

# Nexus 7000 M3模块ELAM程序

## 目录

[简介](#)

[拓扑](#)

[确定入口转发引擎](#)

[配置触发器](#)

[开始捕获](#)

[解释结果](#)

[其他验证](#)

## 简介

本文档介绍在Cisco Nexus 7700(N7700)M3模块上执行ELAM所用的步骤，解释最相关的输出，并说明如何解释结果。

提示：有关ELAM的概述，[请参阅ELAM概述文档](#)。

## 拓扑



在本示例中，VLAN 2500(10.0.5.101)上的主机端口Eth4/1向VLAN 55(10.0.3.101)上的主机端口Eth3/5发送Internet控制消息协议(ICMP)请求。ELAM用于捕获从10.0.5.101到10.0.3.101的单个数据包。请记住，ELAM允许您捕获单个帧。

要在N7K上执行ELAM，您必须首先连接到相应的模块（这需要网络管理员权限）：

```
N7700# attach module 4
Attaching to module 4 ...
module-4#
```

## 确定入口转发引擎

流量应通过端口Eth4/1进入交换机。当您检查系统中的模块时，您会看到模块4是M3模块。请务必记住，N7K是完全分布式的，并且模块（而不是管理引擎）会为数据平面流量做出转发决策。

```
N7700# show module
```

```
Mod  Ports  Module-Type                Model                Status
-----
1    12     100 Gbps Ethernet Module  N77-F312CK-26      ok
3 48 1/10 Gbps Ethernet Module N77-M348XP-23L ok 4    24    10/40 Gbps Ethernet Module
N77-M324FQ-25L            ok
5     0     Supervisor Module-2      N77-SUP2E          active *
6     0     Supervisor Module-2      N77-SUP2E          ha-standby
7    24    10/40 Gbps Ethernet Module N77-F324FQ-25      ok
```

```
Mod  Sw                Hw
-----
1    7.3(0)DX(1)      1.1
3 7.3(0)DX(1) 1.1 4 7.3(0)DX(1) 1.0 5 7.3(0)DX(1) 1.2 6 7.3(0)DX(1) 1.2 7 7.3(0)DX(1) 1.0
```

对于M系列模块，在内部代号为F4的第2层(L2)转发引擎(FE)上执行ELAM。请注意，L2 FE数据总线(DBUS)包含L2和第3层(L3)查找之前的原始报头信息以及结果总线(RBUS))包含L3和L2查找后的结果。

N7K M3模块可以为每个模块使用多个FE，因此您必须确定端口Eth4/1上用于FE的F4 ASIC。输入此命令以验证：

```
module-4# show hardware internal dev-port-map
```

(some output omitted)

```
----- CARD_TYPE: 24 port 40G >Front
Panel ports:24 ----- Device name Dev
role Abbr num_inst: ----- > SLF L3
Driver DEV_LAYER_3_LOOKUP L3LKP 4 > SLF L2FWD driver DEV_LAYER_2_LOOKUP L2LKP 4
+-----+
+-----+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++-----+
+-----+
FP port |  PHYS |  MAC_0 |  RWR_0 |  L2LKP |  L3LKP |  QUEUE | SWICHF
  1      |      0   |    0   |    0   |    0   |    0   |    0   |  0,1
  2      |      0   |    0   |    0   |    0   |    0   |    0   |  0,1
  3      |      0   |    0   |    0   |    0   |    0   |    0   |  0,1
```

在输出中，您可以看到端口Eth4/1位于F4(L2LKP)实例0上。在N77-M312CQ-26L模块上，每个端口组中都有2个端口的6 F4 ASIC。在N77-M324FQ-25L模块上，每个端口组有4个F4 ASIC，带6个端口。N77-M348XP-23L模块有2个F4 ASIC，每个端口组中有12个端口。

**注意：**与F系列模块一样，M3模块ELAM语法使用基于0的值。M1和M2模块使用基于1的值，但情况并非如此。

```
module-4# elam asic f4 instance 0
```

```
module-4(f4-elam)# layer2
```

```
module-4(f4-l2-elam)#
```

## 配置触发器

F4 ASIC支持IPv4、IPv6等的ELAM触发器。ELAM触发器必须与帧类型对齐。如果该帧是IPv4帧，则触发器也必须是IPv4。IPv4帧不会用其他触发器捕获。IPv6也适用同样的逻辑。

使用Nexus操作系统(NX-OS)，您可以使用问号字符来分隔ELAM触发器：

```
module-4(f4-l2-elam)# trigger dbus ipv4 ingress if ?  
(some output omitted)  
destination-index Destination-index  
destination-ipv4-address Destination ipv4 address  
destination-ipv4-mask Destination ipv4 mask  
destination-mac-address Destination mac address  
l4-protocol L4 protocol  
source-index Source-index  
source-ipv4-address Source ipv4 address  
source-ipv4-mask Source ipv4 mask  
source-mac-address Source mac address
```

在本例中，根据源IPv4地址和目的IPv4地址捕获帧，因此仅指定这些值。

**F4需要DBUS和RBUS单独的触发器。**

以下是DBUS触发器：

```
module-4(f4-l2-elam)# trigger dbus ipv4 ingress if source-ipv4-address  
10.0.5.101 destination-ipv4-address 10.0.3.101
```

以下是RBUS触发器：

```
module-4(f4-l2-elam)# trigger rbus ingress result if tr 1
```

## 开始捕获

现在，已选择入口FE并配置了触发器，您可以开始捕获：

```
module-4(f4-l2-elam)# start
```

要检查ELAM的状态，请输入**status**命令：

```
module-4(f4-l2-elam)# status  
ELAM Slot 4 instance 0: L2 DBUS/LBD Configuration: trigger dbus ipv4 ingress if  
source-ipv4-address 10.0.5.101 destination-ipv4-address 10.0.3.101  
L2 DBUS/LBD: Configured  
ELAM Slot 4 instance 0: L2 RBUS Configuration: trigger rbus ingress result if tr 1  
L2 RBUS: Configured  
L2 BIS: Unconfigured  
L2 BPL: Unconfigured  
L2 EGR: Unconfigured  
L2 PLI: Unconfigured  
L2 PLE: Unconfigured
```

FE收到与触发器匹配的帧后，ELAM状态显示为“已触发”：

```
module-4(f4-l2-elam)# status  
ELAM Slot 4 instance 1: L2 DBUS/LBD Configuration: trigger dbus ipv4 ingress if  
source-ipv4-address 10.0.5.101 destination-ipv4-address 10.0.3.101  
L2 DBUS/LBD: Triggered  
ELAM Slot 4 instance 1: L2 RBUS Configuration: trigger rbus ingress result if tr 1  
L2 RBUS: Triggered  
L2 BIS: Unconfigured  
L2 BPL: Unconfigured  
L2 EGR: Unconfigured
```

L2 PLI: Unconfigured  
L2 PLE: Unconfigured 7

## 解释结果

要显示ELAM结果，请输入**show dbus** 和**show rbus** 命令。如果有大量流量与相同的触发器匹配，则DBUS和RBUS可能会在不同帧上触发。因此，检查DBUS和RBUS数据上的内部序列号以确保它们匹配非常重要：

```
module-4(f4-l2-elam)# show dbus | i seq  
port-id : 0x0 sequence-number : 0x868  
module-4(f4-l2-elam)# show rbus | i seq  
de-bri-rslt-valid : 0x1 sequence-number : 0x868
```

以下是与本示例最相关的ELAM数据的摘要（省略部分输出）：

```
module-4(f4-l2-elam)# show dbus  
-----  
LBD IPV4  
-----  
ttl : 0xff l3-packet-length : 0x54  
destination-address: 10.0.3.101  
source-address: 10.0.5.101  
-----  
packet-length : 0x66 vlan : 0x9c4  
segid-lsb : 0x0 source-index : 0xe05  
destination-mac-address : 8c60.4f07.ac65  
source-mac-address : 8c60.4fb7.3dc2  
port-id : 0x0 sequence-number : 0x868  
  
module-4(f4-l2-elam)# show rbus  
-----  
L2 RBUS RSLT CAP DATA  
-----  
de-bri-rslt-valid : 0x1 sequence-number : 0x868  
vlan : 0x37 rbh : 0x65  
cos : 0x0 destination-index : 0x9ed
```

使用DBUS数据，您可以验证该帧是否在VLAN 2500上收到，源MAC地址为**8c60.4fb6.3dc2**，目的MAC地址为**8c60.4f07.ac65**。您还可以看到这是IPv4帧，源自**10.0.5.101**，发往**10.0.3.101**。

**提示：**此输出中还包含其他几个有用字段，如服务类型(TOS)值、IP标志、IP长度和L2帧长度。

要验证帧在哪个端口上收到，请输入**SRC\_INDEX**命令(源本地目标逻辑(LTL))。输入此命令可将LTL映射到N7K的端口或端口组：

```
N7700# show system internal pixm info ltl 0xe05  
  
Member info  
-----  
Type LTL  
-----  
PHY_PORT Eth4/1  
FLOOD_W_FPOE 0xc031
```

输出显示0xe05的SRC\_INDEX映射到端口Eth4/1。这确认该帧在端口Eth4/1上收到。

使用RBUS数据，您可以验证该帧是否已路由到VLAN 55。请注意，TTL在DBUS数据中以0xff开始。此外，您可以从DEST\_INDEX(目标LTL)确认出口端口：

```
N7K# show system internal pixm info ltl 0x9ed
Member info
```

```
-----
Type                LTL
-----
PHY_PORT            Eth3/5
FLOOD_W_FPOE       0x8017
FLOOD_W_FPOE       0x8016
```

输出显示0x9ed的DEST\_INDEX映射到端口Eth3/5。这确认该帧是从端口Eth3/5发送的。

## 其他验证

要验证交换机如何分配LTL池，请输入show system internal pixm info ltl-region命令。如果LTL与物理端口不匹配，此命令的输出对于了解其用途非常有用。Drop LTL就是一个很好的例子：

```
N7700# show system internal pixm info ltl 0xcad
0x0cad is Drop DI LTL
```

```
N7700# show system internal pixm info ltl-region
(some output omitted) ===== PIXM VDC 1 LTL
MAP Version: 3 Description: LTL Map for Crossbow
===== LTL_TYPE SIZE START END
=====
LIBLTLMAP_LTL_TYPE_PHY_PORT 3072 0x0 0xbff LIBLTLMAP_LTL_TYPE_SUP_ETH_INBAND 64 0xc00 0xc3f
LIBLTLMAP_LTL_TYPE_UCAST_VPC_VDC_SI 32 0xc40 0xc5f LIBLTLMAP_LTL_TYPE_EXCEPTION_SPAN 32 0xc60
0xc7f LIBLTLMAP_LTL_TYPE_UCAST_GENERIC 48 0xc80 0xcaf -----
----- SUB-TYPE LTL -----
----- LIBLTLMAP_LTL_TYPE_UCAST_GENERIC_NOT_USED 0xcaf
LIBLTLMAP_LTL_TYPE_DROP_DI_WO_HW_BITSET 0xcae LIBLTLMAP_LTL_TYPE_DROP_DI
0xcad
LIBLTLMAP_LTL_TYPE_SUP_DIAG_SI_V5 0xcac
LIBLTLMAP_LTL_TYPE_RESERVED_ERSPAN_LTL 0xcab
-----
LIBLTLMAP_LTL_TYPE_LC_CPU 192 0xcb0 0xd6f
LIBLTLMAP_LTL_TYPE_UCAST_RESERVED 144 0xd70 0xdff
LIBLTLMAP_LTL_TYPE_PC 1536 0xe00 0x13ff
LIBLTLMAP_LTL_TYPE_DYNAMIC_UCAST 5120 0x1400 0x27ff
LIBLTLMAP_LTL_TYPE_MCAST_RESERVED 48 0x2800 0x282f
LIBLTLMAP_LTL_TYPE_DYNAMIC_MCAST 38848 0x2830 0xbfef
LIBLTLMAP_LTL_TYPE_SAC_FLOOD 16 0xbff0 0xbfff
LIBLTLMAP_LTL_TYPE_FLOOD_WITH_FPOE 16384 0xc000 0xffff
```