

```
mim-destination-mac-address : 0000.0000.0000
```

```
mim-source-mac-address : 0000.0000.0000
```

```
destination-mac-address : 00c1.b1c9.c2c4
```

```
source-mac-address : 000d.0d0d.0d0d
```

I3パケット長がトリガーとして使用されるため、キャプチャを意図しないバックグラウンドパケットによってELAMがトリガーされる可能性があります。したがって、キャプチャ結果のダブルクロスチェックには、キャプチャ内の他のフィールドを使用する必要があります。これにより、キャプチャされたパケットが、送信元IP(sip)、宛先IP(dip)、存続可能時間(ttl)、送信元インデックスなどのフィールドを含む意図した基準に一致することが保証されます。興味深いことに、N7KはVXLAN VNIDをトリガーとして使用することをサポートしていませんが、アウトプットインタープリタでは、「lisp-inst-id: 0x2c8004」フィールドがVXLANヘッダーのVNIDに対応しています。

```
<#root>
```

```
module-7(f1n-l2-elam)# dec
```

```
0x2c8004
```



2916356

Leaf3#

show system internal epm endpoint ip 172.28.1.20

MAC : 0000.2222.1202 ::: Num IPs : 1

IP# 0 : 172.28.1.20 ::: IP# 0 flags : host-tracked| ::: l3-sw-hit: Yes ::: flags2 :

Vlan id : 186 ::: Vlan vnid : 11494 ::: VRF name : zixu:vrf

BD vnid : 16482209 :::

VRF vnid : 2916356

/\* Confirming the VNID from ACI LEAF side \*/

Phy If : 0x1a00b000 ::: Tunnel If : 0

Interface : Ethernet1/12

Flags : 0x80005c04 ::: sclass : 16388 ::: Ref count : 5

EP Create Timestamp : 01/22/2021 15:42:49.243582

EP Update Timestamp : 02/08/2021 11:26:52.882308

EP Flags : local|IP|MAC|host-tracked|sclass|timer|

module-7(fln-12-elam)#

show rbus

cp = 0x100a96fc, buf = 0x100a96fc, end = 0x100b5a4c

-----  
Flanker Instance 00 - Capture Buffer On L2 RBUS:

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

is\_l2\_egress: 0x0000, data\_size: 0x018

[000]: 0015cb30 0000006d 20000000 03000000 00000000 00000000 00000014 2d8000a0 3c3c0000 00000000 02000000  
0000 00000400 00008000 005d0000 001e0002 2bd7c0cf f96002a0 000000ba

Printing packet 0

-----  
L2 RBUS INGRESS CONTENT  
-----

pad	: 0x572c	valid	: 0x1
l2-rbus-trigger	: 0x1	sequence-number	: 0x60
rit-ipv4-id	: 0x0	ipv4-tunnel-encap	: 0x0
rit-mp1s-rw	: 0x0	m12-ptr	: 0x0
m13-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	m12-delta-length	: 0x0
m12-delta-length-plus	: 0x0	m13-delta-length	: 0x0
m13-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	: 0x42d8
vntag-p	: 0x0	frr-recirc	: 0x0
ingress-lif	: 0x5	earl-proxy-vld	: 0x0
md-di-vld	: 0x0	rc	: 0x0
segment-id-valid	: 0x0	t11-out	: 0x1e
t11-mid	: 0x1e	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
tr	: 0x0	de	: 0x0
cos	: 0x0	inner-drop-eligibility	: 0x0

```

inner-cos          : 0x0          acos          : 0x0

di-1tl-index      : 0x3c
    13-multicast-di : 0x3c

source-index      : 0x30          vlan          : 0x4
index-direct     : 0x0          di1-valid     : 0x1
vqi              : 0x4a          di2-valid     : 0x0
v5-fpoe-idx      : 0x0          di2-fpoe-idx : 0x0
13-multicast-v5  : 0x0          dft           : 0x0
dfst             : 0x0          13-learning-ff : 0x0
result-rbh       : 0x40          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
12-copy-cause   : 0x0          13-rit-ptr   : 0x5d
sg-tag           : 0x0          trill-nh-id   : 0x0
ttl-in           : 0x1e          fc-up         : 0x0
up-did           : 0x0          did           : 0x22bd
up-sid           : 0x0          sid           : 0xf819ff
shim-12-tunnel-encap: 0x0          shim-ls-hash  : 0xb
shim-rc          : 0x0          shim-lif      : 0x5
shim-replication-pkt: 0x0          shim-router-mac : 0x1
shim-mark-enable : 0x0          shim-qos-group-id : 0x0

shim-destination-table-index: 0x5d          shim-acos-preserve : 0x0

mim-destination-mac-address : 0000.0000.0000
mim-source-mac-address : 0000.0000.0000

```

```

module-7(f1n-12-elam)#

```

```

show system internal pixmc info 1tl-cb 1tl 0x30

```

```
lt1 | lt1_type | if_index | lc_type | vdc | v4_fpoe | v5_fpoe | base_fpoe_idx | flag
0x0030 | 5 |
Eth7/1
| 2 | 4 | 0x00 | 0x00 | 0x0000 | 0x0
```

```
module-7(fln-12-elam)#
```

```
show system internal pixmc info lt1-cb lt1 0x3c
```

```
lt1 | lt1_type | if_index | lc_type | vdc | v4_fpoe | v5_fpoe | base_fpoe_idx | flag
0x003c | 5 |
Eth7/4
| 2 | 4 | 0x00 | 0x00 | 0x0000 | 0x0
```

同じ方法が、BDをフラッディングするACI内でブロードキャスト、不明なユニキャスト、およびマルチキャスト(BUM)パケットをキャプチャしようとする場合にも機能します。静的ARPを設定して、ホスト上に存在しないMACをポイントし、同じようにPINGを起動する必要があります。

## 関連情報

さまざまなN7K LCでELAMを使用する方法の詳細については、次のリンクを参照できます。

- [ELAM の概要](#)
- [N7K MシリーズモジュールのELAM手順](#)
- [N7K F1モジュールのELAM手順](#)
- [N7K F2モジュールのELAM手順](#)
- [N7K M3モジュールのELAM手順](#)
- [シスコのテクニカルサポートとダウンロード](#)

## 翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。