

# 连结7000 M3模块伊拉姆步骤

## 目录

[简介](#)

[拓扑](#)

[确定入口转发引擎](#)

[配置触发](#)

[开始捕获](#)

[解释结果](#)

[另外的验证](#)

## 简介

本文描述用于的步骤为了执行思科连结的伊拉姆7700个(N7700) M3模块，解释最相关的输出，并且描述如何解释结果。

**提示：**参考一概述的[伊拉姆概述文件](#)在伊拉姆。

## 拓扑



在本例中，在VLAN 2500的一台主机(10.0.5.101)，端口Eth4/1发送互联网控制消息协议(ICMP)请求到在VLAN 55的一台主机(10.0.3.101)，端口Eth3/5。 伊拉姆用于为了获取从10.0.5.101的此单个数据包到10.0.3.101。请记住伊拉姆允许您捕获单个帧。

为了执行N7K的伊拉姆，您必须首先连接到适当的模块(这要求网络Admin权限)：

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

## 确定入口转发引擎

流量预计对入口在端口Eth4/1的交换机。 当您检查在系统时的模块，您看到**模块4**是M3模块。 请记住N7K是充分分布式，并且模块，不是Supervisor，做出dataplane流量的转发决策。

```
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
<b>N77-M324FQ-25L</b>			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，请执行Layer2 (L2)转发引擎的(FE)伊拉姆与内部共同的书目分类编号**F4**。注意L2 FE数据总线(DBUS)在L2和第3层(L3)查找前包含原始报头题头信息，并且结果总线(RBUS)在L3和L2以后包含结果查找。

N7K M3模块能使用多个FEs每个模块，因此您必须确定使用在端口**Eth4/1**的FE的**F4 ASIC**。输入此命令为了验证此：

```
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     0,1
 2       0     0     0     0     0     0,1
 3       0     0     0     0     0     0,1
```

在输出中，您能看到端口**Eth4/1**在**F4 (L2LKP) 0**实例。在N77-M312CQ-26L模块上，有**6 F4 ASIC**用2个端口在每个端口组中。在N77-M324FQ-25L模块上，有**4 F4 ASIC**用6个端口在每个端口组中。N77-M348XP-23L模块有**2个F4 ASIC**用12个端口在每个端口组中。

**注意：**正如F系列模块，M3模块伊拉姆语法使用基于0的值。这不是M1和M2模块的盒，使用基于1的值。

```
module-4# elam asic f4 instance 0
module-4(f4-elam)# layer2
module-4(f4-12-elam)#
```

## 配置触发

**F4 ASIC**支持IPv4,IPv6的伊拉姆触发和其他。伊拉姆触发必须与帧类型对齐。如果帧是IPv4帧，则触发必须也是IPv4。IPv4帧没有用其他触发捕获。同样逻辑适用于IPv6。

使用连结操作系统(NX-OS)，您能使用问号字符为了分离伊拉姆触发：

```
module-4(f4-12-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
```

对于此示例，帧根据源和目的IPv4地址捕获，那么仅那些值指定。

**F4要求DBUS和RBUS的独立的触发。**

这是DBUS触发：

```
module-4(f4-12-elam)# trigger dbus ipv4 ingress if source-ipv4-address  
10.0.5.101 destination-ipv4-address 10.0.3.101
```

这是RBUS触发：

```
module-4(f4-12-elam)# trigger rbus ingress result if tr 1
```

## 开始捕获

既然入口FE选择，并且您配置触发，您能开始捕获：

```
module-4(f4-12-elam)# start
```

为了检查ELAM的状态，请输入**status**命令：

```
module-4(f4-12-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
```

一旦匹配触发的帧由FE接收，伊拉姆状态显示如被触发：

```
module-4(f4-12-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

为了显示伊拉姆结果，请输入**显示d总线并且显示rbus**命令。如果有该大容积的流量匹配同样触发，DBUS，并且RBUS在不同的帧也许触发。所以，检查在DBUS和RBUS数据的内部序号为了保证是重要的他们配比：

```
module-4(f4-12-elam)# show dbus | i seq  
port-id : 0x0 sequence-number : 0x868  
module-4(f4-12-elam)# show rbus | i seq  
de-bri-rslt-valid : 0x1 sequence-number : 0x868
```

这是摘自与此示例是最相关的伊拉姆数据的部分(若干输出省略)：

```
module-4(f4-12-elam)# show dbus
-----
          LBD IPV4
-----
ttl          : 0xff      13-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(f4-12-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
```

有DBUS数据，您能验证帧在与8c60.4fb6.3dc2源MAC地址和8c60.4f07.ac65目标MAC地址的VLAN 2500接收。您能也看到这是从10.0.5.101来源的IPv4帧和被注定对10.0.3.101。

**提示**：有在此输出中没有包括，例如服务类型(ToS)值、IP标志、IP长度和L2帧长度的几个其他有用的字段。

为了验证在哪个端口帧接收，请输入SRC\_INDEX命令(来源Local Target Logic (LTL))。输入此命令为了映射LTL对端口或端口组N7K的：

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

输出显示那0xe05地图SRC\_INDEX到端口Eth4/1。这确认帧在端口Eth4/1接收。

有RBUS数据，您能验证帧路由对VLAN 55。注意TTL开始作为在DBUS数据的0xff。另外，您能确认从DEST\_INDEX(目的地LTL)的输出端口：

```
N7K# show system internal pixm info ltl 0x9ed
Member info
-----
Type      LTL
-----
PHY_PORT  Eth3/5
FLOOD_W_FPOE 0x8017
FLOOD_W_FPOE 0x8016
```

输出显示那0x9ed地图DEST\_INDEX到端口Eth3/5。这确认帧从端口Eth3/5发送。

## 另外的验证

按顺序请验证交换机如何分配LTL池，输入show system内部pixm信息LTL赤道区命令。如果没有匹配到物理端口，从此命令的输出是有用的为了了解LTL的目的。此的好实例是丢弃LTL：

```
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      0ocab
-----
LIBLTLMAP_LTL_TYPE_LC_CPU        192  0xcb0  0xd6f
LIBLTLMAP_LTL_TYPE_UCAST_RESERVED 144   0xd70  0xffff
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  0xffff
LIBLTLMAP_LTL_TYPE_FLOOD_WITH_FPOE 16384 0xc000 0xffff
```