

Nexus 9000云扩展ASIC(Tahoe)NX-OS ELAM

目录

[简介](#)

[适用硬件](#)

[Nexus Tahoe ASIC ELAM流程](#)

[拓扑](#)

[第1步 — 检验端口的ASIC、Slice和SrcId](#)

[步骤2 — 连接到模块](#)

[第3步 — 进入ELAM配置模式并指定第1步中的正确ASIC](#)

[第4步 — 配置ELAM触发器](#)

[第5步 — 使用SRC和DEST IP设置ELAM触发器](#)

[第6步 — 启动ELAM](#)

[第7步 — 验证是否触发了ELAM，然后查看结果](#)

[查看NX-OS 7.0\(3\)I5\(2\)之前版本的ELAM结果](#)

[VXLAN封装的ELAM:](#)

[NX-OS 7.0\(3\)I7\(2\)后的ELAM配置](#)

[简介](#)

mELAM(alyzer)

提示：有关ELAM的概述，请参阅[ELAM概述](#)文档。

适用硬件

本文档中介绍的过程仅适用于以下硬件：

N9K-C93180YC-EX	N9K-C92304QC
N9K-X9736C-EX	N9K-C92300YC
N9K-C93108TC-EX	N9K-X9788TC-FX
N9K-X9732C-EX	N9K-X97284YC-FX
N9K-X97160YC-EX	N9K-C93180YC-FX
N9K-C93180LC-EX	N9K-C93108TC-FX
N9K-C92160YC-X	N9K-C9348GC-

FXP

N9K-C9272Q N9K-X9732C-FX

N9K-C9236C N9K-C9336C-FX2

N9K-C93240YC- N9K-C93300YC-
FX2 FX2

N9K-C9364C N9K-C9332C

Nexus Tahoe ASIC ELAM流程

拓扑



第1步 — 检验端口的ASIC、Slice和SrcId

```
N9K-C92160YC-X-2# show hardware internal tah interface e1/49
IfIndex: 436232192
DstIndex: 5952
IfType: 26
Asic: 0 <<<<<<<<< Asic: 0
AsicPort: 56
SrcId: 48 <<<<<<<< Slice: 1 <<<<<<<<< PortOnSlice: 24
```

警告：ELAM应仅用于一个终端窗口，因为您维护每个片段的全局内容、lu-a2d等。

例如，端口通道(PO)可能有两个链路，即Eth 1/53 (对应于片0)和Eth 1/54 (对应于片1)。一次在两个不同的终端窗口上为不同的片设置ELAM无用，因为后一个片(例如片1)将覆盖第一个片(片0)，最终在两个终端窗口上获得相同的结果。

您可以通过以下方式仔细检查此信息：

```
N9K-C92160YC-X-2# show system internal ethpm info interface e1/49 | i i src
IF_STATIC_INFO: port_name=Ethernet1/49,if_index:0x1a006000,ltl=5952,slot=0,
nxos_port=192,dmod=1,dpid=56,
```


事项中记录的准则和限制“系统路由模式：template-vxlan-scale”不适用于Cisco NX-OS版本7.0(3)I5(2)及更高版本。将VXLAN BGP EVPN与Cisco NX-OS版本7.0(3)I4(x)或NX-OS版本7.0(3)I5(1)结合使用时，“系统路由模式：以下硬件平台需要template-vxlan-scale”：Cisco Nexus 9300-EX交换机带X9700-EX线卡的Cisco Nexus 9500交换机更改“系统路由模式”需要重新加载交换机。

经历CPU传送的流量示例：

```
module-1(TAH-elam-insel6)# report
```

```
SUGARBOWL ELAM REPORT SUMMARY  
=====
```

```
Incoming Interface: Eth1/3  
Src Idx : 0x9, Src BD : 23  
Outgoing Interface Info: dmod 1, dpid 72  
Dst Idx : 0x601, Dst BD : 802
```

```
Packet Type: IPv4
```

```
Dst MAC address: B0:8B:CF:A3:D0:4B  
Src MAC address: 00:10:DB:FF:10:00  
.1q Tag0 VLAN: 23, cos = 0x0
```

```
Dst IPv4 address: 192.0.2.1  
Src IPv4 address: 192.0.2.2  
Ver      = 4, DSCP      = 2, Don't Fragment = 1  
Proto    = 6, TTL       = 49, More Fragments = 0  
Hdr len  = 20, Pkt len  = 60, Checksum      = 0x63c3
```

```
L4 Protocol : 6  
TCP Dst Port : 80  
TCP Src Port : 46340
```

```
Sup hit: 1, Sup Idx : 2720 <<---- CPU punt, use below CLI to resolve the meaning of Sup Idx
```

```
Drop Info:  
-----
```

```
LUA:  
LUB:  
LUC:  
LUD:  
Final Drops:
```

```
# show system internal access-list sup-redirect-stats all | grep 2720 2720 copp-system-p-acl-  
http 63
```

查看NX-OS 7.0(3)I5(2)之前版本的ELAM结果

+它是否有dot1q报头？

```
module-1(TAH-elam-insel6)# report | grep pr_lu_vec_l2v.qtag0 GBL_C++: [MSG] pr_lu_vec_l2v.qtag0_vld:  
0x1 << dot1q yes? 0x1 GBL_C++: [MSG] pr_lu_vec_l2v.qtag0_cos: 0x0 GBL_C++: [MSG]  
pr_lu_vec_l2v.qtag0_de: 0x0 GBL_C++: [MSG] pr_lu_vec_l2v.qtag0_vlan: 0xA << VL 10  
+检查VLAN:
```

```
module-1(TAH-elam-insel6)# report | grep -1 fpx_lookup_vec.lkup.macsakey.key.fid
```

```
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.vld: 0x1
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.fid_type: 0x0
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.fid_vld: 0x0
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.fid: 0xA << dec 0xa = VL 10
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
```

+检查SRC MAC (您实际上也可在上一步中看到这一点) :

```
module-1(TAH-elam-insel6)# report | grep -i fpx_lookup_vec.lkup.macsakey.key.mac
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715 << 00fe.c80e.2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
GBL_C++: [MSG] fpx_lookup_vec.lkup.macsakey.key.mac: 0xFEC80E2715
```

+这是新学习内容吗?

```
module-1(TAH-elam-insel6)# report | grep -1 fpx_lookup_vec.sa_notify_info
GBL_C++: [MSG] fpx_lookup_vec.lkup.ptvec.misc1.tcp_flags: 0x0
GBL_C++: [MSG] fpx_lookup_vec.sa_notify_info: 0x5200000C060
GBL_C++: [MSG] fpx_lookup_vec.sa_notify_info.enable: 0x0 << This will be set to 0x1 for learning
to happen
GBL_C++: [MSG] fpx_lookup_vec.sa_notify_info.conv_learn_only: 0x0
```

+检查SRC和DST IP:

```
module-1(TAH-elam-insel6)# report | grep vec_l3v.ip.*a
GBL_C++: [MSG] pr_lu_vec_l3v.ip.da: 0x0000000000000000c0000201 << DST IP: 192.0.2.1
GBL_C++: [MSG] pr_lu_vec_l3v.ip.sa: 0x0000000000000000c0000202 << SRC IP: 192.0.2.2
```

+验证入口SRC_ID:

```
module-1(TAH-elam-insel6)# report | egrep SRC
GBL_C++: [MSG] SRCID: 0x30
```

```
module-1(TAH-elam-insel6)# report | grep vec.ihdr.ieth.hdr.src_idx
GBL_C++: [MSG] lurw_vec.ihdr.ieth.hdr.src_idx: 0xA9 << sh hardware internal tah int e1/49 | i i niv_idx
```

+如果ELAM未触发,其显示如下:

```
module-1(TAH-elam-insel6)# report
GBL_C++: [MSG] tahusd_elam_wrapper_report:27d:asic type 5 inst 0 slice 1 a_to_d 1 insel 6
outsel 0
GBL_C++: [MSG] Inside tahusd_elam_wrapper_dav_report
GBL_C++: [MSG] ELAM not yet triggered <<<<<<
```

VXLAN封装的ELAM:

由于VXLAN数据包将被封装,因此ELAM需要在内部报头而非外部报头上触发 — 请参阅以下示例了解ARP帧:

```
module-1# debug platform internal tah elam asic 0  
module-1(TAH-elam)# trigger init asic 0 slice 1 in-select 7 out-select 0 use-src-id 48  
module-1(TAH-elam-insel7)# reset  
module-1(TAH-elam-insel7)# set inner arp source-ip-addr 192.0.2.2 target-ip-addr 192.0.2.1  
module-1(TAH-elam-insel7)# start  
module-1(TAH-elam-insel7)# report
```

NX-OS 7.0(3)I7(2)后的ELAM配置

在NX-OS 7.0(3)I7(2)之后，ELAM现在无需指定ASIC或切片编号即可全局触发 — 请参阅以下示例：

```
Nexus-9K# debug platform internal tah elam  
Nexus-9K(TAH-elam)# trigger init  
Nexus-9K(TAH-elam-insel6)# reset  
Nexus-9K(TAH-elam-insel6)# set outer ipv4 dst_ip 192.0.2.1 src_ip 192.0.2.2  
Nexus-9K(TAH-elam-insel6)# start  
Nexus-9K(TAH-elam-insel6)# report
```

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。