Catalyst 9800ワイヤレスLANコントローラ(WLC)でのモビリティトポロジの設定

内容

概要

このドキュメントでは、Catalyst 9800ワイヤレスLANコントローラ(WLC)とAireOS WLC間のト

ポロジをカバーするモビリティ設定シナリオについて説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- ワイヤレスコントローラへのCLIまたはGUIアクセス。

使用するコンポーネント

- **AireOS WLC** バージョン8.10 MR1以降。また、次のコマンドも使用できます。 **Inter Release Controller Mobility (IRCM)** 特別な8.5イメージ
- 9800 WLC、Cisco IOS® XE v17.3.4

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

設定

ネットワーク図

ガイドラインと制限事項

1. **Mobility Group** 9800のデフォルトの名前は「default」です。

注：
1) WLCが異なるサブネットにある場合は、ポートUDP 16666と16667がそれらの間で開いていることを確認します。
2)両方の9800 WLCで同じバージョンを実行して、ローミングするクライアントがレイヤ3ローミングとゲストアンカーの両方のシナリオで一貫したエクスペリエンスを得られるようにすることをお勧めします。

2台のCatalyst 9800 WLC間のモビリティトンネル

この基本的な例では、2台の9800コントローラ間でモビリティを設定する方法について説明します。これは一般に、ゲストアクセス（アンカー）に使用されるか、クライアントがコントローラ間をローミングしてクライアントIDを保持できるようにするために使用されます。

C9800でモビリティを設定する場合、最初に選択するのはモビリティグループ名です。デフォルトでは、事前に入力されたモビリティグループ名が使用されますが、必要な値にカスタマイズできます。

高速レイヤ2ローミングが次のような場合は、コントローラ間で同じモビリティグループ名を設定する必要があります **Fast Transition (FT)** または **Cisco Centralized Key Management (CCKM)** は使用中です。

デフォルトでは、シャーシの基本イーサネットMACアドレスは次のようになります。 **show version** モビリティMACアドレスのGUIに反映されます。

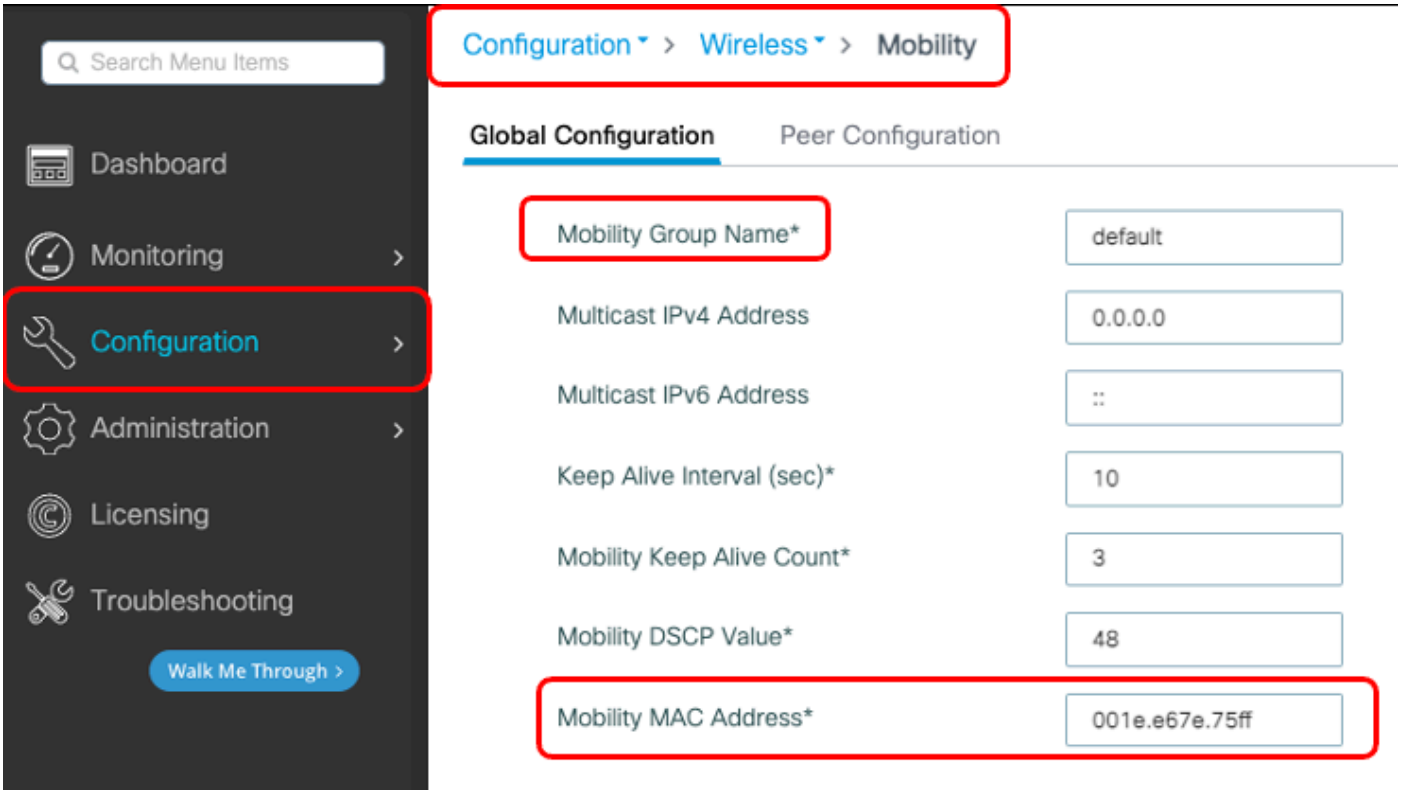
CLIでは、デフォルトでモビリティMACは0000.0000.000です(**show run all | inc mobility mac-address**

9800sが **High Availability (HA) Stateful Switchover (SSO):**

設定をデフォルトのままにし、シャーシのMACアドレスを使用してモビリティトンネルを形成する場合、フェールオーバーが発生すると、アクティブシャーシとモビリティトンネルで障害が発生します。

したがって、モビリティMACアドレスはC9800 HAペア用に設定する必要があります。

ステップ1:GUIで、 **Configuration > Wireless > Mobility > Global Configuration.**



CLIを使用する場合：

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

ステップ 1：両方の9800 WLCのモビリティ設定を収集します。

両方の9800 WLCで、 **Configuration > Wireless > Mobility > Global Configuration** そしてIPv6アドレスの **Mobility Group Name** と **Mobility MAC Address.**

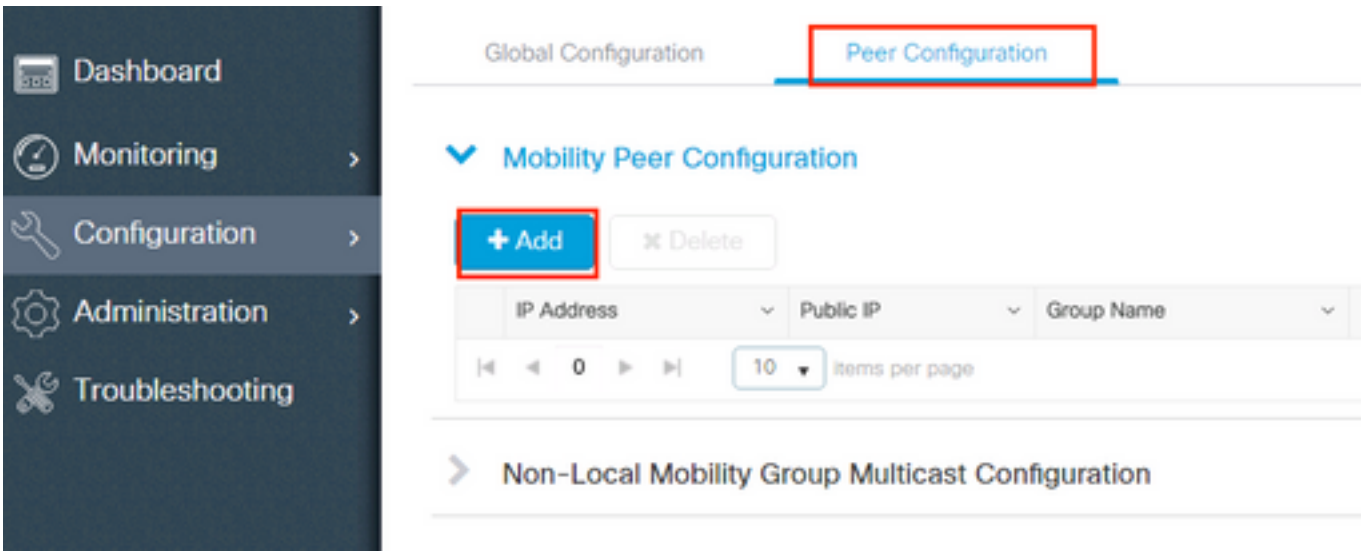
CLIを使用する場合：

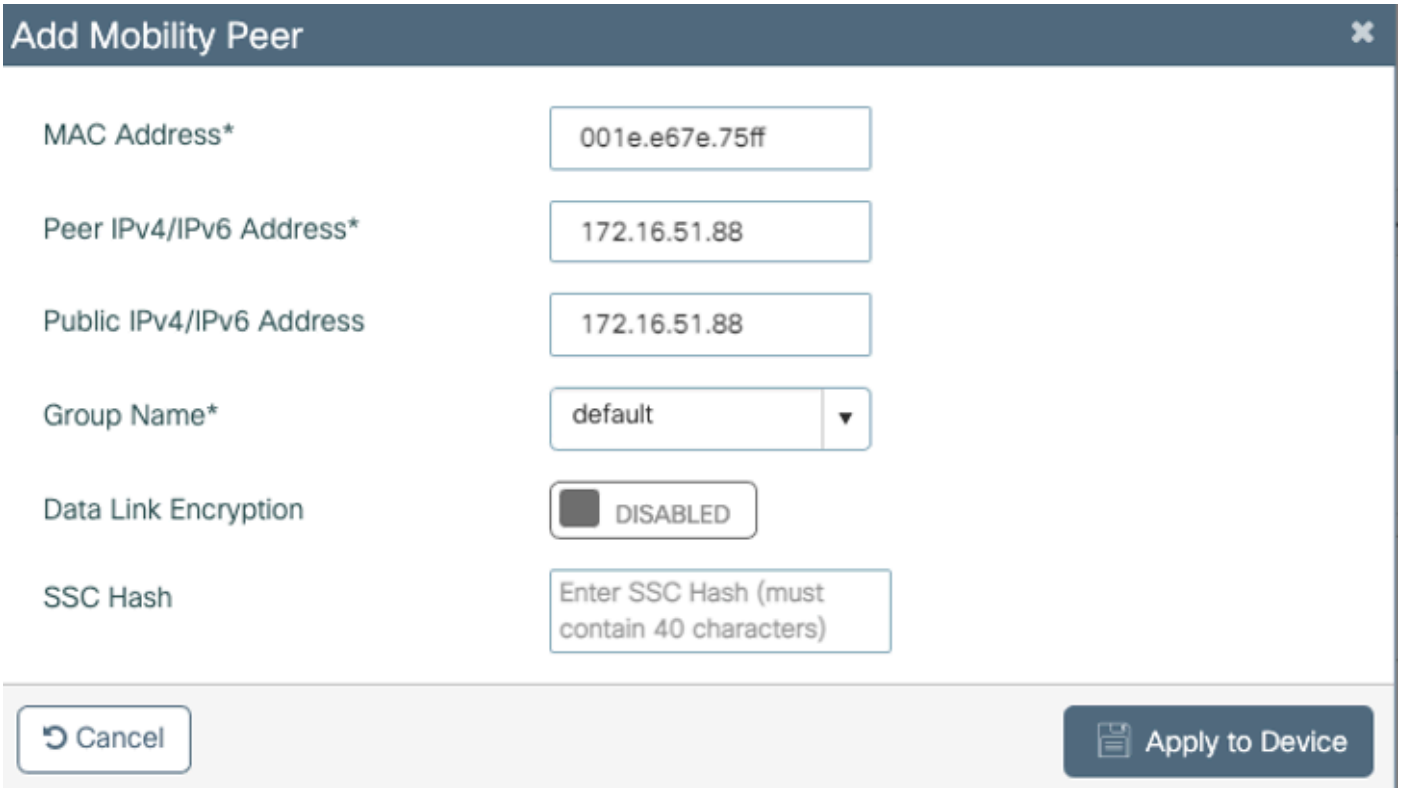
```
#show wireless mobility summary

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
```

ステップ 2：ピア設定の追加

移動先 **Configuration > Wireless > Mobility > Peer Configuration** ピアコントローラ情報を入力します。両方の9800 WLCに対して同じ操作を行います。

GUIを使用する場合：





CLIを使用する場合：

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

注：オプションで、データリンク暗号化を有効にできます。

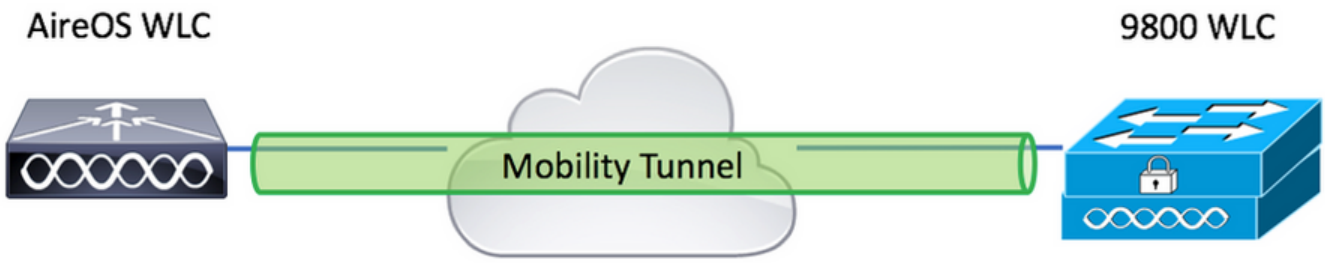
AireOS WLCと9800-CLコントローラ間のモビリティトンネル

このシナリオは、次の場合には正常です **brownfield** ネットワークをAireOSコントローラによって制御されるアクセスポイント(AP)のエリアと9800によって制御されるアクセスポイント(AP)のエリアに分割する導入またはコントローラの移行中。

APは物理エリアまたはRFエリアごとにコントローラ間に分散し、クライアントがコントローラ間を移動するときにだけローミングするようにすることをお勧めします。

避ける **salt and pepper** 導入.オプションで、このモビリティトポロジは次の用途にも使用できます **guest anchor** ここで、9800は外部コントローラとして動作し、AireOSはアンカーコントローラとして動作します。

ネットワーク図



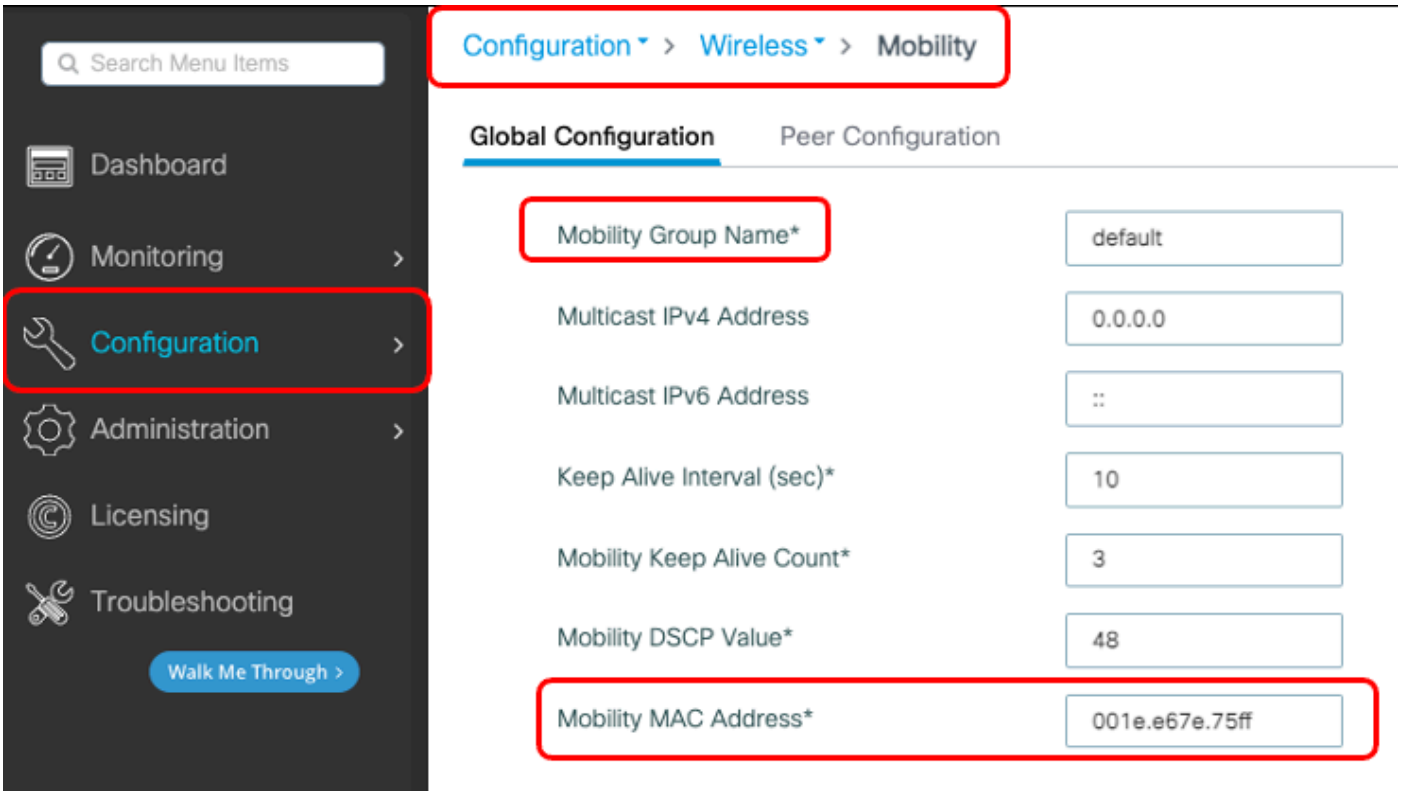
AireOS WLCの設定

9800コントローラが **High Availability**モビリティMACアドレスが設定されていることを確認します。

ステップ 1：9800 WLCモビリティ情報を収集します。

GUIを使用する場合：

移動先 **Configuration > Wireless > Mobility > Global Configuration** そしてIPv6アドレスの **Mobility Group Name** と **Mobility MAC Address.**



CLIを使用する場合：

```
#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
```

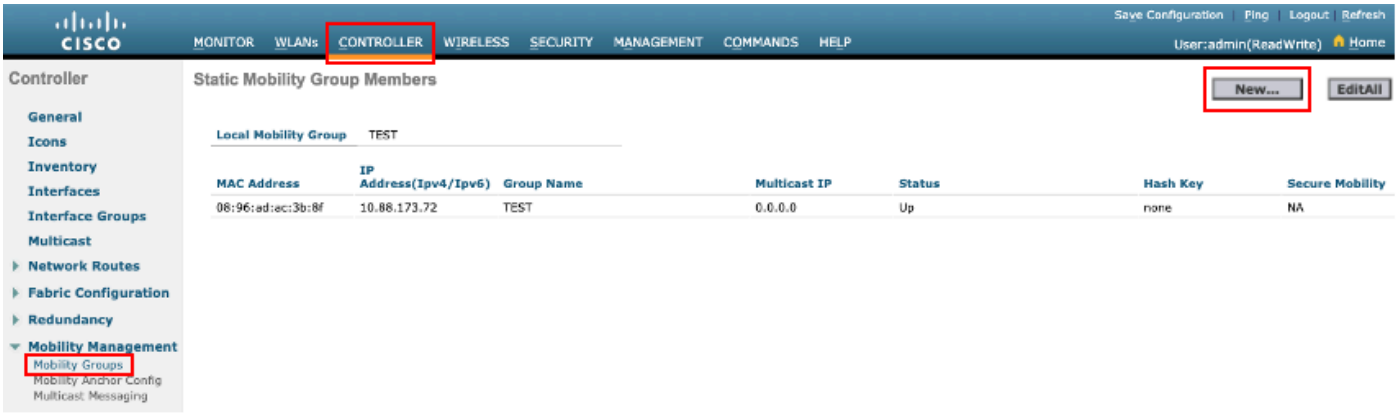
ステップ 2：9800 WLCからハッシュ値を収集します

```
# 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
```

ステップ 3：AireOS WLCに9800 WLC情報を追加します。

GUIを使用する場合：

移動先 **CONTROLLER > Mobility Management > Mobility Groups > New.**



値を入力し、 **Apply.**

> **注**：ハッシュが必要になるのは、9800がC9800-CLなどの自己署名証明書を使用する場合だけです。ハードウェアアプライアンスにはSUDI証明書があり、ハッシュは必要ありません（9800-40、9800-Lなど）。

CLIを使用する場合：

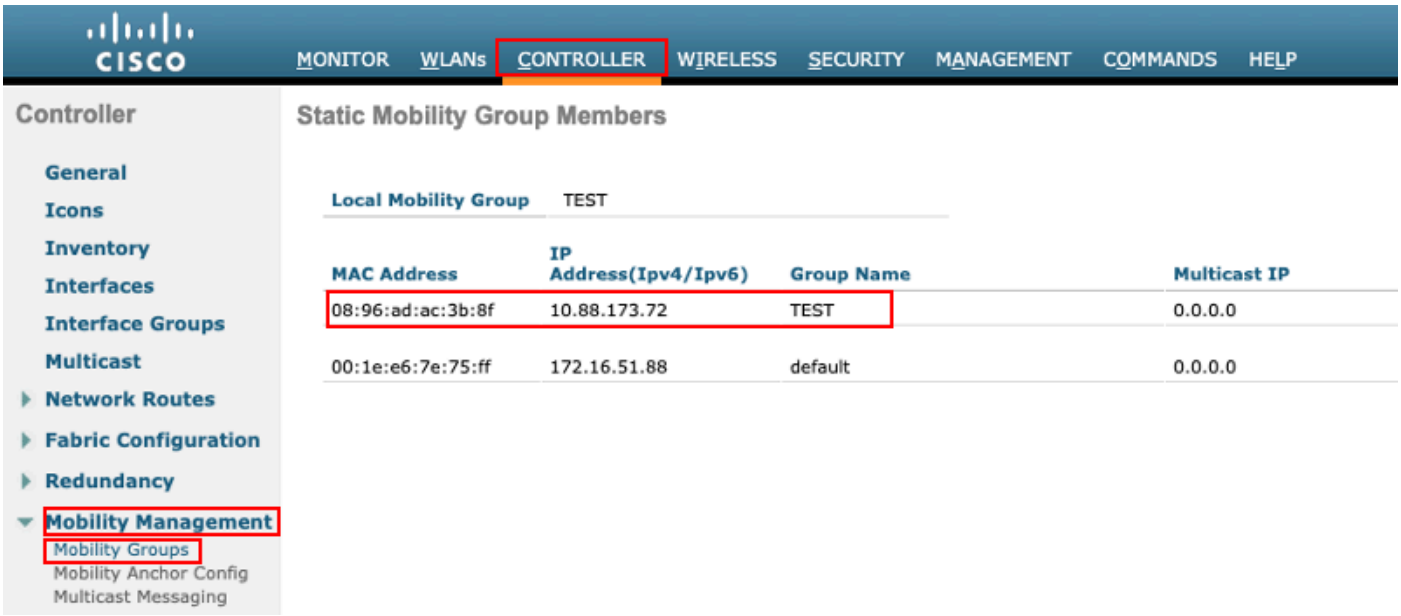
```
>config mobility group member add <9800 mac-address> <9800 WLC-IP> <group-name> encrypt enable
>config mobility group member hash <9800 WLC-IP> <9800 WLC-Hash>
>config mobility group member data-dtls <9800 mac-address> disable
```

9800 WLCの設定

ステップ 1：AireOSモビリティ情報を収集します。

GUIを使用する場合：

AireOS GUIにログインし、 **CONTROLLER > Mobility Management > Mobility Groups** MACアドレス、IPアドレス、およびグループ名を書き留めます。



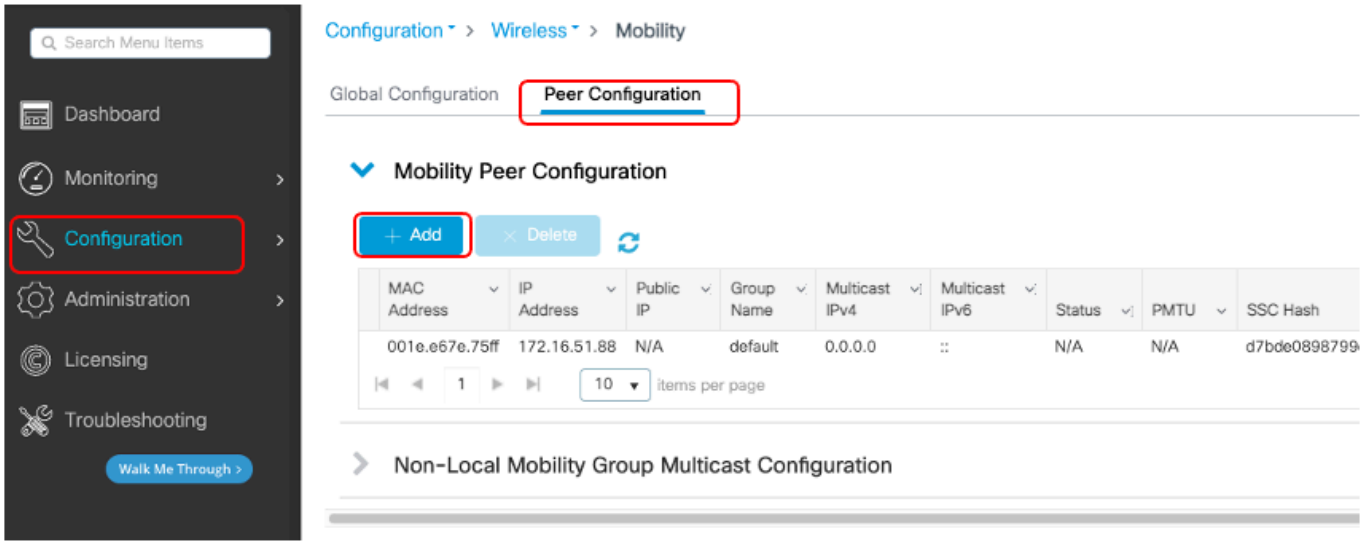
CLIを使用する場合：

```
>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
```

ステップ 2：AireOS WLC情報を9800 WLCに追加する

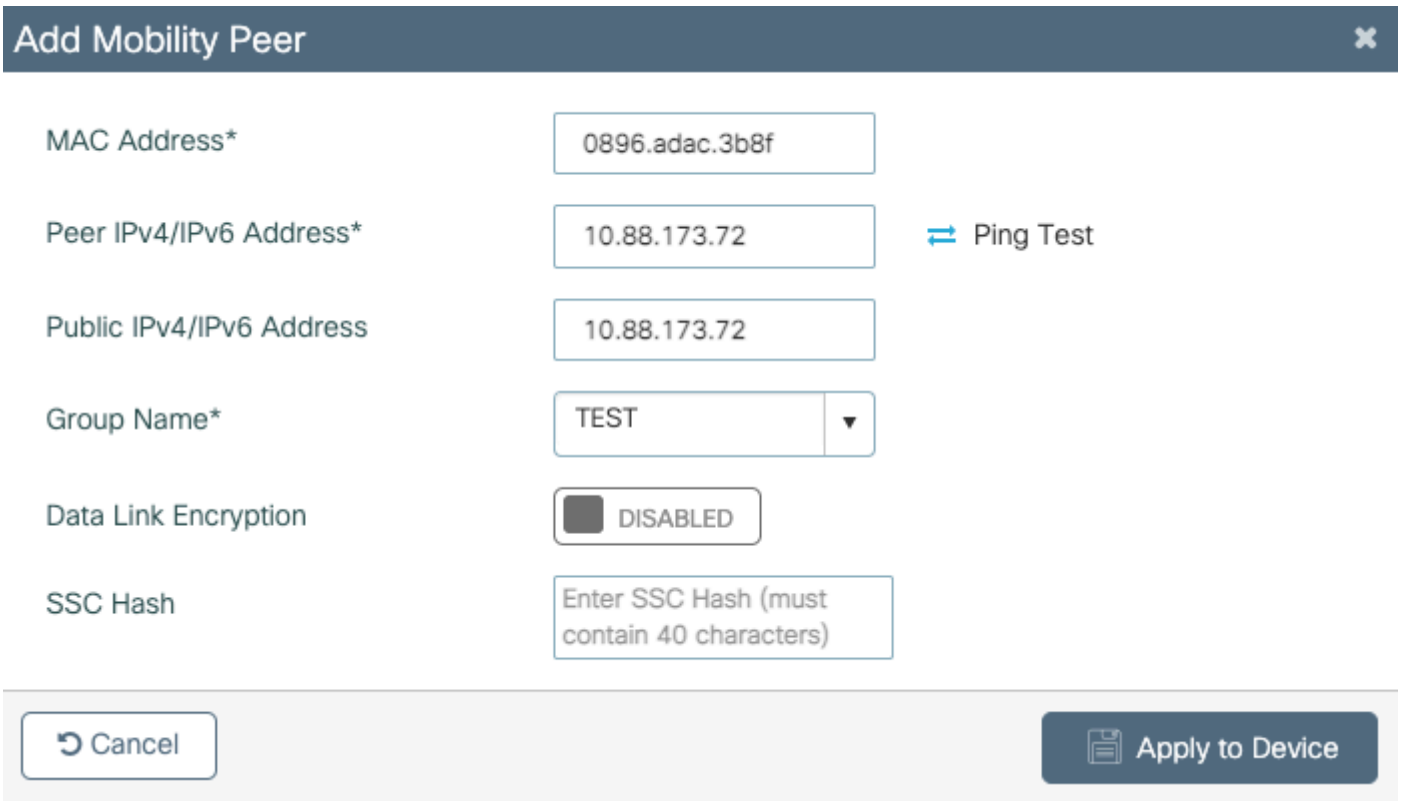
GUIを使用する場合：

移動先 **Configuration > Wireless > Mobility > Peer Configuration > Add**



AireOS WLC情報を入力します。

> **注**:9800 WLCでは、コントロールプレーン暗号化は常に有効になっています。つまり、AireOS側でセキュアモビリティを有効にする必要があります。ただし、データリンク暗号化はオプションです。9800側で有効にする場合は、AireOSで**config mobility group member data-dtls enable**を使用して有効にします。



CLIを使用する場合：

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

確認

ここでは、設定が正常に機能しているかどうかを確認します。

AireOS WLCの確認

```
>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の確認

```
#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
```

トラブルシュート

このセクションでは、設定のトラブルシューティングに使用する情報を提供します。

モビリティトンネルの実装をトラブルシューティングするには、次のコマンドを使用してプロセスをデバッグします。

AireOS WLC

ステップ 1：モビリティデバッグを有効にします。

```
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
```

ステップ 2：設定を再現し、出力を確認します

AirOS WLCで正常にモビリティトンネルが作成された例。

```
*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

デフォルトでは、9800コントローラは特別なデバッグ手順を必要とせずにプロセス情報を継続的にログします。

コントローラに接続し、トラブルシューティングの目的でワイヤレスコンポーネントに関連付けられたログを取得するだけです。

ログは数日にわたることがありますが、これはコントローラのビジー状態によって異なります。

分析を簡素化するには、時間範囲または最後の分数（デフォルトの時間は10分に設定されています）のログを取得し、IPアドレスまたはMACアドレスでフィルタリングできます。

ステップ 1：問題が発生した時点までのログを追跡できるように、コントローラ時刻の現在の状態を確認します。

```
# show clock
```

ステップ 2：問題に関連する可能性のある情報がCisco IOSレベルで存在する場合に備えて、コントローラログを収集します。

```
# show logging
```

ステップ 3：特定のアドレスのAlways-on Notice Level(AVP)トレースを収集します。モビリティピアのIPまたはMACを使用してフィルタリングできます。

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

このコマンドは過去10分間のログを生成します。この時間はコマンドで調整できます **show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt**.

セッションの内容を表示するか、ファイルを外部TFTPサーバにコピーできます。

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

```
or

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

無線アクティブトレース

常時オンのログに、トンネルの設定中に問題を引き起こした原因を知るための十分な情報が記録されていない場合は、条件付きデバッグを有効にしてキャプチャできます **Radio Active (RA)** より詳細なプロセスアクティビティを提供するトレース。

ステップ 1：デバッグ条件が有効になっていないことを確認します。

```
# show debugging
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop


IOSXE Packet Tracing Configs:




Packet Infra debugs:

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

監視するアドレスに関連しない条件が表示された場合は、その条件を無効にします。

特定のアドレスを削除するには、次の手順に従います。

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

すべての条件を削除するには、次の手順に従います（推奨）。

```
# clear platform condition all
```

ステップ 2：監視するアドレスのデバッグ条件を追加します。

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

> **注**：複数のモビリティピアを同時にモニタする場合は、 **debug platform condition feature wireless mac** コマンドをMACアドレスごとに発行します。

ステップ 3：9800 WLCを使用して、指定されたアドレスアクティビティのモニタを開始します。

```
# debug platform condition start
```

> **注**：モビリティアクティビティの出力は、すべてが後で収集できるように内部でバッファされるため表示されません。

ステップ 4：監視する問題または動作を再現します。

ステップ 5：デバッグを停止します。

```
# debug platform condition stop
```

手順 6：アドレスアクティビティの出力を収集します。

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

このコマンドは、過去10分間のログを生成します。この時間は、show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txtコマンドを使用して調整できます。

次のいずれかをコピーできます。 **FILENAME.txt** 外部サーバに送信するか、出力を画面に直接表示します。

ファイルを外部サーバーにコピーします。

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

内容を表示します。

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

手順 7：それでも障害の原因を特定できない場合は、ログの内部レベルを収集します。

(クライアントを再度デバッグする必要はありません。すでに内部に保存されているログを使用しますが、より広範囲のログを収集します)。

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

次のいずれかをコピーできます。 **FILENAME.txt** 外部サーバに送信するか、出力を画面に直接表示します。

ファイルを外部サーバーにコピーします。

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

内容を表示します。

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

ステップ 8：デバッグ条件を削除します。

```
# clear platform condition all
```

注：トラブルシューティングセッションの後は、必ずデバッグ条件を削除してください。

9800 WLCで正常にモビリティトンネルが作成された例。

```
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
```

Embedded Packet Capture

ほとんどの場合、WLC間で交換されるパケットを確認するのに非常に便利です。キャプチャをフィルタ処理する際に特に便利なのは、 **Access Control Lists (ACLs)** キャプチャされたトラフィックを制限します。

これは、CLIに組み込まれたキャプチャ用の設定テンプレートです。

ステップ 1：フィルタACLを作成します。

```
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
```

ステップ 2：キャプチャパラメータを定義します。

```
monitor capture <CAPTURE_NAME> access-list <ACL_NAME> buffer size 10 control-plane both
interface <INTERFACE_NAME> both limit duration 300
```

注：INTERFACE_NAMEパラメータに管理インターフェイスを選択します。

ステップ 3：キャプチャを開始します。

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

ステップ 4：キャプチャを停止します。

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

ステップ 5：GUIで[Troubleshooting] > [Packet Capture] に移動し、パケットキャプチャファイルを収集します。

一般的なトラブルシューティングシナリオ

次の例は、9800 WLC間で形成されるトンネルで構成されています。

接続の問題による制御とデータパスのダウン

Enable **Always-On-Logs** と **Embedded packet captures** トラブルシューティングに必要な追加情報を提供するには、次の手順を実行します。

```
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
```

パケットキャプチャは動作の確認に役立ちます。

```
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
```

デバッグとWLCの両方で、コントロールまたはデータのpingに対する応答がないことが示されます。一般的なシナリオでは、IP接続は許可されるが、ポート16666または16667はネットワーク経由での通信が許可されないことが示されています。

WLC間の設定の不一致

この例では、WLC間のすべてのポートの接続を確認しましたが、キープアライブが失われていることに引き続き気付きます。

Enable **Always-On-Logs** と **Embedded packet captures** トラブルシューティングに必要な追加情報を提供するには、次の手順を実行します。

```
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
```

ピア172.16.55.28の内部ログは、設定の不一致の確認に役立ちます

```
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
```

一般的な設定の不一致には、次のものがあります。グループ名が正しくない、不一致がオンになっている **Data Link Encryption** および不正なモビリティMACアドレス。

グループの不一致ログ：

```
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アドレスのミスマッチログ：

```
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ハンドシェイクの問題

この種の問題は、WLC間のDTLSトンネルの確立に関連しています。データパスがUPであっても制御パスが残っている場合があります **DOWN**.

Enable **Always-On-Logs** と **Embedded packet captures** トラブルシューティングに必要な追加情報を提供するには、次の手順を実行します。

```
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
```

利用 **show wireless management trustpoint** と **show crypto pki trustpoints commands** 証明書の情報を確認します。

HA SSOのシナリオ

ハイアベイラビリティSSOペアのコントローラがある場合、知っておくべき重要なキャッチがあります。モビリティMACアドレスはデフォルトでは設定されておらず、フェールオーバーが発生するとモビリティトンネルがダウンする可能性があります。

show wireless mobility summaryは、現在使用されているモビリティMACを示しますが、必ずしも設定されているわけではありません。設定に**show run**で設定されたモビリティMACがあるかどうかを確認します。 **| i mobility**

モビリティMACが実行コンフィギュレーションで設定されていない場合は、スタンバイWLCへのフェールオーバー時に変更されるため、モビリティトンネルが失敗します。

簡単な解決策は、[Configuration] > [Wireless] > [Mobility] Web UIページに移動し、[apply] をクリックすることです。これにより、現在のモビリティMACが設定に保存されます。その後、MACはフェールオーバー時に同じままになり、モビリティトンネルは保持されます。

この問題は主に、コマンドラインでモビリティ設定を行い、モビリティMACアドレスの設定を忘れた場合に発生します。設定を適用すると、Web UIは自動的にモビリティMACアドレスを保存します。

関連情報

- Catalyst 9800でのWLANアンカーモビリティ機能の設定
- テクニカル サポートとドキュメント – Cisco Systems

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。