Nexus 7000 F2 Module ELAM Procedure

TAC **Document ID: 116647**

Contributed by Andrew Gossett, Cisco TAC Engineer.
Oct 22, 2013

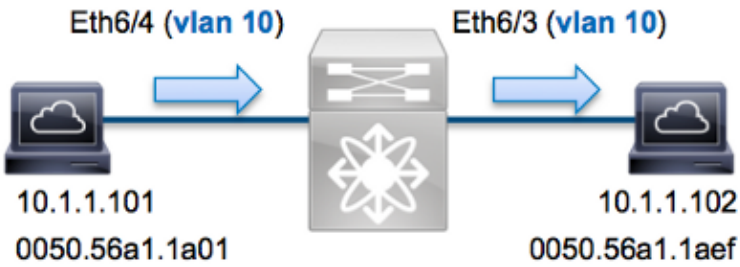
Contents

Introduction

This document describes the steps used in order to perform an ELAM on a Cisco Nexus 7000 (N7K) F2 module, 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 10 (***10.1.1.101*** with MAC address ***0050.56a1.1a01***), port ***Eth6/4*** sends an Internet Control Message Protocol (ICMP) request to a host that is also on VLAN 10 (***10.1.1.102*** with MAC address ***0050.56a1.1aef***), port ***Eth6/3***. ELAM is used in order to capture this single frame from ***10.1.1.101*** to ***10.1.1.102***. It is important to remember that ELAM allows you to capture only 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 6
Attaching to module 6 ...
To exit type 'exit', to abort type '$.'
module-6#
```

Determine the Ingress Forwarding Engine

Traffic is expected to ingress the switch on port ***Eth6/4***. When you check the modules in the system, you see that ***Module 6*** is an F2 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 6
Mod  Ports  Module-Type                 Model              Status
---  -----  -------------------------- ------------------ -------
6    48     1/10 Gbps Ethernet Module  N7K-F248XP-25E     ok
```

For F2 modules, perform the ELAM on the Layer 2 (L2) Forwarding Engine (FE) with internal codename ***Clipper***. 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 N7K F2 has 12 FEs per module, so you must determine the ***Clipper*** ASIC that is used for the FE on port ***Eth6/4***. Enter this command in order to verify:

```
module-6# show hardware internal dev-port-map
--------------------------------------------------------------
CARD_TYPE:         48 port 10G
>Front Panel ports:48
--------------------------------------------------------------
 Device name             Dev role              Abbr num_inst:
--------------------------------------------------------------
> Clipper FWD            DEV_LAYER_2_LOOKUP     L2LKP  12
+-----------------------------------------------------------------------+
+----------------+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++------------+
+-----------------------------------------------------------------------+
FP port |  PHYS | MAC_0 | L2LKP | L3LKP | QUEUE |SWICHF
...
   3       0       0       0       0       0       0
   4       0       0       0       0       0       0
```

In the output, you can see that port ***Eth6/4*** is on ***Clipper (L2LKP)*** instance ***0***.

```
module-6# elam asic clipper instance 0
module-6(clipper-elam)# layer2
module-6(clipper-l2-elam)#
```

Configure the Trigger

The ***Clipper*** ASIC supports ELAM triggers for multiple frame types. 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.

The ***Clipper*** ASIC supports these frame types:

```
module-6(clipper-l2-elam)# trigger dbus ?
  arp    ARP Frame Format
  fc     Fc hdr Frame Format
  ipv4   IPV4 Frame Format
  ipv6   IPV6 Frame Format
  other  L2 hdr Frame Format
  pup    PUP Frame Format
  rarp   Rarp hdr Frame Format
  valid  On valid packet
```

With Nexus Operating Systems (NX−OS), you can use the question mark character in order to separate the ELAM trigger. There are several options available for ELAM on the F2 module:

```
module−6(clipper−l2−elam)# trigger dbus ipv4 ingress if ?
  <CR>
  destination−ipv4−address         destination ipv4 address
  destination−mac−address          Inner destination mac address
  source−index                     Source index
  source−ipv4−address              source ipv4 address
  source−mac−address               Inner source mac address
  vlan                             Vlan
  etc?
```

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

Clipper requires that triggers are set for the DBUS and the RBUS. This differs from M−Series modules, because there is no requirement that you must specify a Packet Buffer (PB) instance. This simplifies the RBUS trigger.

Here is the DBUS trigger:

```
module−6(clipper−l2−elam)# trigger dbus ipv4 ingress if source−ipv4−address
  10.1.1.101 destination−ipv4−address 10.1.1.102
```

Here is the RBUS trigger:

```
module−6(clipper−l2−elam)# trigger rbus ingress if trig
```

Start the Capture

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

```
module−6(clipper−l2−elam)# start
```

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

```
module−6(clipper−l2−elam)# status
ELAM instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if
  source−ipv4−address 10.1.1.101 destination−ipv4−address 10.1.1.102
L2 DBUS Armed
ELAM instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
L2 RBUS Armed
```

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

```
module−6(clipper−l2−elam)# status
ELAM instance 0: L2 DBUS Configuration: trigger dbus ipv4 ingress if
  source−ipv4−address 10.1.1.101 destination−ipv4−address 10.1.1.102
L2 DBUS Triggered
ELAM instance 0: L2 RBUS Configuration: trigger rbus ingress if trig
L2 RBUS Triggered
```

Interpret the Results

In order to display the ELAM results, enter the ***show dbus*** and ***show rbus*** commands. Here is the excerpt from the ELAM data that is most relevant to this example (some output is omitted):

```
module-6(clipper-l2-elam)# show dbus
-------------------------------------------------------------------
                      L2 DBUS CONTENT - IPV4 PACKET
-------------------------------------------------------------------
...
vlan                : 0xa             destination-index   : 0x0
source-index        : 0x3             bundle-port         : 0x0
sequence-number     : 0x3f            vl                  : 0x0
...
source-ipv4-address: 10.1.1.101
destination-ipv4-address: 10.1.1.102
destination-mac-address: 0050.56a1.1aef
source-mac-address: 0050.56a1.1a01


module-6(clipper-l2-elam)#  show rbus
-------------------------------------------------------------------
                       L2 RBUS INGRESS CONTENT
-------------------------------------------------------------------
l2-rbus-trigger     : 0x1             sequence-number     : 0x3f
di-ltl-index        : 0x2             l3-multicast-di     : 0x0
source-index        : 0x3             vlan-id             : 0xa
```

With the ***DBUS*** data, you can verify that the frame is received on VLAN 10 (***vlan: 0xa***) with a source MAC address of ***0050.56a1.1a01*** and a destination MAC address of ***0050.56a1.1aef***. You can also see that this is an IPv4 frame that is sourced from ***10.1.1.101***, and is destined to ***10.1.1.102***.

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 0x3
Type            LTL
--------------------------------
PHY_PORT        Eth6/4
```

The output shows that a ***source-index*** of ***0x3*** maps to port ***Eth6/4***. This confirms that the frame is received on port ***Eth6/4***.

With the ***RBUS data***, you can verify that the frame is switched on VLAN 10 (***vlan-id: 0xa***). Additionally, you can confirm the egress port from the ***di-ltl-index*** (destination LTL):

```
N7K# show system internal pixm info ltl 0x2
Type            LTL
--------------------------------
PHY_PORT        Eth6/3
```

The output shows that a ***di-ltl-index*** of ***0x2*** maps to port ***Eth6/3***. This confirms that the frame is switched from port ***Eth6/3***.

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
```