

ACI交换矩阵发现故障排除 — 初始交换矩阵设置

目录

[简介](#)

[背景信息](#)

[交换矩阵发现工作流程](#)

[Check01 — 系统状态](#)

[Check02 — DHCP状态](#)

[Check03 — AV详细信息](#)

[Check04 — 与APIC的IP可达性](#)

[Check05 — 基础VLAN](#)

[Check06 — LLDP邻接关系](#)

[Check07 — 交换机版本](#)

[Check08 — FPGA/EPLD/BIOS不同步](#)

[Check09 — SSL检查](#)

[Check10 — 下载策略](#)

[Check11 — 时间](#)

[Check12 — 模块、PSU、风扇检查](#)

[中断方案示例](#)

[场景1 — 第一个枝叶未出现在交换矩阵成员资格中](#)

[场景2 — 其他APIC不加入集群](#)

[场景3 — 交换矩阵成员中未显示主干](#)

[场景4 — 初次发现交换矩阵后，集群在完全适应和降级之间摇摆](#)

简介

本文档介绍了解初始交换矩阵发现过程并对其进行故障排除的步骤，包括问题场景示例。

背景信息

本文档中的内容摘自 [思科以应用为中心的基础设施故障排除（第二版）](#) 书，特别是 [交换矩阵发现 — 初始交换矩阵设置](#) 第章。

交换矩阵发现工作流程

ACI交换矩阵发现过程遵循特定的事件序列。基本步骤如下：

1. 连接到第一个APIC的**KVM控制台**，并通过输入值（如交换矩阵名称、APIC集群大小和隧道终端[TEP]地址池）完成**设置脚本**。
2. 完成后，APIC1将开始通过其交换矩阵端口发送**LLDP**。LLDP数据包包含具有信息的特殊TLV，例如**infra VLAN**及其作为APIC（也称为控制器）的角色。
3. 在从APIC1接收这些LLDP数据包时，枝叶会对检测到APIC的所有端口上的基础设施VLAN进行编程。

4. 枝叶开始在现在已知的基础VLAN上发送DHCP发现。
5. 用户通过HTTPS登录APIC1的OOB IP，并在Fabric Membership子菜单中注册第一个枝叶节点。
6. 一旦为枝叶提供节点ID, APIC1将使用配置的TEP地址池中的IP地址进行响应，DHCP过程完成。
7. 注册的枝叶将DHCP发现从通过LLDP发现的其他直连主干中继到APIC1。
8. 用户将看到动态发现的脊柱显示在Fabric Membership子菜单中，并且可以注册这些脊柱。
9. 一旦脊柱完成注册，APIC1就会使用TEP池中的IP地址做出响应，然后这些节点的DHCP完成。
10. 主干从Pod1的所有其他节点中继DHCP发现。（这假设主干和枝叶交换机之间存在全网状，这是建议的且是典型的架构）。
11. 一旦连接到其他APIC的枝叶节点注册，即可通过它们之间的TCP通信建立APIC集群。确保完成APIC2和APIC3上的设置对话框。
12. 确认所有APIC已形成集群且完全适合。如果是这种情况，交换矩阵发现已完成。

从4.2开始，交换矩阵节点上可以使用新的CLI命令来帮助诊断常见发现问题。以下各节将介绍所执行的检查并提供其他验证命令以帮助排除故障。

```
leaf101# show discoveryissues
Checking the platform type.....LEAF!
Check01 - System state - in-service [ok]
Check02 - DHCP status [ok]
TEP IP: 10.0.72.67 Node Id: 101 Name: leaf101
Check03 - AV details check [ok]
Check04 - IP rechability to apic [ok]
Ping from switch to 10.0.0.1 passed
Check05 - infra VLAN received [ok]
infra vLAN:3967
Check06 - LLDP Adjacency [ok]
Found adjacency with SPINE
Found adjacency with APIC
Check07 - Switch version [ok]
version: n9000-14.2(1j) and apic version: 4.2(1j)
Check08 - FPGA/BIOS out of sync test [ok]
Check09 - SSL check [check]
SSL certificate details are valid
Check10 - Downloading policies [ok]
Check11 - Checking time [ok]
2019-09-11 07:15:53
Check12 - Checking modules, power and fans [ok]
```

Check01 — 系统状态

当枝叶已分配节点ID并注册到交换矩阵时，它将开始下载其引导程序，然后转换到**服务中**状态。

```
Check01 - System state - out-of-service [FAIL]
```

```
Check01 - System state - downloading-boot-script [FAIL]
```

要验证枝叶的当前状态，用户可以运行**moquery -c topSystem**

```
leaf101# moquery -c topSystem
Total Objects shown: 1
```

```
# top.System
```

```
address          : 10.0.72.67
bootstrapState   : done
...
serial           : FDO20160TPS
serverType       : unspecified
siteId           : 1
state            : in-service
status           :
systemUpTime     : 00:18:17:41.000
tepPool          : 10.0.0.0/16
unicastXrEpLearnDisable : no
version          : n9000-14.2(1j)
virtualMode      : no
```

Check02 — DHCP状态

```
Check02 - DHCP status [FAIL]
ERROR: node Id not configured
ERROR: Ip not assigned by dhcp server
ERROR: Address assigner's IP not populated
TEP IP: unknown Node Id: unknown Name: unknown
```

枝叶需要通过DHCP从APIC1接收TEP地址，然后建立到其他APIC的IP连接。枝叶的**物理TEP(PTEP)**已分配给loopback0。如果未分配地址，则用户可以验证枝叶正在使用tpcdump实用程序发送DHCP发现。请注意，我们将使用接口kpm_inb，该接口允许您查看所有CPU带内控制平面网络流量。

```
(none)# tcpdump -ni kpm_inb port 67 or 68
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on kpm_inb, link-type EN10MB (Ethernet), capture size 65535 bytes
16:40:11.041148 IP 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from a0:36:9f:c7:a1:0c,
length 300
^C
1 packets captured
1 packets received by filter
0 packets dropped by kernel
```

用户还可以验证dhcpd在APIC上运行并在bond0子接口上侦听。绑定接口表示面向APIC端口的交换矩阵。我们将使用格式bond0.<infra VLAN>。

```
apic1# ps aux | grep dhcp
root      18929  1.3  0.2 818552 288504 ?        Ssl  Sep26   87:19 /mgmt//bin/dhcpd.bin -f -4 -cf
/data//dhcp/dhcpd.conf -lf /data//dhcp/dhcpd.lease -pf /var/run//dhcpd.pid --no-pid bond0.3967
admin    22770   0.0  0.0   9108   868 pts/0    S+   19:42   0:00 grep dhcp
```

Check03 — AV详细信息

```
Check03 - AV details check [ok]
```

枝叶将验证注册的APIC是否具有有效范围内的TEP池的IP。如果尚未记录APIC信息，此检查将通过。用户可通过“acidiag avread”命令从枝叶节点的角度查看当前APIC信息。请注意，在以下示例中，当枝叶/主干提示符显示(none)#时，表明枝叶/主干尚未成为交换矩阵的成员。

```
(none)# acidiag avread
Cluster of 0 lm(t):0(zeroTime) appliances (out of targeted 0 lm(t):0(zeroTime)) with
FABRIC_DOMAIN name=Undefined Fabric Domain Name set to version= lm(t):0(zeroTime);
discoveryMode=PERMISSIVE lm(t):0(zeroTime); drrMode=OFF lm(t):0(zeroTime)
-----
```

```
clusterTime=<diff=0 common=2019-10-01T18:51:50.315+00:00 local=2019-10-01T18:51:50.315+00:00
pF=<displForm=1 offsSt=0 offsVlu=0 lm(t):0(zeroTime)>>
```

```
-----
leaf101# acidiag avread
Cluster of 3 lm(t):0(2019-09-30T18:45:10.320-04:00) appliances (out of targeted 3 lm(t):0(2019-
10-01T14:52:55.217-04:00)) with FABRIC_DOMAIN name=ACIFabric1 set to version=apic-4.2(1j)
lm(t):0(2019-10-01T14:52:55.217-04:00); discoveryMode=PERMISSIVE lm(t):0(1969-12-
31T20:00:00.003-04:00); drrMode=OFF lm(t):0(1969-12-31T20:00:00.003-04:00); kafkaMode=OFF
lm(t):0(1969-12-31T20:00:00.003-04:00)
    appliance id=1 address=10.0.0.1 lm(t):2(2019-09-27T17:32:08.669-04:00) tep
address=10.0.0.0/16 lm(t):1(2019-07-09T19:41:24.672-04:00) routable address=192.168.1.1
lm(t):2(2019-09-30T18:37:48.916-04:00) oob address=0.0.0.0 lm(t):0(zeroTime) version=4.2(1j)
lm(t):1(2019-09-30T18:37:49.011-04:00) chassisId=c67d1076-a2a2-11e9-874e-a390922be712
lm(t):1(2019-09-30T18:37:49.011-04:00) capabilities=0X3EEFFFFFFFFF--0X2020--0X1 lm(t):1(2019-09-
26T09:32:20.747-04:00) rK=(stable,absent,0) lm(t):0(zeroTime) aK=(stable,absent,0)
lm(t):0(zeroTime) oobrK=(stable,absent,0) lm(t):0(zeroTime) oobaK=(stable,absent,0)
lm(t):0(zeroTime) cntrlSbst=(APPROVED, FCH1929V153) lm(t):1(2019-10-01T12:46:44.711-04:00)
(targetMbSn= lm(t):0(zeroTime), failoverStatus=0 lm(t):0(zeroTime)) podId=1 lm(t):1(2019-09-
26T09:26:49.422-04:00) commissioned=YES lm(t):101(2019-09-30T18:45:10.320-04:00) registered=YES
lm(t):3(2019-09-05T11:42:41.371-04:00) standby=NO lm(t):0(zeroTime) DRR=NO lm(t):101(2019-09-
30T18:45:10.320-04:00) apicX=NO lm(t):0(zeroTime) virtual=NO lm(t):0(zeroTime) active=YES
    appliance id=2 address=10.0.0.2 lm(t):2(2019-09-26T09:47:34.709-04:00) tep
address=10.0.0.0/16 lm(t):2(2019-09-26T09:47:34.709-04:00) routable address=192.168.1.2
lm(t):2(2019-09-05T11:45:36.861-04:00) oob address=0.0.0.0 lm(t):0(zeroTime) version=4.2(1j)
lm(t):2(2019-09-30T18:37:48.913-04:00) chassisId=611febfe-89c1-11e8-96b1-c7a7472413f2
lm(t):2(2019-09-30T18:37:48.913-04:00) capabilities=0X3EEFFFFFFFFF--0X2020--0X7 lm(t):2(2019-09-
26T09:53:07.047-04:00) rK=(stable,absent,0) lm(t):0(zeroTime) aK=(stable,absent,0)
lm(t):0(zeroTime) oobrK=(stable,absent,0) lm(t):0(zeroTime) oobaK=(stable,absent,0)
lm(t):0(zeroTime) cntrlSbst=(APPROVED, FCH2045V1X2) lm(t):2(2019-10-01T12:46:44.710-04:00)
(targetMbSn= lm(t):0(zeroTime), failoverStatus=0 lm(t):0(zeroTime)) podId=1 lm(t):2(2019-09-
26T09:47:34.709-04:00) commissioned=YES lm(t):101(2019-09-30T18:45:10.320-04:00) registered=YES
lm(t):2(2019-09-26T09:47:34.709-04:00) standby=NO lm(t):0(zeroTime) DRR=NO lm(t):101(2019-09-
30T18:45:10.320-04:00) apicX=NO lm(t):0(zeroTime) virtual=NO lm(t):0(zeroTime) active=YES
    appliance id=3 address=10.0.0.3 lm(t):3(2019-09-26T10:12:34.114-04:00) tep
address=10.0.0.0/16 lm(t):3(2019-09-05T11:42:27.199-04:00) routable address=192.168.1.3
lm(t):2(2019-10-01T13:19:08.626-04:00) oob address=0.0.0.0 lm(t):0(zeroTime) version=4.2(1j)
lm(t):3(2019-09-30T18:37:48.904-04:00) chassisId=99bade8c-cff3-11e9-bba7-5b906a49dc39
lm(t):3(2019-09-30T18:37:48.904-04:00) capabilities=0X3EEFFFFFFFFF--0X2020--0X4 lm(t):3(2019-09-
26T10:18:13.149-04:00) rK=(stable,absent,0) lm(t):0(zeroTime) aK=(stable,absent,0)
lm(t):0(zeroTime) oobrK=(stable,absent,0) lm(t):0(zeroTime) oobaK=(stable,absent,0)
lm(t):0(zeroTime) cntrlSbst=(APPROVED, FCH1824V2VR) lm(t):3(2019-10-01T12:48:03.726-04:00)
(targetMbSn= lm(t):0(zeroTime), failoverStatus=0 lm(t):0(zeroTime)) podId=2 lm(t):3(2019-09-
26T10:12:34.114-04:00) commissioned=YES lm(t):101(2019-09-30T18:45:10.320-04:00) registered=YES
lm(t):2(2019-09-05T11:42:54.935-04:00) standby=NO lm(t):0(zeroTime) DRR=NO lm(t):101(2019-09-
30T18:45:10.320-04:00) apicX=NO lm(t):0(zeroTime) virtual=NO lm(t):0(zeroTime) active=YES
```

```
-----
clusterTime=<diff=15584 common=2019-10-01T14:53:01.648-04:00 local=2019-10-01T14:52:46.064-04:00
pF=<displForm=0 offsSt=0 offsVlu=-14400 lm(t):21(2019-09-26T10:40:35.412-04:00)>>
```

Check04 — 与APIC的IP可达性

枝叶收到IP地址后，它将尝试与APIC建立TCP会话，并开始下载其配置的过程。用户可以使用“iping”实用程序验证与APIC的IP连接。

```
leaf101# iping -v overlay-1 10.0.0.1
PING 10.0.0.1 (10.0.0.1) from 10.0.0.30: 56 data bytes
64 bytes from 10.0.0.1: icmp_seq=0 ttl=64 time=0.651 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.474 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.477 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.54 ms
```

```
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.5 ms
```

```
--- 10.0.0.1 ping statistics --- 5 packets transmitted, 5 packets received, 0.00% packet loss  
round-trip min/avg/max = 0.474/0.528/0.651 ms
```

Check05 — 基础VLAN

```
Check05 - infra VLAN received [ok]
```

仅当节点连接到存在APIC的Pod时，基础VLAN检查才会成功。如果不是这种情况，用户可以忽略该消息，因为检查预期会失败。

枝叶将根据从其他ACI节点收到的LLDP数据包确定基础设施VLAN。当交换机处于发现状态时，将接受它收到的第一个数据包。

```
(none)# moquery -c lldpInst
```

```
Total Objects shown: 1
```

```
# lldp.Inst  
adminSt      : enabled  
childAction  :  
ctrl        :  
dn          : sys/lldp/inst  
holdTime    : 120  
infraVlan   : 3967  
initDelayTime : 2  
lcOwn       : local  
modTs       : 2019-09-12T07:25:33.194+00:00  
monPolDn    : uni/fabric/monfab-default  
name        :  
operErr     :  
optTlvSel   : mgmt-addr,port-desc,port-vlan,sys-cap,sys-desc,sys-name  
rn          : inst  
status      :  
sysDesc     : topology/pod-1/node-101  
txFreq     : 30
```

```
(none)# show vlan encap-id 3967
```

VLAN Name	Status	Ports
8 infra:default	active	Eth1/1

VLAN Type	Vlan-mode
8 enet	CE

如果尚未在连接到APIC的交换机端口接口上编程基础设施VLAN，请检查枝叶交换机是否检测到布线问题。

```
(none)# moquery -c lldpIf -f 'lldp.If.wiringIssues!=""'
```

```
Total Objects shown: 1
```

```
# lldp.If  
id          : eth1/1  
adminRxSt  : enabled  
adminSt    : enabled  
adminTxSt  : enabled  
childAction :  
descr      :
```

```
dn          : sys/lldp/inst/if-[eth1/1]
lcOwn      : local
mac        : E0:0E:DA:A2:F2:83
modTs     : 2019-09-30T18:45:22.323+00:00
monPolDn   : uni/fabric/monfab-default
name       :
operRxSt   : enabled
operTxSt   : enabled
portDesc   :
portMode   : normal
portVlan   : unspecified
rn         : if-[eth1/1]
status     :
sysDesc    :
wiringIssues : infra-vlan-mismatch
```

Check06 — LLDP邻接关系

```
Check06 - LLDP Adjacency [FAIL]
Error: leaf not connected to any spine
```

为了确定哪些端口连接到其他ACI设备，枝叶必须接收来自其他交换矩阵节点的LLDP。要验证是否已收到LLDP，用户可以选中“show lldp neighbors”。

```
(none)# show lldp neighbors
Capability codes:
 (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
 (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID          Local Intf      Hold-time  Capability  Port ID
-----
apic1              Eth1/1         120                eth2-1
apic2              Eth1/2         120                eth2-1
switch             Eth1/51        120                BR          Eth2/32
switch             Eth1/54        120                BR          Eth1/25
Total entries displayed: 4
```

Check07 — 交换机版本

```
Check07 - Switch version [ok]
version: n9000-14.2(1j) and apic version: 4.2(1j)
```

如果APIC和枝叶版本不同，交换矩阵发现可能会失败。要验证枝叶上运行的版本，请使用“show version”或“vsh -c 'show version'”。

```
(none)# show version
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Documents:
http://www.cisco.com/en/US/products/ps9372/tsd\_products\_support\_series\_home.html Copyright (c)
2002-2014, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
```

Software

```

BIOS:          version 07.66
kickstart:     version 14.2(1j) [build 14.2(1j)]
  system:      version 14.2(1j) [build 14.2(1j)]
  PE:         version 4.2(1j)
BIOS compile time:    06/11/2019
kickstart image file is: /bootflash/aci-n9000-dk9.14.2.1j.bin
kickstart compile time: 09/19/2019 07:57:41 [09/19/2019 07:57:41]
system image file is:  /bootflash/auto-s
system compile time:  09/19/2019 07:57:41 [09/19/2019 07:57:41]
...

```

相同的命令也适用于APIC。

```

apic1# show version
Role      Pod      Node      Name      Version
-----
controller 1        1         apic1     4.2(1j)
controller 1        2         apic2     4.2(1j)
controller 2        3         apic3     4.2(1j)
leaf       1        101       leaf101   n9000-14.2(1j)
leaf       1        102       leaf102   n9000-14.2(1j)
leaf       1        103       leaf103   n9000-14.2(1j)
spine     1        1001      spine1    n9000-14.2(1j)
spine     1        1002      spine2    n9000-14.2(1j)

```

Check08 — FPGA/EPLD/BIOS不同步

FPGA、EPLD和BIOS版本可能会影响枝叶节点按预期启动模块的能力。如果这些设置已过时，交换机接口可能无法启动。用户可以使用以下moquery命令验证运行和预期版本的FPGA、EPLD和BIOS。

```

(none)# moquery -c firmwareCardRunning
Total Objects shown: 2

# firmware.CardRunning
biosVer      : v07.66(06/11/2019)
childAction  :
descr        :
dn           : sys/ch/supslot-1/sup/running
expectedVer  : v07.65(09/04/2018)  interimVer    : 14.2(1j)
internalLabel :
modTs        : never
mode         : normal
monPolDn     : uni/fabric/monfab-default
operSt       : ok
rn           : running
status       :
ts           : 1970-01-01T00:00:00.000+00:00
type         : switch
version      : 14.2(1j)

# firmware.CardRunning
biosVer      : v07.66(06/11/2019)
childAction  :
descr        :
dn           : sys/ch/lcslot-1/lc/running
expectedVer  : v07.65(09/04/2018)  interimVer    : 14.2(1j)
internalLabel :
modTs        : never
mode         : normal
monPolDn     : uni/fabric/monfab-default

```

```
operSt      : ok
rn          : running
status      :
ts          : 1970-01-01T00:00:00.000+00:00
type        : switch
version     : 14.2(1j)
```

```
(none)# moquery -c firmwareCompRunning
```

```
Total Objects shown: 2
```

```
# firmware.CompRunning
childAction :
descr       :
dn          : sys/ch/supslot-1/sup/fpga-1/running
expectedVer : 0x14 internalLabel :
modTs       : never
mode        : normal
monPolDn    : uni/fabric/monfab-default
operSt      : ok
rn          : running
status      :
ts          : 1970-01-01T00:00:00.000+00:00
type        : controller
version     : 0x14
```

```
# firmware.CompRunning
childAction :
descr       :
dn          : sys/ch/supslot-1/sup/fpga-2/runnin
expectedVer : 0x4
internalLabel :
modTs       : never
mode        : normal
monPolDn    : uni/fabric/monfab-default
operSt      : ok
rn          : running
status      :
ts          : 1970-01-01T00:00:00.000+00:00
type        : controller
version     : 0x4
```

如果运行的FPGA版本与预期的FPGA版本不匹配，则可以使用“枝叶/主干EPLD/FPGA不正确，F1582”场景“交换矩阵发现”一章“设备更换”一节中的步骤进行更新。

Check09 — SSL检查

```
Check09 - SSL check [check]
```

```
SSL certificate details are valid
```

在所有交换矩阵节点之间使用SSL通信来确保控制平面流量的加密。使用的SSL证书是在制造过程中安装的，根据机箱的序列号生成。主题的格式应如下：

```
subject= /serialNumber=PID:N9K-C93xxxxx SN:FDOxxxxxxxx/CN=FDOxxxxxxxx
```

要在发现交换机时验证SSL证书，请使用以下命令。

```
(none)# cd /securedata/ssl && openssl x509 -noout -subject -in server.crt
subject= /serialNumber=PID:N9K-C93180YC-EX SN:FDO20432LH1/CN=FDO20432LH1
```

请注意，如果交换机节点仍处于发现状态，则以上命令将仅作为非根用户运行。

可使用以下命令找到机箱序列号。

```
(none)# show inventory
NAME: "Chassis", DESCR: "Nexus C93180YC-EX Chassis"
PID: N9K-C93180YC-EX , VID: V00 , SN: FDO20160TPS
...
```

此外，证书必须在当前有效。要查看证书的有效日期，请使用openssl命令中的“— dates”标志。

```
(none)# cd /securedata/ssl && openssl x509 -noout -dates -in server.crt
notBefore=Nov 28 17:17:05 2016 GMT
notAfter=Nov 28 17:27:05 2026 GMT
```

Check10 — 下载策略

```
Check10 - Downloading policies [FAIL]
Registration to all PM shards is not complete
Policy download is not complete
```

枝叶对APIC具有IP可达性后，它将从APIC下载其配置，APIC将确认下载已完成。可使用以下命令查看此进程的状态。

```
(none)# moquery -c pconsBootStrap
Total Objects shown: 1

# pcons.BootStrap
allLeaderAked          : no
allPortsInService      : yes
allResponsesFromLeader : yes
canBringPortInService  : no
childAction             :
completedPolRes        : no
dn                      : rescont/bootstrap
lcOwn                   : local
modTs                   : 2019-09-27T22:52:48.729+00:00
rn                      : bootstrap
state                   : completed
status                  :
timerTicks              : 360
try                     : 0
worstCaseTaskTry       : 0
```

Check11 — 时间

```
Check11 - Checking time [ok]
2019-10-01 17:02:34
```

此检查显示用户的当前时间。如果APIC和交换机时间之间存在太多差异，则发现可能会失败。在APIC上，可以使用date命令检查时间。

```
apic1# date
Tue Oct 1 14:35:38 UTC 2019
```

Check12 — 模块、PSU、风扇检查

为了使交换机能够连接到其他设备，模块需要启动并联机。这可以通过“show module”和“show

environment”命令进行验证。

(none)# **show module**

Mod	Ports	Module-Type	Model	Status
1	54	48x10/25G+6x40/100G Switch	N9K-C93180YC-EX	ok

Mod	Sw	Hw
1	14.2(1j)	0.3050

Mod	MAC-Address(es)	Serial-Num
1	e0-0e-da-a2-f2-83 to e0-0e-da-a2-f2-cb	FDO20160TPS

Mod	Online Diag Status
1	pass

(none)# **show environment**

Power Supply:
Voltage: 12.0 Volts

Power Supply	Model	Actual Output (Watts)	Total Capacity (Watts)	Status
1	NXA-PAC-650W-PI	0 W	650 W	shut
2	NXA-PAC-650W-PI	171 W	650 W	ok

Module	Model	Actual Draw (Watts)	Power Allocated (Watts)	Status
1	N9K-C93180YC-EX	171 W	492 W	Powered-Up
fan1	NXA-FAN-30CFM-B	N/A	N/A	Powered-Up
fan2	NXA-FAN-30CFM-B	N/A	N/A	Powered-Up
fan3	NXA-FAN-30CFM-B	N/A	N/A	Powered-Up
fan4	NXA-FAN-30CFM-B	N/A	N/A	Powered-Up

N/A - Per module power not available

Power Usage Summary:

Power Supply redundancy mode (configured)	Non-Redundant (combined)
Power Supply redundancy mode (operational)	Non-Redundant (combined)

Total Power Capacity (based on configured mode)	650 W
Total Power of all Inputs (cumulative)	650 W
Total Power Output (actual draw)	171 W
Total Power Allocated (budget)	N/A
Total Power Available for additional modules	N/A

Fan:

Fan	Model	Hw	Status
Fan1(sys_fan1)	NXA-FAN-30CFM-B	--	ok
Fan2(sys_fan2)	NXA-FAN-30CFM-B	--	ok
Fan3(sys_fan3)	NXA-FAN-30CFM-B	--	ok
Fan4(sys_fan4)	NXA-FAN-30CFM-B	--	ok
Fan_in_PS1	--	--	unknown

```
Fan_in_PS2      --          --          ok
Fan Speed: Zone 1: 0x7f
Fan Air Filter : Absent
```

Temperature:

Module	Sensor	MajorThresh (Celsius)	MinorThres (Celsius)	CurTemp (Celsius)	Status
1	Inlet(1)	70	42	35	normal
1	outlet(2)	80	70	37	normal
1	x86 processor(3)	90	80	38	normal
1	Sugarbowl(4)	110	90	60	normal
1	Sugarbowl vrm(5)	120	110	50	normal

如果模块未联机，请重新安装模块并检查FPGA、EPLD或BIOS不匹配。

中断方案示例

场景1 — 第一个枝叶未出现在交换矩阵成员资格中

在此场景中，用户在完成设置脚本后登录APIC1，且交换矩阵成员中未显示交换机。为了成功发现第一个枝叶，APIC应在发现阶段从枝叶接收DHCP发现。

检查APIC1是否正在发送与设置脚本中设置的参数匹配的LLDP TLV。

```
apic1# acidiag run lldptool out eth2-1
Chassis ID TLV
  MAC: e8:65:49:54:88:a1
Port ID TLV
  MAC: e8:65:49:54:88:a1
Time to Live TLV
  120
Port Description TLV
  eth2-1
System Name TLV
  apic1
System Description TLV
  topology/pod-1/node-1
Management Address TLV
  IPv4: 10.0.0.1
  Ifindex: 4
Cisco Port State TLV
  1
Cisco Node Role TLV
  0
Cisco Node ID TLV
  1
Cisco POD ID TLV
  1
Cisco Fabric Name TLV
  ACIFabric1
Cisco Appliance Vector TLV
  Id: 1
  IPv4: 10.0.0.1
  UUID: c67d1076-a2a2-11e9-874e-a390922be712
Cisco Node IP TLV
  IPv4:10.0.0.1
Cisco Port Role TLV
  2
```

```
Cisco Infra VLAN TLV
3967
```

```
Cisco Serial Number TLV
FCH1929V153
```

```
Cisco Authentication Cookie TLV
1372058352
```

```
Cisco Standby APIC TLV
0
```

```
End of LLDPDU TLV
```

还验证APIC1是否从直接连接的枝叶节点接收LLDP。

```
apic1# acidiag run lldptool in eth2-1
```

```
Chassis ID TLV
```

```
MAC: e0:0e:da:a2:f2:83
```

```
Port ID TLV
```

```
Local: Eth1/1
```

```
Time to Live TLV
```

```
120
```

```
Port Description TLV
```

```
Ethernet1/1
```

```
System Name TLV
```

```
switch
```

```
System Description TLV
```

```
Cisco Nexus Operating System (NX-OS) Software 14.2(1j)
```

```
TAC support: http://www.cisco.com/tac Copyright (c) 2002-2020, Cisco Systems, Inc. All rights reserved.
```

```
System Capabilities TLV
```

```
System capabilities: Bridge, Router
```

```
Enabled capabilities: Bridge, Router
```

```
Management Address TLV
```

```
MAC: e0:0e:da:a2:f2:83
```

```
Ifindex: 83886080
```

```
Cisco 4-wire Power-via-MDI TLV
```

```
4-Pair PoE supported
```

```
Spare pair Detection/Classification not required
```

```
PD Spare pair Desired State: Disabled
```

```
PSE Spare pair Operational State: Disabled
```

```
Cisco Port Mode TLV
```

```
0
```

```
Cisco Port State TLV
```

```
1
```

```
Cisco Serial Number TLV
```

```
FDO20160TPS
```

```
Cisco Model TLV
```

```
N9K-C93180YC-EX
```

```
Cisco Firmware Version TLV
```

```
n9000-14.2(1j)
```

```
Cisco Node Role TLV
```

```
1
```

```
Cisco Infra VLAN TLV
```

```
3967
```

```
Cisco Node ID TLV
```

```
0
```

```
End of LLDPDU TLV
```

如果APIC1从直接连接的枝叶节点接收LLDP，枝叶应在连接到APIC的端口上编程基础VLAN。此VLAN编程可通过“show vlan encap-id <x>”命令进行验证，其中“x”是已配置的基础VLAN。

```
(none)# show vlan encap-id 3967
```

```
VLAN Name                               Status    Ports
```

```
-----
```

```
8      infra:default          active      Eth1/1
```

```
VLAN Type  Vlan-mode
-----
8      enet   CE
```

如果基础VLAN尚未编程，请检查枝叶节点检测到的布线问题。

```
(none)# moquery -c lldpIf -f 'lldp.If.wiringIssues!=""'
Total Objects shown: 1
```

```
# lldp.If
id          : eth1/1
adminRxSt   : enabled
adminSt     : enabled
adminTxSt   : enabled
childAction :
descr       :
dn          : sys/lldp/inst/if-[eth1/1]
lcOwn       : local
mac         : E0:0E:DA:A2:F2:83
modTs       : 2019-09-30T18:45:22.323+00:00
monPolDn    : uni/fabric/monfab-default
name        :
operRxSt    : enabled
operTxSt    : enabled
portDesc    :
portMode    : normal
portVlan    : unspecified
rn          : if-[eth1/1]
status      :
sysDesc     :
wiringIssues : infra-vlan-mismatch
```

当布线问题属性设置为“infra-vlan-mismatch”时，表示枝叶已获知与APIC发送的值不同的基础设施VLAN（APIC发送值可以使用命令“moquery -c lldpInst”进行验证）。如果枝叶从曾经是另一个交换矩阵一部分的节点接收LLDP，则可能会发生这种情况。实际上，发现中的节点将接受通过LLDP接收的第一个基础VLAN。要解决此问题，请删除此枝叶与其他ACI节点（APIC除外）之间的连接，然后使用“acdiag touch clean”和“reload”命令清除重新加载交换机。交换机启动后，检验是否正确配置了基础设施VLAN。如果为true，则连接可以恢复到其他节点，并且用户可以继续设置ACI交换矩阵。

场景2 — 其他APIC不加入集群

在此方案中，已发现所有交换矩阵节点，但APIC2和3尚未加入APIC集群。

验证APIC之间的设置脚本值。必须匹配的值为：

- 交换矩阵域
- 交换矩阵ID
- TEP池
- 基础设施VLAN
- GIPo
- 集群大小
- 固件版本

```
apic1# cat /data/data_admin/sam_exported.config
```

Setup for Active and Standby APIC

```
fabricDomain = ACIFabric1
fabricID = 1
systemName = apic1
controllerID = 1
tepPool = 10.0.0.0/16
infraVlan = 3967
GIPO = 225.0.0.0/15
clusterSize = 3
standbyApic = NO
enableIPv4 = Y
enableIPv6 = N
firmwareVersion = 4.2(1j)
ifcIpAddr = 10.0.0.1
apicX = NO
podId = 1
oobIpAddr = 10.48.22.69/24
```

在所有3个APIC上使用“acidiag cluster”命令检验常见问题。

```
apic1# acidiag cluster
Admin password:
```

```
Product-name = APIC-SERVER-M1
Serial-number = FCH1906V1XV
Running...
```

```
Checking Core Generation: OK
Checking Wiring and UUID: OK
Checking AD Processes: Running
Checking All Apics in Commission State: OK
Checking All Apics in Active State: OK
Checking Fabric Nodes: OK
Checking Apic Fully-Fit: OK
Checking Shard Convergence: OK
Checking Leadership Degration: Optimal leader for all shards
Ping OOB IPs:
APIC-1: 10.48.22.69 - OK
APIC-2: 10.48.22.70 - OK
APIC-3: 10.48.22.71 - OK
Ping Infra IPs:
APIC-1: 10.0.0.1 - OK
APIC-2: 10.0.0.2 - OK
APIC-3: 10.0.0.3 - OK
Checking APIC Versions: Same (4.2(1j))
Checking SSL: OK
```

Done!

最后，使用“avread”验证这些设置是否在所有的APIC中匹配。请注意，此命令与显示类似输出的典型“acidiag avread”命令不同，但会进行解析以便于使用。

```
apic1# avread
Cluster:
```

```
-----
fabricDomainName      ACIFabric1
discoveryMode         PERMISSIVE
clusterSize           3
version               4.2(1j)
drrMode               OFF
operSize              3
```

APICs:

```
-----  
version          APIC 1          APIC 2          APIC 3  
address          10.0.0.1       10.0.0.2       10.0.0.3  
oobAddress       10.48.22.69/24 10.48.22.70/24 10.48.22.71/24  
routableAddress  0.0.0.0        0.0.0.0        0.0.0.0  
tepAddress       10.0.0.0/16    10.0.0.0/16    10.0.0.0/16  
podId            1              1              1  
chassisId        3c9e5024-.-5a78727f 573e12c0-.-6b8da0e5 44c4bf18-.-20b4f52&  
cntrlSbst_serial (APPROVED,FCH1906V1XV) (APPROVED,FCH1921V1Q9) (APPROVED,FCH1906V1PW)  
active           YES            YES            YES  
flags            cra-          cra-          cra-  
health           255           255           255  
apic1#
```

场景3 — 交换矩阵成员中未显示主干

在此场景中，在交换矩阵中发现了第一个枝叶，但是在“交换矩阵成员资格”子菜单下没有出现用于发现的主干。

验证从枝叶到主干的物理连接。在下面的示例中，枝叶交换机通过接口e1/49连接到主干。

```
leaf101# show int eth1/49  
Ethernet1/49 is up  
admin state is up, Dedicated Interface  
Hardware: 1000/10000/100000/40000 Ethernet, address: 0000.0000.0000 (bia e00e.daa2.f3f3)  
MTU 9366 bytes, BW 100000000 Kbit, DLY 1 usec  
reliability 255/255, txload 1/255, rxload 1/255  
Encapsulation ARPA, medium is broadcast  
Port mode is routed  
full-duplex, 100 Gb/s  
...
```

如果端口处于out-of-service状态，请检查脊柱上是否已从直接连接的枝叶接收LLDP。

```
(none)# show lldp neighbors  
Capability codes:  
(R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device  
(W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other  
Device ID          Local Intf      Hold-time  Capability  Port ID  
leaf102            Eth2/27        120       BR          Eth1/53  
leaf103            Eth2/29        120       BR          Eth1/49  
leaf101            Eth2/32        120       BR          Eth1/51  
Total entries displayed: 3
```

另一个验证是验证枝叶和主干之间不存在版本差异。如果存在，请通过将较新版本复制到主干的/bootflash来修复这种情况。然后，使用以下命令配置交换机以引导至软件：

```
(none)# ls -alh /bootflash  
total 3.0G  
drwxrwxr-x  3 root admin 4.0K Oct  1 20:21 .  
drwxr-xr-x 50 root root  1.3K Oct  1 00:22 ..  
-rw-r--r--  1 root root  3.5M Sep 30 21:24 CpuUsage.Log  
-rw-rw-rw-  1 root root  1.7G Sep 27 14:50 aci-n9000-dk9.14.2.1j.bin  
-rw-r--r--  1 root root  1.4G Sep 27 21:20 auto-s  
-rw-rw-rw-  1 root root    2 Sep 27 21:25 diag_bootup  
-rw-r--r--  1 root root   54 Oct  1 20:20 disk_log.txt  
-rw-rw-rw-  1 root root  693 Sep 27 21:23 libmon.logs
```

```
drwxr-xr-x 4 root root 4.0K Sep 26 15:24 lxc
-rw-r--r-- 1 root root 384K Oct 1 20:20 mem_log.txt
-rw-r--r-- 1 root root 915K Sep 27 21:10 mem_log.txt.old.gz
-rw-rw-rw- 1 root root 12K Sep 27 21:17 urib_api_log.txt
```

```
(none)# setup-bootvars.sh aci-n9000-dk9.14.2.1j.bin
In progress
In progress
In progress
In progress
Done
```

如果从Bootflash中持续删除新映像，则通过删除较旧的映像或auto-s文件确保文件夹未半；通过在交换机上使用“df -h”检查空间利用率。

设置引导变量后，重新加载交换机，交换机应引导至新版本。

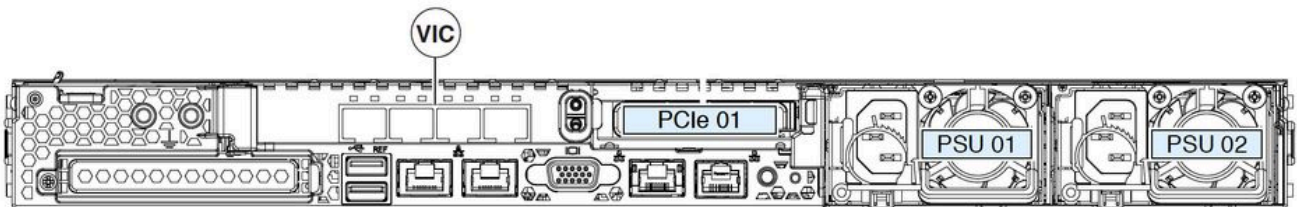
重新加载后可能需要FPGA、EPLD和BIOS验证。有关此问题的进一步故障排除，请参阅“枝叶/主干EPLD/FPGA不正确，F1582”子部分。

场景4 — 初次发现交换矩阵后，集群在完全适应和降级之间摇摆

如果在新交换矩阵设置后发生这种情况，则可能是由于连接到交换矩阵的APIC-M3或APIC-L3布线不正确造成的。您可以在连接到APIC的两台枝叶交换机上执行“show lldp neighbors”来确认此类错误布线。在执行此多次操作后，您会发现两台枝叶交换机看到的是相同的APIC接口。

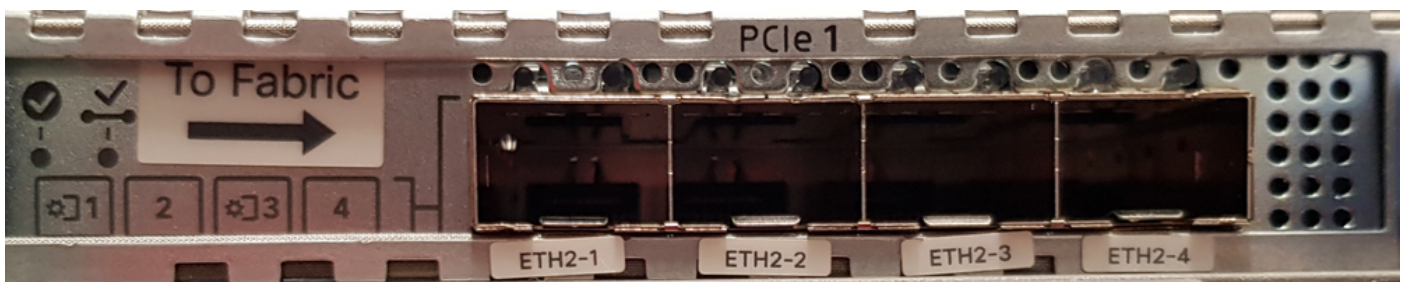
APIC-M3/L3服务器的背面如下所示：

APIC-M3/L3服务器的后视图



请注意，对于APIC-M3/L3,VIC卡有4个端口：ETH2-1、ETH2-2、ETH2-3和ETH2-4，如下所示：

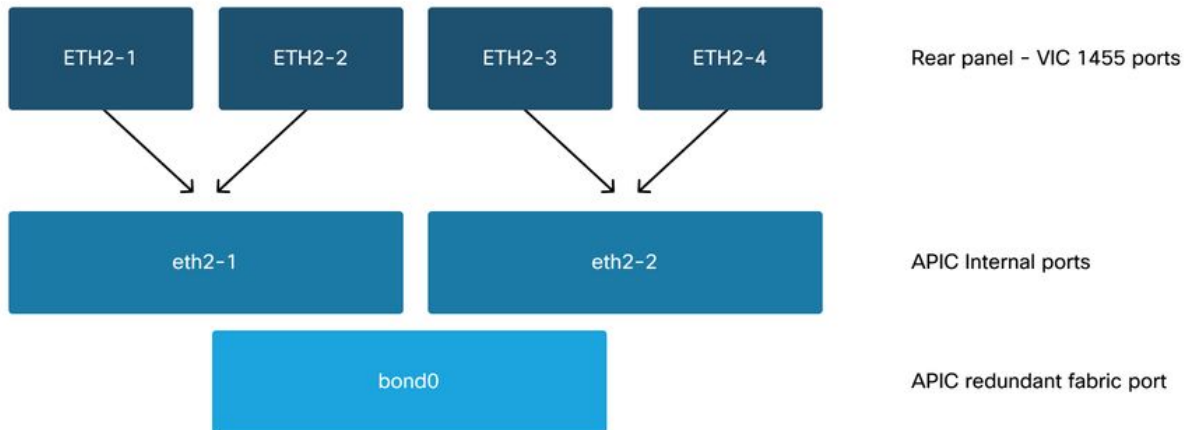
带标签的APIC VIC 1455视图



将APIC服务器连接到枝叶交换机的规则如下：

- 所有端口都必须具有相同的速度，10千兆或25千兆。
 - ETH2-1和ETH2-2是一个端口通道对，对应于APIC操作系统的eth2-1（“ifconfig”输出）。
 - ETH2-3和ETH2-4是另一个端口通道对，对应于APIC操作系统上的eth2-2（“ifconfig”输出）。
 - 每个端口通道对只允许一个连接。例如，将一根电缆连接到ETH2-1或ETH2-2，将另一根电缆连接到ETH2-3或ETH2-4(切勿在端口通道对中连接两个ETH。这将导致交换矩阵发现问题。)
- 为了进一步理解，以下是VIC端口映射到APIC绑定的表示。

VIC 1455端口 — APIC冗余交换矩阵端口



关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。