Troubleshoot ACI Intra-Fabric Forwarding Tools

Contents

Introduction

This document describes tools natively included in ACI that can be used to debug forwarding problems.

Background Information

The material from this document was extracted from the Troubleshooting Cisco Application Centric Infrastructure, Second Edition book, specifically the **Intra-Fabric fowarding - Tools** chapter.

Additionally, deeper explanations of ELAM and Ftriage can be found in the CiscoLive On-Demand library in session BRKDCN-3900b.

What Can These Tools Help With?

In order to troubleshoot a forwarding problem from an ACI perspective, understand:

1. Which switch is receiving a flow?
2. What forwarding decision is that switch making?
3. Is the switch dropping it?

ACI includes several tools which allow the user to gain in-depth insights into what is happening to a specific flow. The next several sections demonstrate these tools in detail so only a high-level introduction is provided here.

SPAN and ERSPAN

SPAN and ERSPAN are both tools that allow all or some traffic received at a specific location to be replicated to another location. The end device that the replicated traffic is sent to is expected to be running some type of packet sniffer/analyzer application. Traditional SPAN involves replicating traffic that is being received on one port and passing out through another port. ACI supports doing this in addition to ERSPAN.

ERSPAN follows the same concept except replicating the traffic out a local port; the replicated traffic is encapsulated in GRE and sent to a remote destination. In ACI, this ERSPAN destination must only be learned as a Layer 3 endpoint and it can be any EPG in any VRF.

It is a good idea to always have SPAN destinations connected to the fabric to minimize preparation time during troubleshooting and allow for rapid ERSPAN session config and capture.

ELAM

Overview

Embedded Logic Analyzer Module (ELAM) is a tool that allows a user to set conditions in hardware and capture the first packet or frame that matches the set conditions. A successful capture causes the ELAM status to show as "triggered." Once triggered, the ELAM is disabled and a dump can be collected to analyze the vast number of forwarding decisions that the switch ASIC is making with that packet/frame. ELAM is implemented at the ASIC level and does not impact CPU or other resources on the switch.

The forwarding examples in this book uses ELAM as a means of verifying what is happening with the flow. Examples shows both the leaf CLI version and the ELAM Assistant App.

This guide does not cover usage of ELAM on first generation leaf switches (switches without EX, FX, or FX2 suffix).

Before using the tool, it is important to understand the structure of the command syntax.

Example on leaf CLI:

```
vsh_lc                                       [This command enters the line card shell where ELAMs are r
debug platform internal <asic> elam asic 0   [refer to the ASICs table]
```

Set Conditions to "trigger"

```
trigger reset                                           [ensures no existing triggers are running]
trigger init in-select <number> out-select <number>     [determines what information about a packet is
set outer/inner                                         [sets conditions]
start                                                   [starts the trigger]
status                                                  [checks if a packet is captured]
```

Generate the dump containing the packet analysis

```
ereport             [display detailed forwarding decision for the packet]
```

Continue to enter the **status** command to view the state of the trigger. Once a packet matching the defined conditions is detected on the ASIC, the output of **status** shows "triggered." Once the ELAM has been triggered, the details of the switch forwarding decisions can be shown with 'ereport'. Prior to ACI version 4.2, 'report' must be used.

ASICs

Within the ELAM syntax, note that the ASIC must be specified. Since the ASIC is dependent on the switch model, refer to this table to determine which ASIC to specify:

ASICs Table

Switch/Line card Family	Asic for Elam
-EX switches/LCs	TAH
-FX(P) switches/LCs	ROC
-FX2 switches/LCs	ROC
C switches (9364C,9332C)	ROC
-GX switches	APP
-GX2 switches	CHO
-FX3 switches	ROC

ELAM Trigger in-select

The other component of the ELAM that must be understood when running from the CLI is the **in-select**. The **in-select** defines which headers the packet/frame is expected to have, and which to match on.

For example, a packet coming from a downlink port that is not VXLAN encapsulated would only have outer Layer 2, Layer 3, and Layer 4 headers.

A packet coming from a front-panel (downlink) port that is VXLAN encapsulated (such as Cisco ACI Virtual Edge in VXLAN mode) or coming from an upstream spine would have VXLAN encapsulation. This means it would have potentially both outer and inner Layer 2, Layer 3, and Layer 4 headers.

All trigger options are:

<#root>

```
leaf1#

vsh_lc


module-1#

debug platform internal tah elam asic 0


module-1(DBG-elam)#

trigger reset


module-1(DBG-elam)#

trigger init in-select ?


 10  Outerl4-innerl4-ieth
 13  Outer(l2|l3|l4)-inner(l2|l3|l4)-noieth
 14  Outer(l2(vntag)|l3|l4)-inner(l2|l3|l4)-ieth
 15  Outer(l2|l3|l4)-inner(l2|l3|l4)-ieth
 6   Outerl2-outerl3-outerl4
 7   Innerl2-innerl3-innerl4
 8   Outerl2-innerl2-ieth
 9   Outerl3-innerl3
```

If **in-select 6** is selected the only option is to set conditions and display headers from the outer Layer 2, 3, or 4 headers. If **in-select 14** is selected the only option is to set conditions for and see the details of the outer and inner Layer 2, 3, and 4 headers.

Best practices note:

To capture a packet coming with VLAN encapsulation on a downlink port, use 'in-select 6'

To capture a packet with VXLAN encapsulation (either from a spine or from a vleaf with VXLAN encapsulation) use **in-select 14**.

ELAM Trigger out-select

The **out-select** allows some ability to control which lookup results are displayed in the ELAM report. For most practical purposes **out-select 0** can be used as it contains most information including the **drop vector**, which tells if the result of the lookup is to drop the packet/frame.

Note that when **report** instead of **ereport** or **report detail** is used to get ELAM results, **drop vector** only shows up in **out-select 1**. However, one can always perform **ereport** or **report detail** with **out-select 0**.

ELAM set Conditions

ELAM supports a large amount of Layer 2, 3, and 4 conditions to look for in a packet. Specifying **inner** vs. **outer** determines if the condition can be checked in the inner header (VXLAN encapsulated packet) or outer header.

ARP example:

```
set outer arp source-ip-address 10.0.0.1 target-ip-address 10.0.0.2
```

MAC address example:

```
set outer l2 src_mac aaaa.bbbb.cccc dst_mac cccc.bbbb.aaaa
```

IP address in inner header example:

```
set inner ipv4 src_ip 10.0.0.1 dst_ip 10.0.0.2
```

Viewing the ELAM Report

Verify that the ELAM has triggered with **status**:

<#root>

```
module-1(DBG-elam-insel6)#
```

```
status
```

```
ELAM STATUS
===========
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Triggered
```

ereport can be used to display the result of the ELAM in an easy to understand format. Note that the ELAM report is saved in the /var/log/dme/log/ folder on the switch. There are two files for the ELAM under the folder.

- elam_<timestamp>.txt
- pretty_elam_<timestamp>.txt

Full ELAM Example

This example would capture a non-VXLAN encapsulated traffic (matching on outer header) coming from a downlink port on an -EX switch:

<#root>

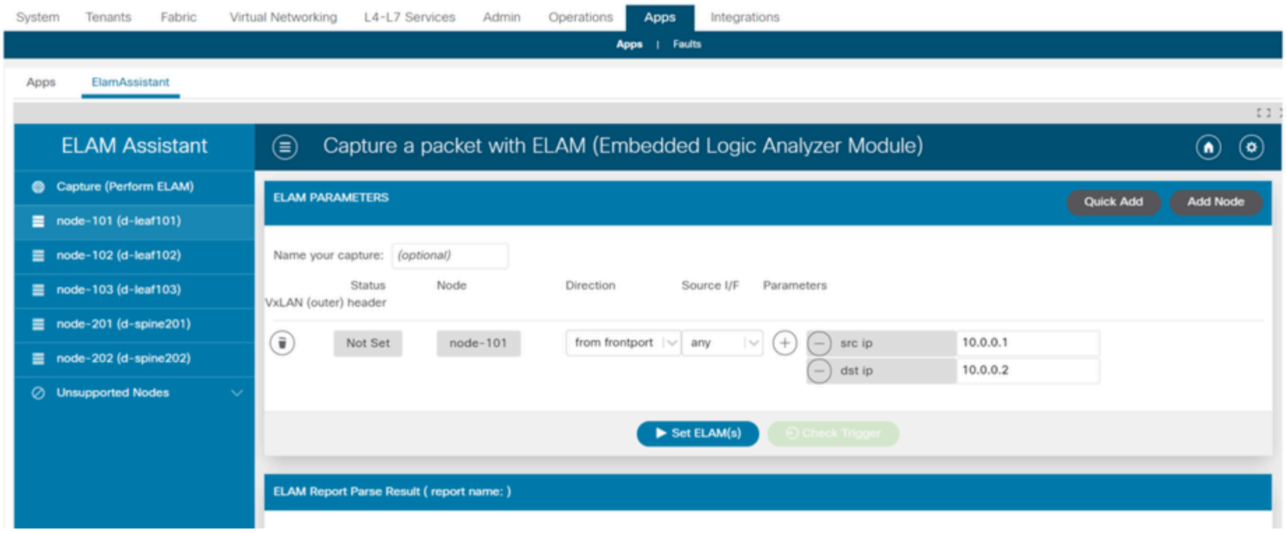
```
module-1#

debug platform internal tah elam asic 0


module-1(DBG-elam)#

trigger reset


module-1(DBG-elam)#

trigger init in-select 6 out-select 0


module-1(DBG-elam-insel6)#

set outer ipv4 src_ip 10.0.0.1 dst_ip 10.0.0.2


module-1(DBG-elam-insel6)#

start


module-1(DBG-elam-insel6)#

status


module-1(DBG-elam-insel6)#

ereport
```

ELAM Assistant Application

The troubleshooting examples in this book also show the usage of the ELAM Assistant app which can be downloaded through the Cisco DC App Center (https://dcappcenter.cisco.com). This tool automates the deployment and interpretation of ELAMs through the GUI on the APIC.

This example shows the deployment of an ELAM matching a specific source and destination IP on node-101 downlink port

ElamAssistant



Capture a packet with ELAM (Embedded Logic Analyzer Module)

ELAM PARAMETERS

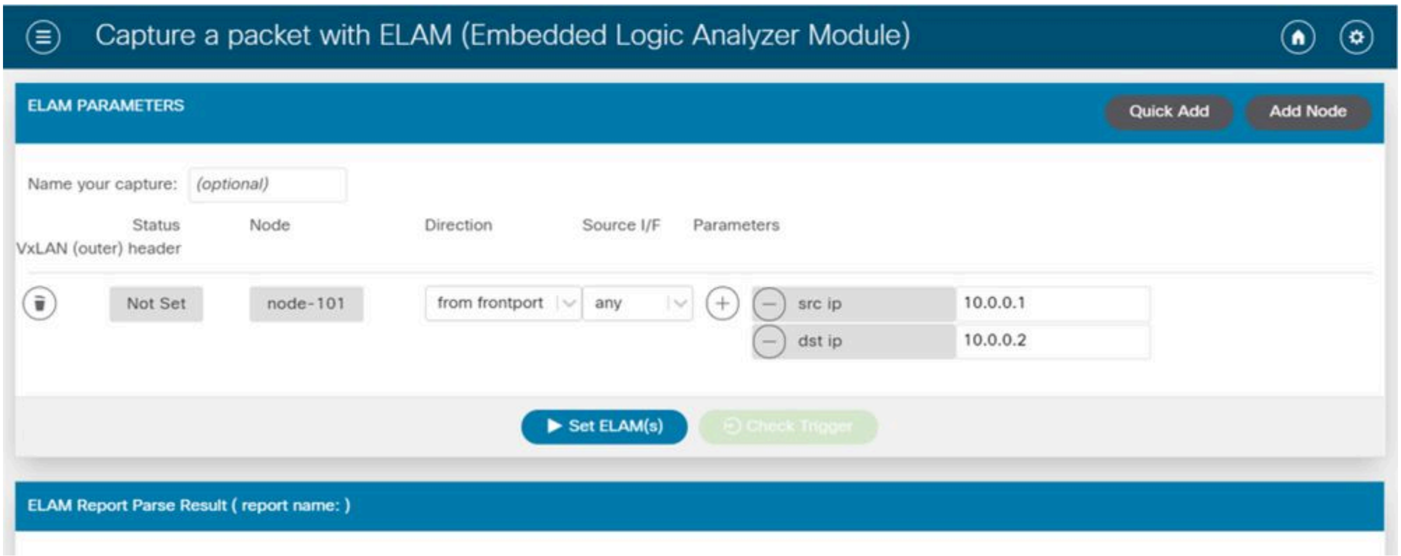
Name your capture: (optional)

Status	Node	Direction	Source I/F	Parameters	
Not Set	node-101	from frontport	any	src ip	10.0.0.1
				dst ip	10.0.0.2

VxLAN (outer) header

ELAM Report Parse Result (report name:)

ElamAssistant - Detail



Capture a packet with ELAM (Embedded Logic Analyzer Module)

ELAM PARAMETERS

Name your capture: (optional)

Status	Node	Direction	Source I/F	Parameters	
Not Set	node-101	from frontport	any	src ip	10.0.0.1
				dst ip	10.0.0.2

VxLAN (outer) header

ELAM Report Parse Result (report name:)

The ELAM Assistant also allows for easy usage of more complex matching parameters such as the source interface or VXLAN values.

fTriage

fTriage is an APIC CLI-based tool that is intended to provide end-to-end automation of ELAM configuration and interpretation. The premise of the tool is that a user can define a specific flow as well as the leaf where the flow is expected to start. The tool then executes ELAMs on each node, one by one, to examine the forwarding flow. It is particularly useful in large topologies where it is unclear which path a packet takes.

fTriage generates a large log file containing the output of each command executed. The name of this file is visible on the first few lines of the **fTriage** output.

fTriage completion can take up to 15 minutes.

Examples

Map out the flow for routed communication between 10.0.1.1 and 10.0.2.1 starting on leaf 104:

```
ftriage route -ii LEAF:104 -dip 10.0.2.1 -sip 10.0.1.1
```

Map out a Layer 2 flow starting on leaf 104:

```
ftriage bridge -ii LEAF:104 -dmac 02:02:02:02:02:02
```

Full fTriage help can be seen by running **ftriage --help** on the APIC.

tcpdump

tcpdump can be leveraged on ACI switches to capture traffic to and from the control-plane. Note that **only control plane traffic** sent to the switch CPU can be observed in a **tcpdump** capture. Some examples are: routing protocols, LLDP/CDP, LACP, ARP, and so on. To capture dataplane (and control plane) traffic please make use of SPAN and/or ELAM.

To capture on the CPU, specify the kpm_inb interface**.** Most traditional **tcpdump** options and filters are available.

Example to capture ICMP destined to an SVI on the leaf switch:

<#root>

```
leaf205#
```

`tcpdump -ni kpm_inb icmp`

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on kpm_inb, link-type EN10MB (Ethernet), capture size 65535 bytes
20:24:12.921981 IP 10.0.2.100 > 10.0.2.1: ICMP echo request, id 62762, seq 4096, length 64
20:24:12.922059 IP 10.0.2.1 > 10.0.2.100: ICMP echo reply, id 62762, seq 4096, length 64
20:24:13.922064 IP 10.0.2.100 > 10.0.2.1: ICMP echo request, id 62762, seq 4352, length 64
20:24:13.922157 IP 10.0.2.1 > 10.0.2.100: ICMP echo reply, id 62762, seq 4352, length 64
20:24:14.922231 IP 10.0.2.100 > 10.0.2.1: ICMP echo request, id 62762, seq 4608, length 64
20:24:14.922303 IP 10.0.2.1 > 10.0.2.100: ICMP echo reply, id 62762, seq 4608, length 64
```

In addition, the **-w** option allows the tcpdump to write the packet capture to a PCAP file so that it can be opened in tools such as Wireshark.

To use **tcpdump** on the eth0 interface, which is the out-of-band interface on the switch. This is useful to troubleshoot connectivity of any traffic going through the out-of-band physical port of the switch. This would mainly be control plane-based traffic such as SSH, SNMP, and so on.

On-demand Atomic Counters

On-demand atomic counters are intended to count packets within a specific flow as they leave on a leaf uplink and are received on another leaf fabric port. They allow some granularity into whether packets were missed or received in excess.