在Catalyst 9800无线LAN控制器(WLC)上配置移动拓扑

目录

简介

本文档介绍涵盖Catalyst 9800无线LAN控制器(WLC)和AireOS WLC之间的拓扑的移动配置方案。

先决条件

要求

建议掌握下列主题的相关知识：

- 对无线控制器的CLI或GUI访问。

使用的组件

- **AireOS WLC** 8.10 MR1或更高版本。您还可以使用 **Inter Release Controller Mobility (IRCM)** 特殊的8.5图像
- 9800 WLC、Cisco IOS® XE v17.3.4

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

配置

网络图

准则和限制

1. **Mobility Group** 9800开箱即用的名称为“default”。

注意：
1)如果WLC位于不同的子网中，请确保它们之间的端口UDP 16666和16667处于打开状态。
2)建议两个9800 WLC运行相同的版本，以便漫游的客户端在第3层漫游和访客锚点场景中具有一致的体验。

两个Catalyst 9800 WLC之间的移动隧道

此基本示例介绍如何跨两个9800控制器设置移动性。这通常用于访客接入（锚点），或允许客户端在控制器之间漫游并保持客户端身份。

在C9800上配置移动性时，首先要选择移动组名称。预填充的移动组名称是默认值，但您可以将其自定义为所需的值。

当快速第2层漫游时，必须在控制器之间配置相同的移动组名称 **Fast Transition (FT)** 或 **Cisco Centralized Key Management (CCKM)** 正在使用中。

默认情况下，机箱的基本以太网MAC地址，如所示 **show version** 在GUI上反映移动MAC地址。

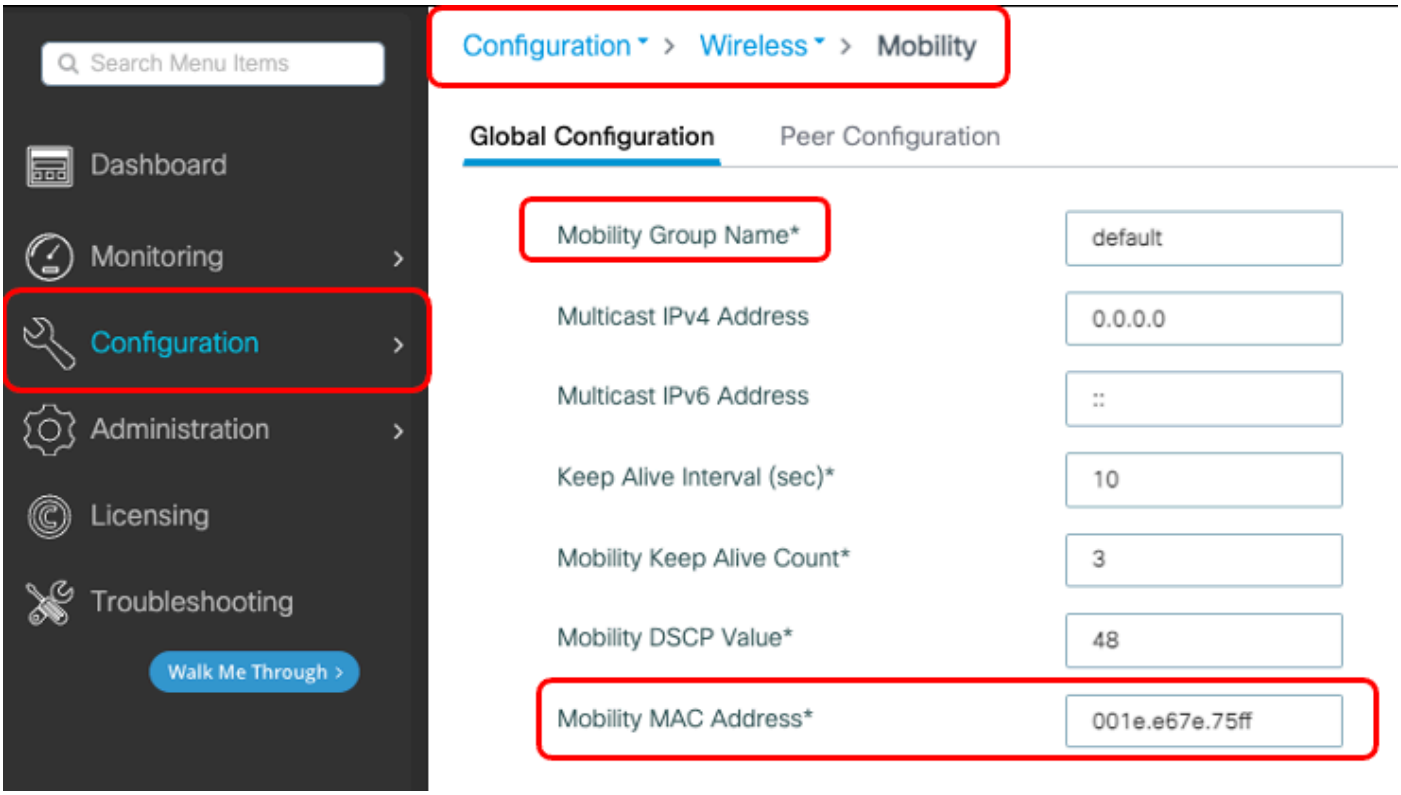
默认情况下，在CLI上，移动mac地址为0000.0000.000，如中所示 **show run all | inc mobility mac-address**

如果将9800配对 **High Availability (HA) Stateful Switchover (SSO):**

如果配置保留为默认值，并且机箱MAC地址用于形成移动隧道，则发生故障转移时，主用机箱和移动隧道将失败。

因此，强制为C9800 HA对配置移动MAC地址。

第1步：在GUI上，导航至 **Configuration > Wireless > Mobility > Global Configuration.**



通过CLI:

```
# config t
# wireless mobility mac-address  <AAAA.BBBB.CCCC>
# wireless mobility group name <mobility-group-name>
```

步骤1:收集两个9800 WLC的移动配置。

对于两个9800 WLC，请导航至 **Configuration > Wireless > Mobility > Global Configuration** 并注意其 **Mobility Group Name** 和 **Mobility MAC Address.**

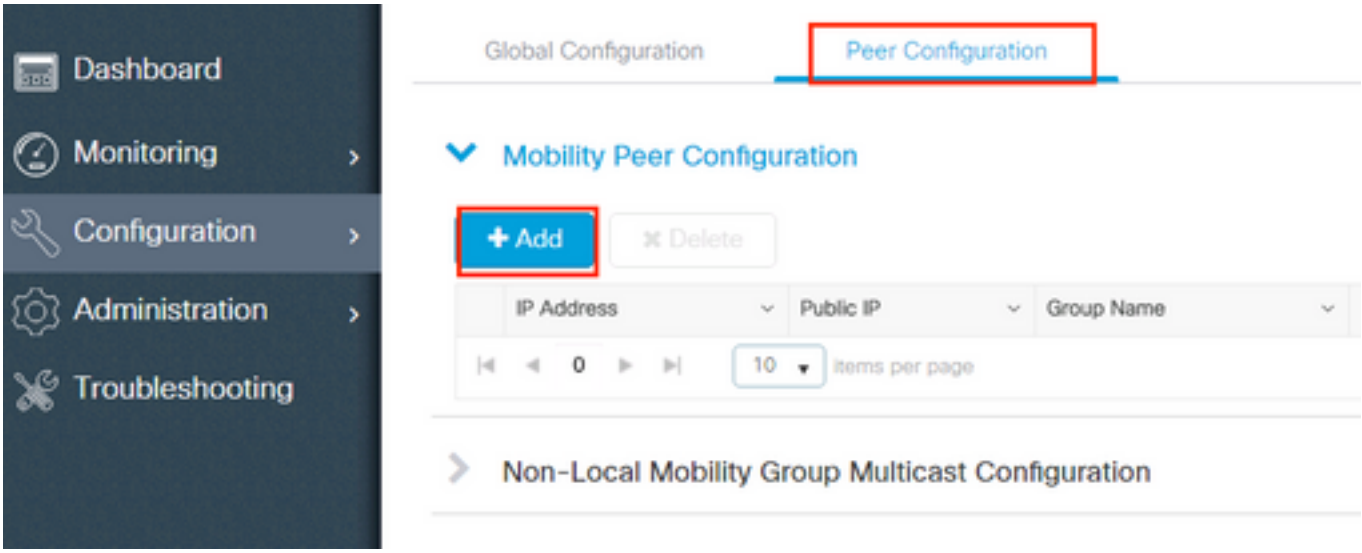
通过CLI:

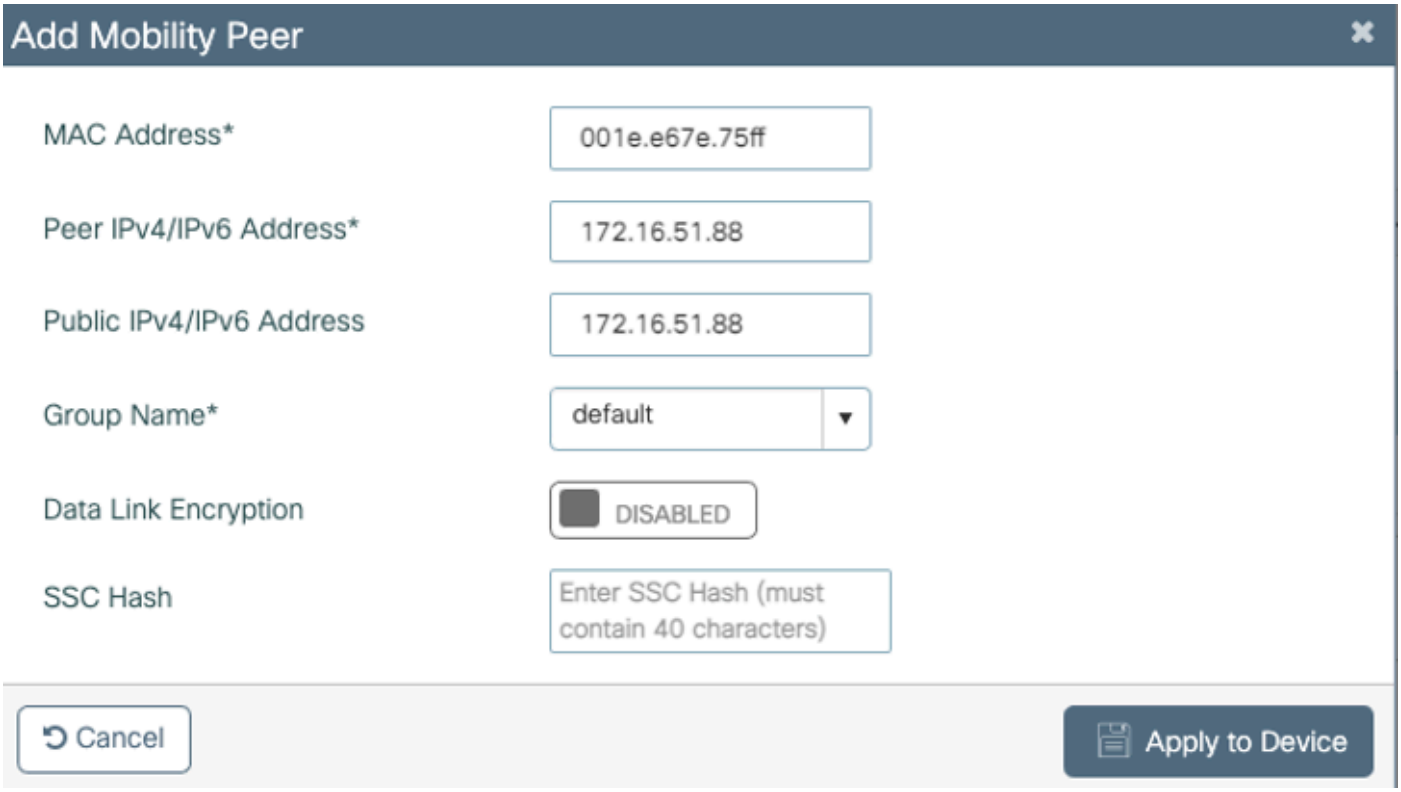
```
#show wireless mobility summary

Mobility Summary

Wireless Management VLAN: 2652
Wireless Management IP Address: 172.16.51.88
Wireless Management IPv6 Address:
Mobility Control Message DSCP Value: 48
Mobility Keepalive Interval/Count: 10/3
Mobility Group Name: default
Mobility Multicast Ipv4 address: 0.0.0.0
Mobility Multicast Ipv6 address: ::
Mobility MAC Address: 001e.e67e.75ff
Mobility Domain Identifier: 0x34ac
```

第二步：添加对等配置

导航至 **Configuration > Wireless > Mobility > Peer Configuration** 并输入对等控制器信息。对两个9800 WLC执行相同操作。

通过GUI:





通过CLI:

```
# config t
# wireless mobility group member mac-address <peer-mac-address> ip <peer-ip-address> group
<group-name> [ data-link-encryption ]
```

注意：或者，您可以启用数据链路加密。

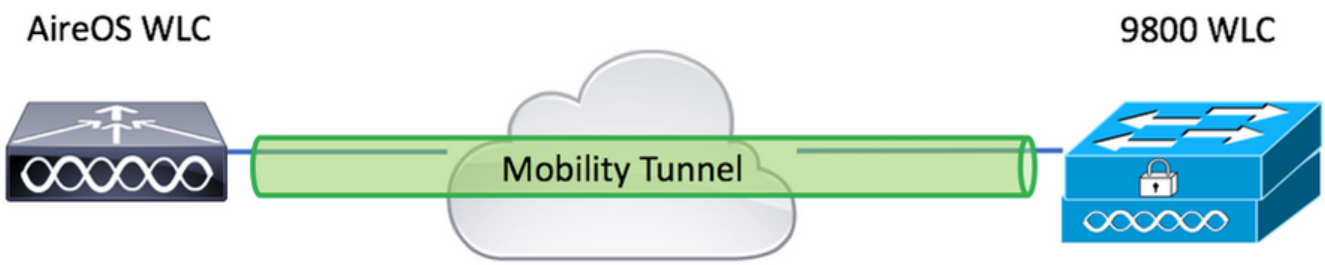
AireOS WLC和9800-CL控制器之间的移动隧道

此场景适用于 **brownfield** 在部署或控制器迁移期间，我们将网络划分到由AireOS控制器控制的接入点区域(AP)，另一个由9800控制。

建议将AP按物理或RF区域分布到控制器中，以便客户端在控制器之间移动时只在控制器之间漫游。

避免 **salt and pepper** 部署.或者，也可以将此移动拓扑用于 **guest anchor** 其中9800充当外部控制器，AireOS充当锚点控制器。

网络图



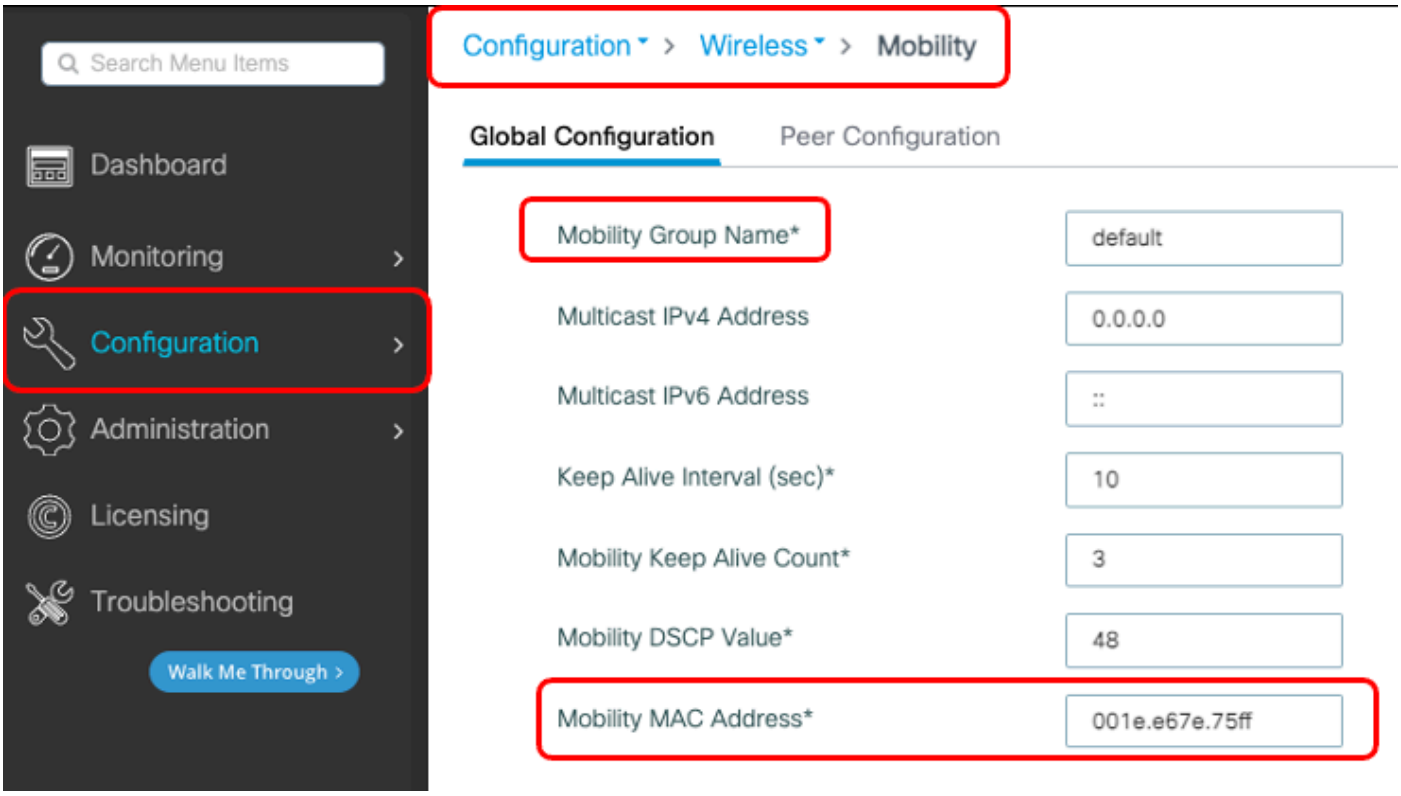
AireOS WLC配置

如果您的9800控制器处于 **High Availability** ，确保已配置移动MAC地址。

步骤1:收集9800 WLC移动性信息。

通过GUI:

导航至 **Configuration > Wireless > Mobility > Global Configuration** 并注意其 **Mobility Group Name** 和 **Mobility MAC Address.**



通过CLI:

```
#show wireless mobility summary

Mobility Summary

Wireless Management VLAN: 2652
Wireless Management IP Address: 172.16.51.88
Wireless Management IPv6 Address:
Mobility Control Message DSCP Value: 48
Mobility Keepalive Interval/Count: 10/3
Mobility Group Name: default
Mobility Multicast Ipv4 address: 0.0.0.0
Mobility Multicast Ipv6 address: ::
Mobility MAC Address: 001e.e67e.75ff
Mobility Domain Identifier: 0x34ac
```

第二步：从9800 WLC收集哈希值

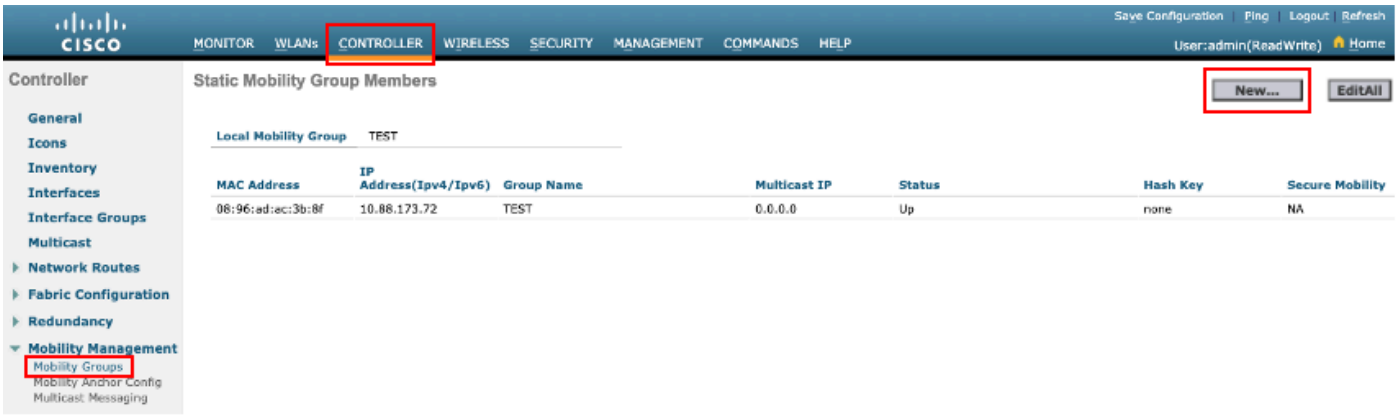
```
# show wireless management trustpoint
Trustpoint Name : Jay-9800_WLC_TP
```

```
Certificate Info : Available
Certificate Type : SSC
Certificate Hash : d7bde0898799dbfeffd4859108727d3372d3a63d
Private key Info : Available
FIPS suitability : Not Applicable
```

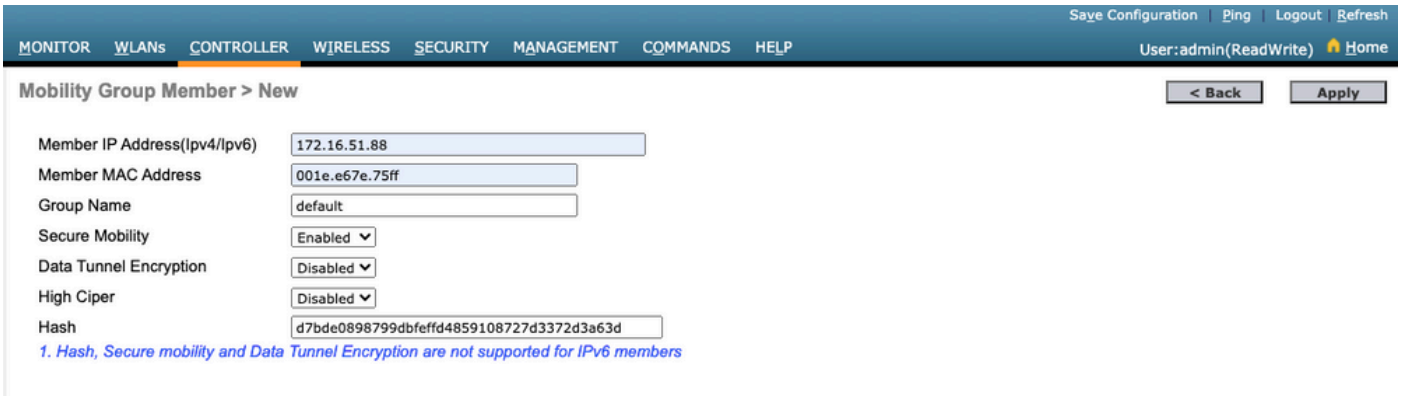
第三步：将9800 WLC信息添加到AireOS WLC中。

通过GUI:

导航至 **CONTROLLER > Mobility Management > Mobility Groups > New.**



输入值，然后单击 **Apply.**



> **注**：仅当9800使用自签名证书（例如C9800-CL）时才需要散列。硬件设备具有SUDI证书，不需要散列（例如9800-40、9800-L等）。

通过CLI:

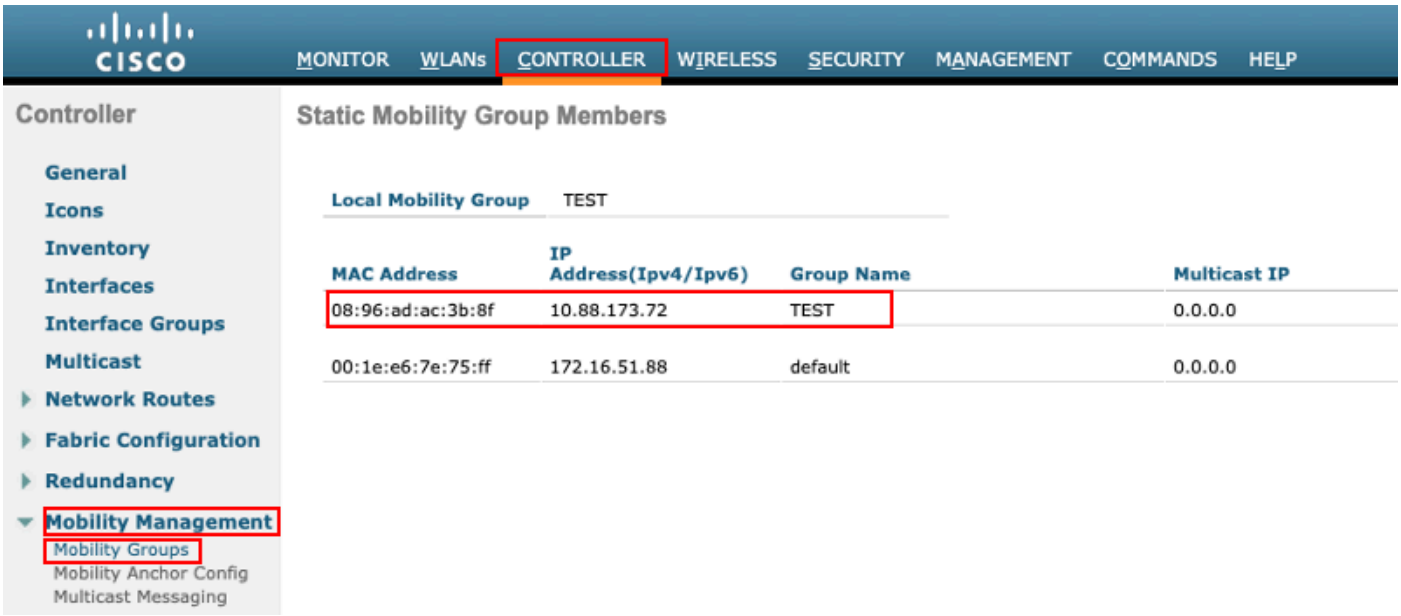
```
>config mobility group member add <9800 mac-address> <9800 WLC-IP> <group-name> encrypt enable
>config mobility group member hash <9800 WLC-IP> <9800 WLC-Hash>
>config mobility group member data-dtls <9800 mac-address> disable
```

9800 WLC配置

步骤1:收集AireOS移动信息。

通过GUI:

登录到AireOS GUI并导航至 **CONTROLLER > Mobility Management > Mobility Groups** 并记录MAC地址、IP地址和组名。



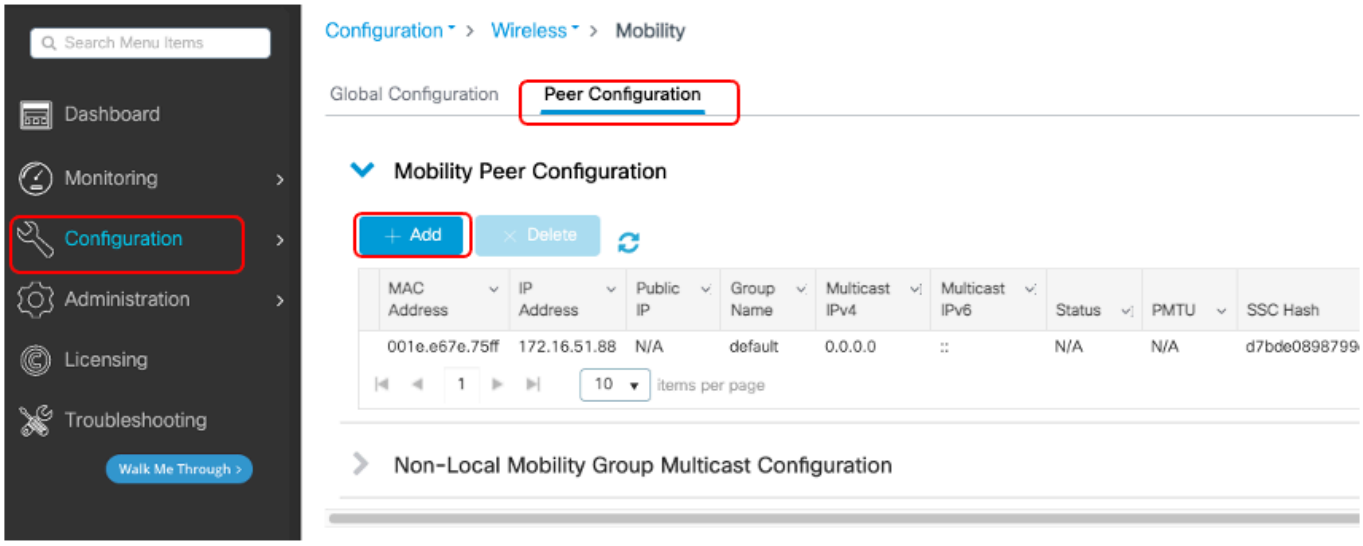
通过CLI:

```
>show mobility summary

Mobility Protocol Port.......................... 16666
Default Mobility Domain......................... TEST
Multicast Mode ................................. Disabled
Mobility Domain ID for 802.11r.................. 0x6ef9
Mobility Keepalive Interval..................... 10
Mobility Keepalive Count........................ 3
Mobility Group Members Configured............... 2
Mobility Control Message DSCP Value............. 48

Controllers configured in the Mobility Group

 MAC Address        IP Address         Group Name                  Multicast IP
Status
08:96:ad:ac:3b:8f   10.88.173.72         TEST                        0.0.0.0
Up
```

第二步：将AireOS WLC信息添加到9800 WLC

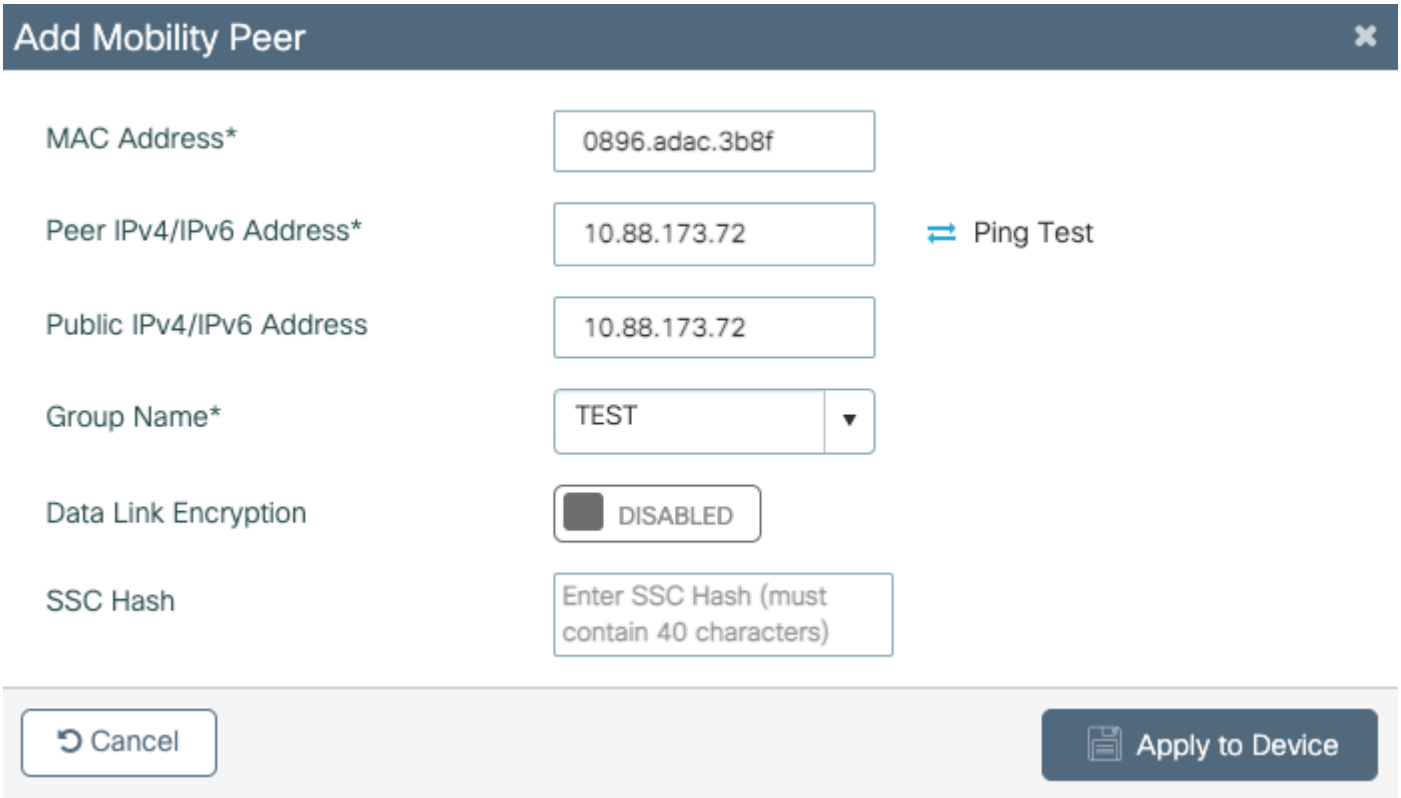
通过GUI:

导航至 **Configuration > Wireless > Mobility > Peer Configuration > Add**



输入AireOS WLC信息。

注：在9800 WLC上，控制平面加密始终启用，这意味着您需要在AireOS端启用安全移动。但是，数据链路加密是可选的。如果在9800端启用它，请在AireOS上启用它，并使用**config mobility group member data-dtls enable**



通过CLI:

```
# config t
# wireless mobility group member mac-address <peer-mac-address> ip <ip-address> group <group-name>
```

验证

使用本部分可确认配置能否正常运行。

AireOS WLC验证

```
>show mobility summary

Mobility Protocol Port.......................... 16666
Default Mobility Domain......................... TEST
Multicast Mode ................................. Disabled
Mobility Domain ID for 802.11r.................. 0x6ef9
Mobility Keepalive Interval..................... 10
Mobility Keepalive Count........................ 3
Mobility Group Members Configured............... 2
Mobility Control Message DSCP Value............. 48

Controllers configured in the Mobility Group
 MAC Address        IP Address                                      Group Name
Multicast IP                                    Status
 00:1e:e6:7e:75:ff  172.16.51.88                                     default
0.0.0.0                                         Up
 08:96:ad:ac:3b:8f  10.88.173.72                                    TEST
0.0.0.0                                         Up
```

Catalyst 9800 WLC验证

```
#show wireless mobility summary
Mobility Summary

Wireless Management VLAN: 2652
Wireless Management IP Address: 172.16.51.88
Mobility Control Message DSCP Value: 48
Mobility Keepalive Interval/Count: 10/3
Mobility Group Name: mb-kcg
Mobility Multicast Ipv4 address: 0.0.0.0
Mobility Multicast Ipv6 address: ::
Mobility MAC Address: 001e.e67e.75ff

Controllers configured in the Mobility Domain:

 IP               Public Ip         Group Name                      Multicast IPv4    Multicast
IPv6                               Status                       PMTU
-------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------
172.16.51.88       N/A               default                         0.0.0.0           ::
N/A                           N/A
10.88.173.72      10.88.173.72        TEST                           0.0.0.0           ::
Up                            1385
```

故障排除

本节提供用于排除配置故障的信息。

要对移动隧道实施进行故障排除，请使用以下命令调试该过程：

AireOS WLC

步骤1:启用移动调试。

```
debug mobility handoff enable
debug mobility error enable
debug mobility dtls error enable
debug mobility dtls event enable
debug mobility pmtu-discovery enable
debug mobility config enable
debug mobility directory enable
```

第二步：复制配置并检验输出

在AirOS WLC上成功创建移动隧道的示例。

```
*capwapPingSocketTask: Feb 07 09:53:38.507: Client initiating connection on 172.16.0.5:16667 <->
172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.507: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.508: Received DTLS packet from mobility peer 172.16.0.21
bytes: 48
*capwapPingSocketTask: Feb 07 09:53:38.508: mm_dtls2_process_data_rcv_msg:1207 rcvBufLen 48
clr_pkt_len 2048 peer ac100015
*capwapPingSocketTask: Feb 07 09:53:38.508: Record    : type=22, epoch=0, seq=0
*capwapPingSocketTask: Feb 07 09:53:38.508:    Hndshk : type=3, len=23 seq=0, frag_off=0,
frag_len=23
*capwapPingSocketTask: Feb 07 09:53:38.508: Handshake in progress for link 172.16.0.5:16667 <->
172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.508: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.508: DTLS consumed packet from mobility peer 172.16.0.21
bytes: 48
!
!<--output-omited-->
!
*capwapPingSocketTask: Feb 07 09:53:38.511: dtls2_cert_verify_callback: Forcing Certificate
validation as success
*capwapPingSocketTask: Feb 07 09:53:38.511: Peer certificate verified.
*capwapPingSocketTask: Feb 07 09:53:38.511: Handshake in progress for link 172.16.0.5:16667 <->
172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.511: Nothing to send on link 172.16.0.5:16667 <->
172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.511: DTLS consumed packet from mobility peer 172.16.0.21
bytes: 503
*capwapPingSocketTask: Feb 07 09:53:38.511: Received DTLS packet from mobility peer 172.16.0.21
bytes: 56
*capwapPingSocketTask: Feb 07 09:53:38.511: mm_dtls2_process_data_rcv_msg:1207 rcvBufLen 56
clr_pkt_len 2048 peer ac100015
*capwapPingSocketTask: Feb 07 09:53:38.511: Record    : type=22, epoch=0, seq=6
*capwapPingSocketTask: Feb 07 09:53:38.511:    Hndshk : type=13, len=6 seq=3, frag_off=0,
frag_len=6
*capwapPingSocketTask: Feb 07 09:53:38.523: Handshake in progress for link 172.16.0.5:16667 <->
172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.524: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.524: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.524: DTLS consumed packet from mobility peer 172.16.0.21
bytes: 56
*capwapPingSocketTask: Feb 07 09:53:38.527: Received DTLS packet from mobility peer 172.16.0.21
bytes: 91
*capwapPingSocketTask: Feb 07 09:53:38.527: mm_dtls2_process_data_rcv_msg:1207 rcvBufLen 91
clr_pkt_len 2048 peer ac100015
```

```
*capwapPingSocketTask: Feb 07 09:53:38.527: Record    : type=20, epoch=0, seq=8
*capwapPingSocketTask: Feb 07 09:53:38.527: Connection established for link 172.16.0.5:16667 <->
172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.527: ciperspec 1
*capwapPingSocketTask: Feb 07 09:53:38.527: Nothing to send on link 172.16.0.5:16667 <->
172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.527: DTLS consumed packet from mobility peer 172.16.0.21
bytes: 91
*mmMobility: Feb 07 09:53:38.527: DTLS Action Result message received
*mmMobility: Feb 07 09:53:38.527:  Key plumb succeeded
*mmMobility: Feb 07 09:53:38.527: mm_dtls2_callback: Connection established with
172.16.0.21:16667
*mmMobility: Feb 07 09:53:38.527: mm_dtls2_db_status_up:895 Connections status up for entry
172.16.0.21:16667
*mmMobility: Feb 07 09:53:38.527: mm_dtls2_callback: DTLS Connection established with
172.16.0.21:16667, Sending update msg to mobility HB
```

Catalyst 9800 WLC

默认情况下，9800控制器持续记录进程信息，无需任何特殊的调试过程。

只需连接到控制器并检索与任何无线组件相关的日志即可进行故障排除。

日志可能跨越数天；这取决于控制器的繁忙程度。

为简化分析，请提取时间范围或最后分钟数的日志（默认时间设置为10分钟），您可以按IP或MAC地址过滤。

步骤1:检查控制器时间上的当前时间，以便您可以跟踪问题发生时的登录时间。

```
# show clock
```

第二步：收集控制器日志，以防Cisco IOS级别出现可能与问题相关的任何信息。

```
# show logging
```

第三步：收集特定地址的不间断通知级别跟踪。您可以使用移动对等IP或MAC进行过滤。

```
# show logging profile wireless filter ipv4 to-file bootflash:ra-AAAA.BBBB.CCCC.txt
```

此命令生成过去10分钟的日志，可以使用此命令调整此时间 **show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt.**

您可以显示会话中的内容，也可以将文件复制到外部TFTP服务器。

```
# more bootflash:always-on-<FILENAME.txt>

or

# copy bootflash:always-on-<FILENAME.txt> tftp://a.b.c.d/path/always-on-<FILENAME.txt>
```

无线电活动跟踪

如果永远在线日志没有提供足够的信息来了解隧道配置期间触发的问题，则可以启用条件调试和捕

获 **Radio Active (RA)** 跟踪，提供更加详细的流程活动。

步骤1:验证是否未启用调试条件。

```
# show debugging
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop


IOSXE Packet Tracing Configs:



Packet Infra debugs:

Ip Address                                              Port
------------------------------------------------------|----------
```

如果发现与要监控的地址无关的任何条件，请将其禁用。

要删除特定地址，请执行以下操作：

```
# no debug platform condition feature wireless { mac <aaaa.bbbb.cccc> | ip <a.b.c.d> }
```

要删除所有条件（建议方式），请执行以下操作：

```
# clear platform condition all
```

第二步：为要监控的地址添加调试条件。

```
# debug platform condition feature wireless ip <a.b.c.d>
```

注：如果要同时监控多个移动对等体，请使用 **debug platform condition feature wireless mac** 命令。

第三步：让9800 WLC启动指定地址活动的监控。

```
# debug platform condition start
```

注意：移动活动的输出不会显示，因为所有内容都在内部缓冲，稍后收集。

第四步：重现要监控的问题或行为。

第五步：停止调试。

```
# debug platform condition stop
```

第六步：收集地址活动的输出。

```
# show logging profile wireless filter ipv4 to-file bootflash:ra-AAAA.BBBB.CCCC.txt
```

此命令生成过去10分钟的日志。可以使用show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt命令调整此时间。

您可以复制 **FILENAME.txt** 或直接在屏幕上显示输出。

将文件复制到外部服务器：

```
# copy bootflash:FILENAME.txt tftp://a.b.c.d/ra-FILENAME.txt
```

显示内容：

```
# more bootflash:ra-FILENAME.txt
```

步骤 7.如果仍然无法找到故障的原因，请收集内部日志级别。

(无需再次调试客户端。使用已内部存储的日志，但收集范围更广的日志)。

```
# show logging profile wireless internal filter ipv4 to-file bootflash:raInternal-
AAAA.BBBB.CCCC.txt
```

您可以复制 **FILENAME.txt** 或直接在屏幕上显示输出。

将文件复制到外部服务器：

```
# copy bootflash:FILENAME.txt tftp://a.b.c.d/ra-FILENAME.txt
```

显示内容：

```
# more bootflash:ra-FILENAME.txt
```

步骤 8删除调试条件。

```
# clear platform condition all
```

注意：在进行故障排除会话后，请始终删除调试条件。

在9800 WLC上成功创建移动隧道的示例。

```
2021/09/28 10:20:50.497612 {mobilityd_R0-0}{1}: [errmsg] [26516]: (info): %MM_NODE_LOG-6-
MEMBER_ADDED: Adding Mobility member (IP: IP: 172.16.55.28: default)
2021/09/28 10:20:52.595483 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_data of XID (0) to (ipv4: 172.16.55.28 )
2021/09/28 10:20:52.595610 {mobilityd_R0-0}{1}: [mm-pmtu] [26516]: (debug): Peer IP:
172.16.55.28 PMTU size is 1385 and calculated additional header length is 148
2021/09/28 10:20:52.595628 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_ctrl_req of XID (80578) to (ipv4: 172.16.55.28 )
2021/09/28 10:20:52.595686 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 keepalive data packet misssed, total missed packet = 1
2021/09/28 10:20:52.595694 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
```

```
172.16.55.28 keepalive ctrl packet misssed, total missed packet = 1
2021/09/28 10:21:02.596500 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_data of XID (0) to (ipv4: 172.16.55.28 )
2021/09/28 10:21:02.596598 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 keepalive data packet misssed, total missed packet = 2
2021/09/28 10:21:02.598898 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
001e.e68c.5dff Received keepalive_data, sub type: 0 of XID (0) from (ipv4: 172.16.55.28 )
2021/09/28 10:21:12.597912 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_data of XID (0) to (ipv4: 172.16.55.28 )
2021/09/28 10:21:12.598009 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 Data link set state to UP (was DOWN)
2021/09/28 10:21:12.598361 {mobilityd_R0-0}{1}: [errmsg] [26516]: (note): %MM_NODE_LOG-5-
KEEP_ALIVE: Mobility Data tunnel to peer IP: 172.16.55.28 changed state to UP

! !<--output-omited--> !

2021/09/28 10:21:22.604098 {mobilityd_R0-0}{1}: [ewlc-infra-evq] [26516]: (debug): DTLS record
type: 22, handshake
2021/09/28 10:21:22.604099 {mobilityd_R0-0}{1}: [ewlc-infra-evq] [26516]: (info): DTLS client
hello
2021/09/28 10:21:22.611477 {mobilityd_R0-0}{1}: [ewlc-infra-evq] [26516]: (debug): DTLS record
type: 22, handshake
2021/09/28 10:21:22.611555 {mobilityd_R0-0}{1}: [ewlc-infra-evq] [26516]: (debug): DTLS record
type: 22, handshake
2021/09/28 10:21:22.611608 {mobilityd_R0-0}{1}: [ewlc-infra-evq] [26516]: (debug): DTLS record
type: 22, handshake
2021/09/28 10:21:22.611679 {mobilityd_R0-0}{1}: [ewlc-infra-evq] [26516]: (debug): DTLS record
type: 22, handshake
2021/09/28 10:21:22.611933 {mobilityd_R0-0}{1}: [mm-dtls] [26516]: (note): Peer IP: 172.16.55.28
Port: 16666, Local IP: 172.16.51.88 Port: 16666 DTLS_SSC_HASH_VERIFY_CB: SSC hash validation
success
2021/09/28 10:21:22.612163 {mobilityd_R0-0}{1}: [ewlc-dtls-sessmgr] [26516]: (info): Remote
Host: 172.16.55.28[16666] Completed cert verification, status:CERT_VALIDATE_SUCCESS

! !<--output-omited--> !

2021/09/28 10:21:52.603200 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 Control link set state to UP (was DOWN)
2021/09/28 10:21:52.604109 {mobilityd_R0-0}{1}: [errmsg] [26516]: (note): %MM_NODE_LOG-5-
KEEP_ALIVE: Mobility Control tunnel to peer IP: 172.16.55.28 changed state to UP
```

嵌入式数据包捕获

大多数情况下，它非常有助于检查WLC之间交换的数据包。它对于过滤捕获特别有用， **Access Control Lists (ACLs)** 以限制捕获的流量。

这是CLI上嵌入式捕获的配置模板。

步骤1:创建过滤器ACL:

```
conf t
ip access-list extended <ACL_NAME>
10 permit ip host <WLC_IP_ADDR> host <PEER_WLC_IP_ADDR>
20 permit ip host <PEER_WLC_IP_ADDR>host <WLC_IP_ADDR>
end
```

第二步：定义捕获参数：

```
monitor capture <CAPTURE_NAME> access-list <ACL_NAME> buffer size 10 control-plane both
```

```
interface <INTERFACE_NAME> both limit duration 300
```

注意：为INTERFACE_NAME参数选择管理接口

第三步：开始捕获：

```
monitor capture <CAPTURE_NAME> start
```

第四步：停止捕获：

```
monitor capture <CAPTURE_NAME> stop
```

第五步：在GUI上导航到**故障排除>数据包捕获**以收集数据包捕获文件。

常见故障排除场景

下一个示例包括在9800 WLC之间形成的隧道。

由于连接问题导致控制和数据路径关闭

enable **AIways-On-Logs** 和 **Embedded packet captures** 提供故障排除的其他信息：

```
2021/09/28 09:54:22.490625 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_ctrl_req of XID (80552) to (ipv4: 172.16.55.28 )
2021/09/28 09:54:22.490652 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 keepalive data packet misssed, total missed packet = 29
2021/09/28 09:54:22.490657 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 keepalive ctrl packet misssed, total missed packet = 10
2021/09/28 09:54:32.491952 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_data of XID (0) to (ipv4: 172.16.55.28 )
2021/09/28 09:54:32.492127 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 keepalive data packet misssed, total missed packet = 30
```

数据包捕获对于确认行为非常有用。

```
 90 2021-09-28 12:33:52.924939 172.16.51.88   172.16.55.28   116 Mobi-Control - PingReq[Malformed Packet]
 91 2021-09-28 12:34:02.925946 172.16.51.88   172.16.55.28   172 Mobi-Data Keep-Alive - Mobility CAPWAP Ping Request
 92 2021-09-28 12:34:12.925946 172.16.51.88   172.16.55.28   172 Mobi-Data Keep-Alive - Mobility CAPWAP Ping Request
 93 2021-09-28 12:34:22.927945 172.16.51.88   172.16.55.28   172 Mobi-Data Keep-Alive - Mobility CAPWAP Ping Request
 94 2021-09-28 12:34:22.927945 172.16.51.88   172.16.55.28   116 Mobi-Control - PingReq[Malformed Packet]
 95 2021-09-28 12:34:32.927945 172.16.51.88   172.16.55.28   172 Mobi-Data Keep-Alive - Mobility CAPWAP Ping Request
 96 2021-09-28 12:34:42.929944 172.16.51.88   172.16.55.28   172 Mobi-Data Keep-Alive - Mobility CAPWAP Ping Request
 97 2021-09-28 12:34:52.930951 172.16.51.88   172.16.55.28   172 Mobi-Data Keep-Alive - Mobility CAPWAP Ping Request
```

请注意，debug和WLC均显示没有响应控制或数据ping。常见情况显示允许IP连接，但不允许端口16666或16667通过网络通信。

WLC之间的配置不匹配

在本例中，我们确认了WLC之间所有端口的连通性，但继续发现keepalive miss。

enable **AIways-On-Logs** 和 **Embedded packet captures** 提供故障排除的其他信息：

```
2021/09/28 11:34:22.927477 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_data of XID (0) to (ipv4: 172.16.55.28 )
2021/09/28 11:34:22.928025 {mobilityd_R0-0}{1}: [mm-pmtu] [26516]: (debug): Peer IP:
172.16.55.28 PMTU size is 1385 and calculated additional header length is 148
2021/09/28 11:34:22.928043 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_ctrl_req of XID (80704) to (ipv4: 172.16.55.28 )
2021/09/28 11:34:22.928077 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 keepalive data packet misssed, total missed packet = 8
2021/09/28 11:34:22.928083 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 keepalive ctrl packet misssed, total missed packet = 3
```

对等172.16.55.28的内部日志可帮助我们确认配置不匹配

```
2021/09/28 17:33:22.963 {mobilityd_R0-0}{1}: [mm-keepalive] [27081]: (ERR): Peer IP:
172.16.51.88 Failed to validate endpoint: Invalid argument
2021/09/28 17:33:22.963 {mobilityd_R0-0}{1}: [errmsg] [27081]: (ERR): %MM_NODE_LOG-3-
PING_DROPPED: Drop data ping from IP: 172.16.51.88. Failed to validate endpoint
```

常见配置不匹配包括：组名称不正确，上的不匹配 **Data Link Encryption** 和错误的移动mac地址。

组不匹配日志：

```
2021/09/28 17:33:22.963 {mobilityd_R0-0}{1}: [errmsg] [27081]: (ERR): %MM_INFRA_LOG-3-
MSG_PROC_FAILED_GROUP_NAME_HASH: Pkt group name hash: 82FE070E6E9A37A543CEBED96DB0388F Peer
group name hash: 3018E2A00F10176849AC824E0190AC86 Failed to validate endpoint. reason: Group
name hash mismatch.
```

MAC地址不匹配日志：

```
2021/09/28 19:09:33.455 {mobilityd_R0-0}{1}: [errmsg] [27081]: (ERR): %MM_INFRA_LOG-3-
MSG_PROC_FAILED_MAC_ADDR: Pkt MAC: 001e.e67e.75fa Peer MAC: 001e.e67e.75ff Failed to validate
endpoint. reason: MAC address mismatch.
```

DTLS握手问题

此类问题与WLC之间的DTLS隧道建立有关。这可能是Data path is UP but Control path remained的情况 **DOWN**.

enable **Always-On-Logs** 和 **Embedded packet captures** 提供故障排除的其他信息：

```
2021/09/28 19:30:23.534 {mobilityd_R0-0}{1}: [mm-msg] [27081]: (ERR): Peer IP: 172.16.51.88
Port: 16666 DTLS_MSG: DTLS message process failed. Error: Invalid argument
2021/09/28 19:30:23.534 {mobilityd_R0-0}{1}: [errmsg] [27081]: (warn): %MM_NODE_LOG-4-
DTLS_HANDSHAKE_FAIL: Mobility DTLS Ctrl handshake failed for 172.16.51.88 HB is down, need to
re-initiate DTLS handshake
2021/09/28 19:30:23.534 {mobilityd_R0-0}{1}: [ewlc-capwapmsg-sess] [27081]: (ERR): Source
IP:172.16.51.88[16666], DTLS message process failed. length:52
```

使用 **show wireless management trustpoint** 和 **show crypto pki trustpoints commands** 验证证书信息。

HA SSO场景

如果您在高可用性SSO对中有控制器，需要了解一个重要问题。默认情况下未配置移动MAC地址，如果发生故障转移，可能导致移动隧道关闭。

show wireless mobility summary为您提供了当前使用的移动MAC，但不需要进行配置。检查配置是

否使用**show run**配置了移动MAC **| i移动性**

如果在运行配置中未配置移动mac，则它在故障切换至备用WLC后会更改，这会导致移动隧道失败。

简单的解决方案是导航到**Configuration > Wireless > Mobility** Web UI页面并点击**apply**。这会将当前移动MAC保存到配置。然后，在保留故障切换和移动隧道时，MAC保持不变。

如果您通过命令行执行移动配置并忘记配置移动MAC地址，则主要会发生此问题。当您应用设置时，Web UI会自动保存移动MAC地址。

相关信息

- 在Catalyst 9800上配置WLAN锚点移动功能
- 技术支持和文档 - Cisco Systems

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）。