Nexus 7000 M−Series Module ELAM Procedure

Document ID: 116645

Contributed by Andrew Gossett and Yogesh Ramdoss, Cisco TAC
Engineers.
Oct 21, 2013

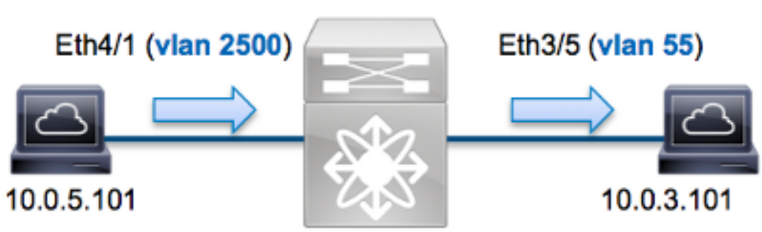
Contents

Introduction

This document describes the steps used in order to perform an ELAM on Cisco Nexus 7000 (N7K) M−Series modules, explains the most relevant outputs, and describes how to interpret the results.

Tip: Refer to the ELAM Overview document for an overview on ELAM.

Topology



In this example, a host on VLAN 2500 (***10.0.5.101***), port ***Eth4/1*** sends an Internet Control Message Protocol (ICMP) request to a host on VLAN 55 (***10.0.3.101***), port ***Eth3/5***. ELAM is used in order to capture this single packet from ***10.0.5.101*** to ***10.0.3.101***. It is important to remember that ELAM allows you to capture a single frame.

In order to perform an ELAM on the N7K, you must first connect to the appropriate module (this requires the network−admin privilege):

```
N7K# attach module 4
Attaching to module 4 ...
To exit type 'exit', to abort type '$.'
module−4#
```

Determine the Ingress Forwarding Engine

Traffic is expected to ingress the switch on port ***Eth4/1***. When you check the modules in the system, you see that ***Module 4*** is an M−Series module. It is important to remember that the N7K is fully−distributed, and that the modules, not the supervisor, make the forwarding decisions for dataplane traffic.

```
N7K# show module
Mod  Ports  Module−Type                         Model              Status
---  -----  ----------------------------------- ------------------ ----------
3    32     10 Gbps Ethernet Module             N7K−M132XP−12      ok
4    48     10/100/1000 Mbps Ethernet Module    N7K−M148GT−11      ok
5    0      Supervisor module−1X                N7K−SUP1           active *
6    0      Supervisor module−1X                N7K−SUP1           ha−standby
```

For M−Series modules, perform the ELAM on the Layer 2 (L2) Forwarding Engine (FE) with internal codename ***Eureka***. Note that the L2 FE Data Bus (DBUS) contains the original header information before the L2 and Layer 3 (L3) lookups, and the Result Bus (RBUS) contains the results after both L3 and L2 lookups. The L3 lookup is performed by the L3/Layer 4 (L4) FE with internal codename ***Lamira***, which is the same process used on the Cisco Catalyst 6500 Series switch platform that runs Supervisor Engine 2T.

N7K M−Series modules can use multiple FEs for each module, so you must determine the ***Eureka*** ASIC that is used for the FE on port ***Eth4/1***. Enter this command in order to verify this:

```
module−4# show hardware internal dev−port−map
(some output omitted)
--------------------------------------------------------------
CARD_TYPE:         48 port 1G
>Front Panel ports:48
--------------------------------------------------------------
 Device name             Dev role              Abbr num_inst:
--------------------------------------------------------------
> Eureka                 DEV_LAYER_2_LOOKUP     L2LKP  1
+----------------------------------------------------------------+
+-----------+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++--------+
+----------------------------------------------------------------+
FP port|PHYS |SECUR |MAC_0 |RWR_0 |L2LKP |L3LKP |QUEUE |SWICHF
   1     0      0      0      0      0      0      0      0
   2     0      0      0      0     0      0      0      0
```

In the output, you can see that port ***Eth4/1*** is on ***Eureka (L2LKP)*** instance ***0***.

Note: For M−Series modules, the ELAM syntax uses 1−based values, so instance ***0*** becomes instance ***1*** when you configure the ELAM. This is not the case for F−Series modules.

```
module−4# elam asic eureka instance 1
module−4(eureka−elam)#
```

Configure the Trigger

The ***Eureka*** ASIC supports ELAM triggers for IPv4, IPv6, and others. The ELAM trigger must align with the frame type. If the frame is an IPv4 frame, then the trigger must also be IPv4. An IPv4 frame is not captured with an *other* trigger. The same logic applies to IPv6.

With Nexus Operating Systems (NX−OS), you can use the question mark character in order to separate the ELAM trigger:

```
module−4(eureka−elam)# trigger dbus dbi ingress ipv4 if ?
  (some output omitted)
  destination−flood          Destination Flood
  destination−index          Destination Index
  destination−ipv4−address  Destination IP Address
```

```
  destination-mac-address   Destination MAC Address
  ip-tos                    IP TOS
  ip-total-len              IP Total Length
  ip-ttl                    IP TTL
  source-mac-address        Source MAC Address
  vlan-id                   Vlan ID Number
```

For this example, the frame is captured according to the source and destination IPv4 addresses, so only those values are specified.

Eureka requires that triggers are set for the DBUS and the RBUS. There are two different Packet Buffers (PB) in which the RBUS data can reside. Determination of the correct PB instance is dependent upon the exact module type and ingress port. Typically, it is recommended that you configure PB1, and if the RBUS does not trigger, then repeat the configuration with PB2.

Here is the DBUS trigger:

```
module-4(eureka-elam)# trigger dbus dbi ingress ipv4 if source-ipv4-address
  10.0.5.101 destination-ipv4-address 10.0.3.101 rbi-corelate
```

Here is the RBUS trigger:

```
module-4(eureka-elam)# trigger rbus rbi pb1 ip if cap2 1
```

Note: The ***rbi−correlate*** keyword at the end of the DBUS trigger is required in order for the RBUS to correctly trigger on the ***cap2*** bit.

Start the Capture

Now that the ingress FE is selected and you configured the trigger, you can start the capture:

```
module-4(eureka-elam)# start
```

In order to check the status of the ELAM, enter the ***status*** command:

```
module-4(eureka-elam)# status
Instance: 1
EU-DBUS: Armed
trigger dbus dbi ingress ipv4 if source-ipv4-address 10.0.5.101
  destination-ipv4-address 10.0.3.101 rbi-corelate
EU-RBUS: Armed
trigger rbus rbi pb1 ip if cap2 1
LM-DBUS: Dis-Armed
No configuration
LM-RBUS: Dis-Armed
No configuration
```

Once the frame that matches the trigger is received by the FE, the ELAM status shows as ***Triggered***:

```
module-4(eureka-elam)# status
Instance: 1
EU-DBUS: Triggered
trigger dbus dbi ingress ipv4 if source-ipv4-address 10.0.5.101
  destination-ipv4-address 10.0.3.101 rbi-corelate
EU-RBUS: Triggered
trigger rbus rbi pb1 ip if cap2 1
LM-DBUS: Dis-Armed
No configuration
LM-RBUS: Dis-Armed
```

```
No configuration
```

Interpret the Results

In order to display the ELAM results, enter the ***show dbus*** and ***show rbus*** commands. If there is a high volume of traffic that matches the same triggers, the DBUS and RBUS might trigger on different frames. Therefore, it is important to check the internal sequence numbers on the DBUS and RBUS data in order to ensure that they match:

```
module-4(eureka-elam)# show dbus | i seq
seq = 0x05
module-4(eureka-elam)# show rbus | i seq
seq = 0x05
```

Here is the excerpt from the ELAM data that is most relevant to this example (some output is omitted):

```
module-4(eureka-elam)# show dbus
seq = 0x05
vlan = 2500
source_index = 0x00a21
l3_protocol = 0x0   (0:IPv4, 6:IPv6)
l3_protocol_type = 0x01, (1:ICMP, 2:IGMP, 4:IP, 6:TCP, 17:UDP)
dmac = 00.00.0c.07.ac.65
smac = d0.d0.fd.b7.3d.c2
ip_ttl = 0xff
ip_source = 010.000.005.101
ip_destination = 010.000.003.101

module-4(eureka-elam)# show rbus
seq = 0x05
flood = 0x0
dest_index = 0x009ed
vlan = 55
ttl = 0xfe
data(rit/dmac/recir) = 00.05.73.a9.55.41
data(rit/smac/recir) = 84.78.ac.0e.47.41
```

With the ***DBUS*** data, you can verify that the frame is received on VLAN 2500 with a source MAC address of ***d0d0.fdb7.3dc2*** and a destination MAC address of ***0000.0c07.ac65***. You can also see that this is an IPv4 frame that is sourced from ***10.0.5.101***, and is destined to ***10.0.3.101***.

Tip: There are several other useful fields that are not included in this output, such as Type of Service (TOS) value, IP flags, IP length, and L2 frame length.

In order to verify on which port the frame is received, enter the ***SRC_INDEX*** command (the source Local Target Logic (LTL)). Enter this command in order to map an LTL to a port or group of ports for the N7K:

```
N7K# show system internal pixm info ltl 0xa21
Member info
------------------
Type             LTL
---------------------------------
PHY_PORT        Eth4/1
FLOOD_W_FPOE    0x8014
```

The output shows that the ***SRC_INDEX*** of ***0xa21*** maps to port ***Eth4/1***. This confirms that the frame is received on port ***Eth4/1***.

With the ***RBUS*** data, you can verify that the frame is routed to VLAN 55, and that the TTL is decremented from ***0xff*** in the ***DBUS*** data to ***0xfe*** in the ***RBUS*** data. You can see that the source and destination MAC addresses are rewritten to ***8478.ac0e.4741*** and ***0005.73a9.5541***, respectively. Additionally, you can confirm the egress port from the ***DEST_INDEX*** (destination LTL):

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

The output shows that the ***DEST_INDEX*** of ***0x9ed*** maps to port ***Eth3/5***. This confirms that the frame is sent from port ***Eth3/5***.

Additional Verification

In order to verify how the switch allocates the LTL pool, enter the ***show system internal pixm info ltl−region*** command. The output from this command is useful in order to understand the purpose of an LTL if it is not matched to a physical port. A good example of this is a ***Drop*** LTL:

```
N7K# show system internal pixm info ltl 0x11a0
0x11a0 is not configured

N7K# show system internal pixm info ltl-region
 LTL POOL TYPE                          SIZE        RANGE
 =====================================================================
 DCE/FC Pool                            1024       0x0000 to 0x03ff
 SUP Inband LTL                           32       0x0400 to 0x041f
 MD Flood LTL                              1       0x0420
 Central R/W                               1       0x0421
 UCAST Pool                             1536       0x0422 to 0x0a21
 PC Pool                                1720       0x0a22 to 0x10d9
 LC CPU Pool                              32       0x1152 to 0x1171
 EARL Pool                                72       0x10da to 0x1121
 SPAN Pool                                48       0x1122 to 0x1151
 UCAST VDC Use Pool                       16       0x1172 to 0x1181
 UCAST Generic Pool                       30       0x1182 to 0x119f
 LISP Pool                                 4       0x1198 to 0x119b
 Invalid SI                                1       0x119c to 0x119c
 ESPAN SI                                  1       0x119d to 0x119d
 Recirc SI                                 1       0x119e to 0x119e
 Drop DI                                   2       0x119f to 0x11a0
 UCAST (L3_SVI_SI) Region                 31       0x11a1 to 0x11bf
 UCAST (Fex/GPC/SVI-ES)       3648       0x11c0 to 0x1fff
 UCAST Reserved for Future Use Region   2048       0x2000 to 0x27ff
=====================> UCAST MCAST BOUNDARY <=====================
 VDC OMF Pool                             32       0x2800 to 0x281f
```