Konfiguration von Mobilitätstopologien auf Catalyst 9800 Wireless LAN Controllern (WLCs)

Inhalt

Einleitung

In diesem Dokument werden Szenarien für die Mobilitätskonfiguration beschrieben, die

Topologien zwischen Catalyst 9800 Wireless LAN Controllern (WLCs) und AireOS WLCs abdecken.

Voraussetzungen

Anforderungen

Cisco empfiehlt, sich mit folgenden Themen vertraut zu machen:

- Zugriff auf die Wireless Controller über die Kommandozeile oder die Benutzeroberfläche

Verwendete Komponenten

- **AireOS WLC** Version 8.10 MR1 oder höher. Sie können auch **Inter Release Controller Mobility (IRCM)** Spezial 8.5 Bilder
- 9800 WLC, Cisco IOS® XE v17.3.4

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle verstehen.

Konfigurieren

Netzwerkdiagramm

Richtlinien und Einschränkungen

1. **Mobility Group** auf 9800 aus dem Feld heraus ist "default".

Anmerkung:
1) Befinden sich die WLCs in unterschiedlichen Subnetzen, stellen Sie sicher, dass die Ports UDP 16666 und 16667 zwischen den Subnetzen offen sind.
2) Es wird empfohlen, dass beide 9800 WLCs die gleiche Version ausführen, sodass für die Clients, die roamen, eine konsistente Umgebung sowohl im Layer-3-Roam- als auch im Gast-Anker-Szenario zur Verfügung steht.

Mobility-Tunnel zwischen zwei Catalyst 9800 WLCs

In diesem einfachen Beispiel wird die Einrichtung von Mobilitätsfunktionen für zwei 9800-Controller beschrieben. Dies wird häufig für den Gastzugriff (Auslöser) oder für Clients zum Roamen über Controller verwendet, um die Client-Identität beizubehalten.

Wenn Sie die Mobilität auf dem C9800 konfigurieren, wählen Sie zunächst den Namen der Mobilitätsgruppe aus. Der vorab ausgefüllte Mobilitätsgruppenname ist standardmäßig enthalten, kann jedoch auf einen gewünschten Wert angepasst werden.

Sie müssen denselben Mobilitätsgruppennamen für alle Controller konfigurieren, wenn ein schneller Layer-2-Roaming-Vorgang wie **Fast Transition (FT)** Oder **Cisco Centralized Key Management (CCKM)** wird verwendet.

Standardmäßig wird die Ethernet-MAC-Adresse des Chassis wie in **show version** wird auf der Benutzeroberfläche für die MAC-Adresse der Mobilität angezeigt.

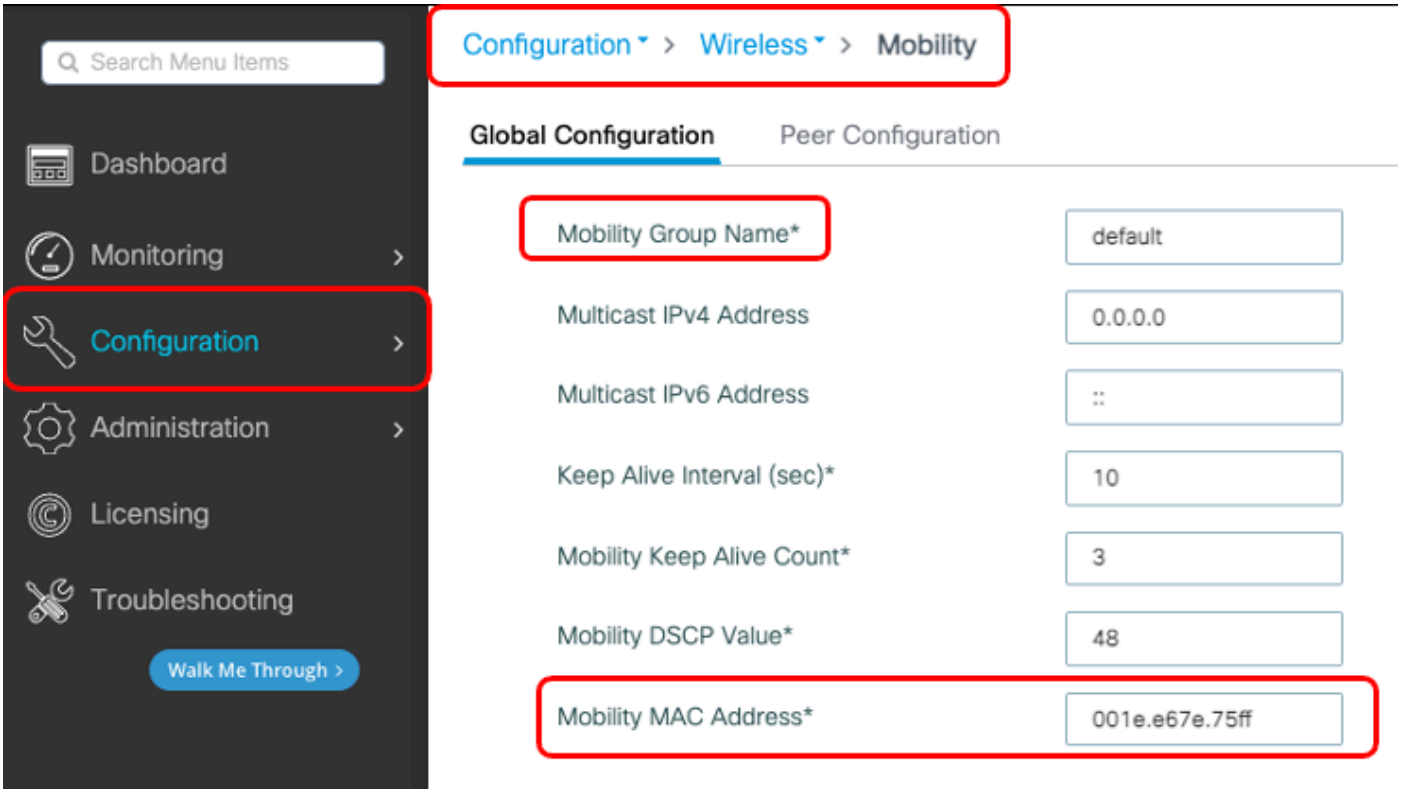
In CLI lautet die Mobilitäts-MAC standardmäßig 0000.0000.000, wie in **show run all | inc mobility mac-address**

In Fällen, in denen 9800s gekoppelt sind **High Availability (HA) Stateful Switchover (SSO):**

Wenn die Konfiguration standardmäßig beibehalten wird und die MAC-Adresse des Gehäuses zur Bildung des Mobilitätstunnels verwendet wird, schlagen das aktive Gehäuse und der Mobilitätstunnel fehl, wenn ein Failover stattfindet.

Aus diesem Grund muss eine Mobility-MAC-Adresse für das C9800 HA-Paar konfiguriert werden.

Schritt 1: Navigieren Sie in der GUI zu **Configuration > Wireless > Mobility > Global Configuration.**



Über die Kommandozeile:

```
# config t
```

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

Schritt 1: Erfassen Sie die Mobilitätskonfiguration beider 9800 WLCs.

Navigieren Sie für beide 9800 WLCs zu **Configuration > Wireless > Mobility > Global Configuration** und nimmt seine **Mobility Group Name** und **Mobility MAC Address.**

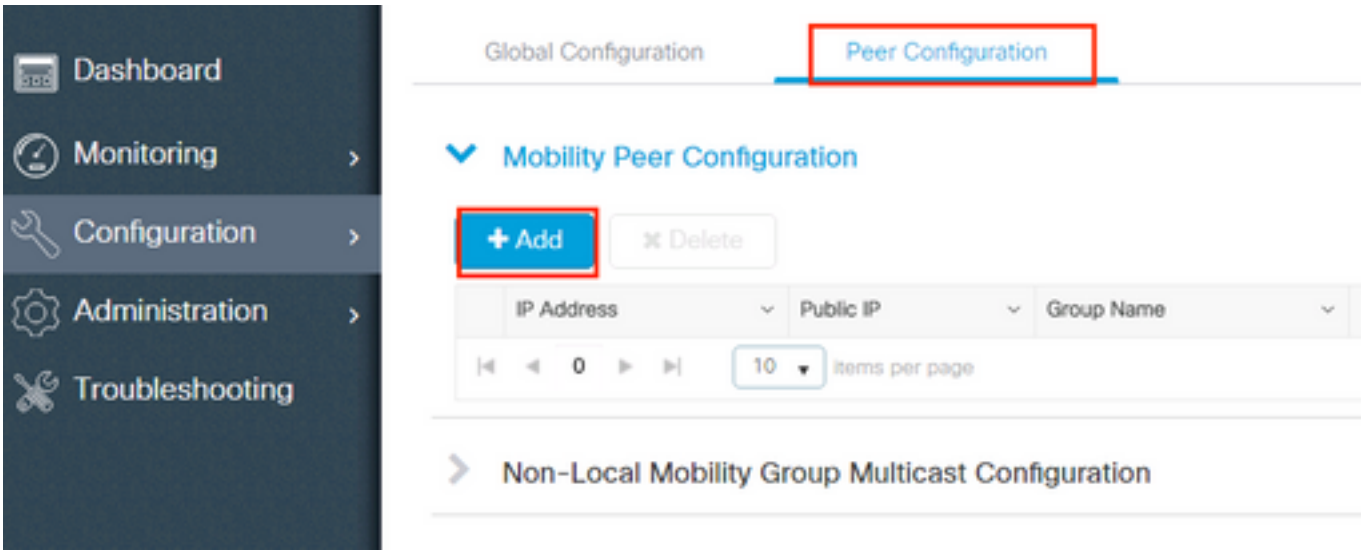
Über die Kommandozeile:

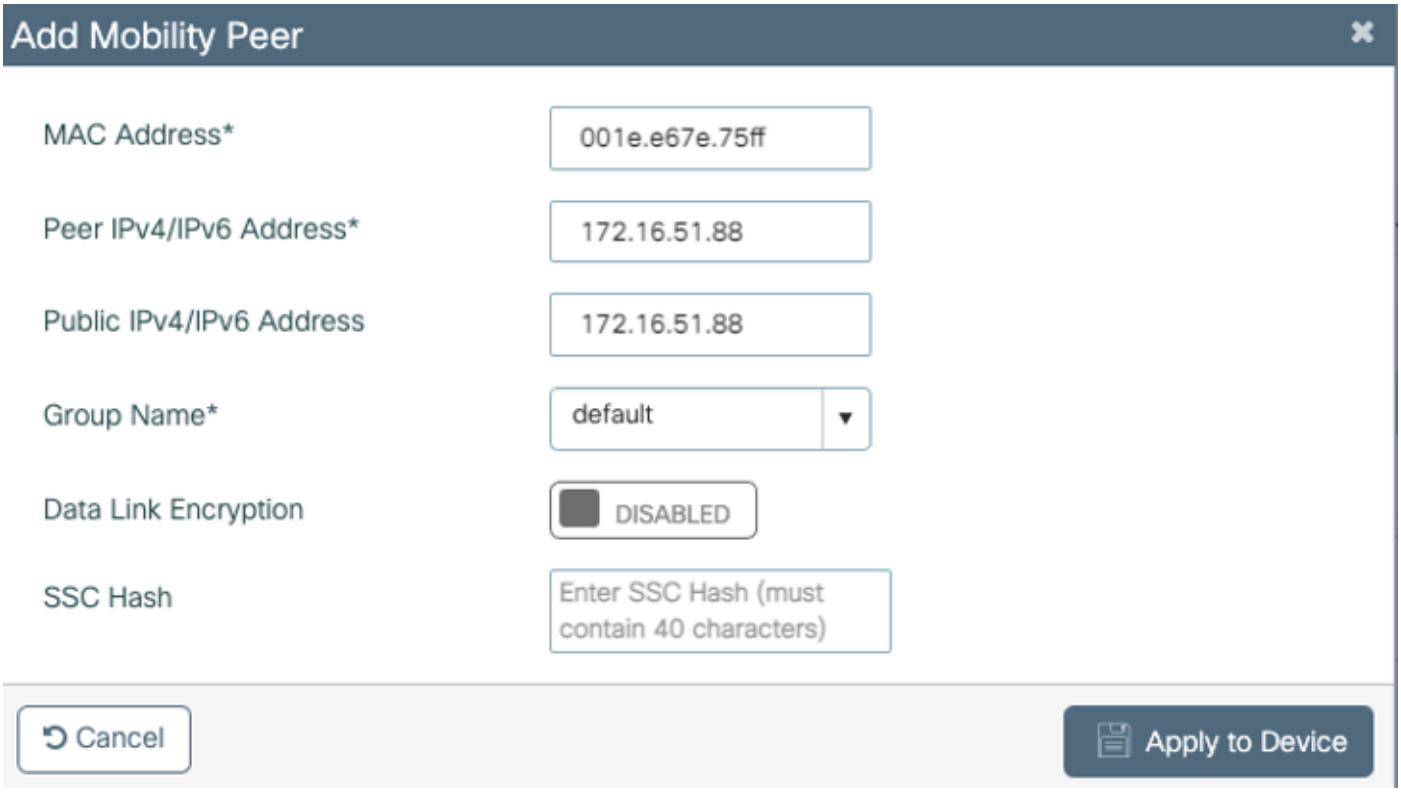
```
#show wireless mobility summary

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
```

Schritt 2: Peer-Konfiguration hinzufügen

Navigieren Sie zu **Configuration > Wireless > Mobility > Peer Configuration** und geben Sie die Informationen zum Peer-Controller ein. Führen Sie für beide 9800 WLCs den gleichen Vorgang aus.

Über die grafische Benutzeroberfläche:





Über die Kommandozeile:

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

Hinweis: Optional können Sie die Datenverschlüsselung aktivieren.

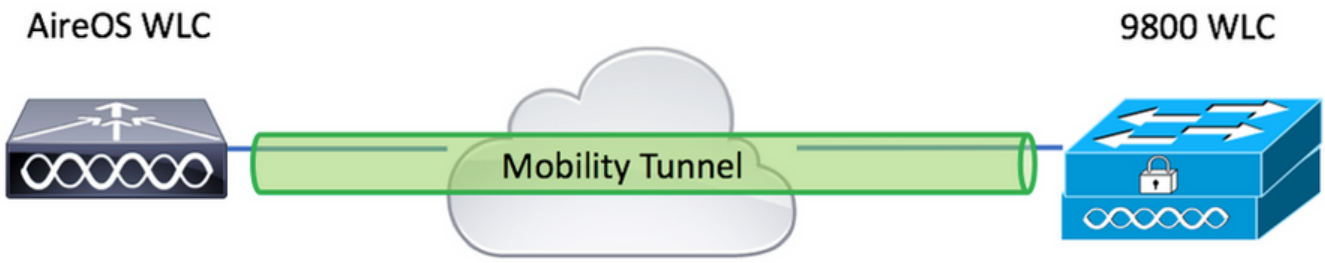
Mobility-Tunnel zwischen AireOS WLC- und 9800-CL-Controllern

Dieses Szenario ist normal für **brownfield** Bereitstellungen oder während der Controller-Migration, wobei das Netzwerk in einen Bereich von Access Points (APs) aufgeteilt wird, der von einem AireOS-Controller und einem anderen Access Point (9800) gesteuert wird.

Es ist ratsam, die APs über Controller in den einzelnen physischen oder RF-Bereichen zu verteilen, sodass Clients nur dann zwischen Controllern wechseln, wenn sie diese durchlaufen.

Vermeiden **salt and pepper** bereitstellung. Optional kann diese Mobilitätstopologie auch für **guest anchor** wobei 9800 als Foreign und ein AireOS als Anker-Controller agiert.

Netzwerkdiagramm



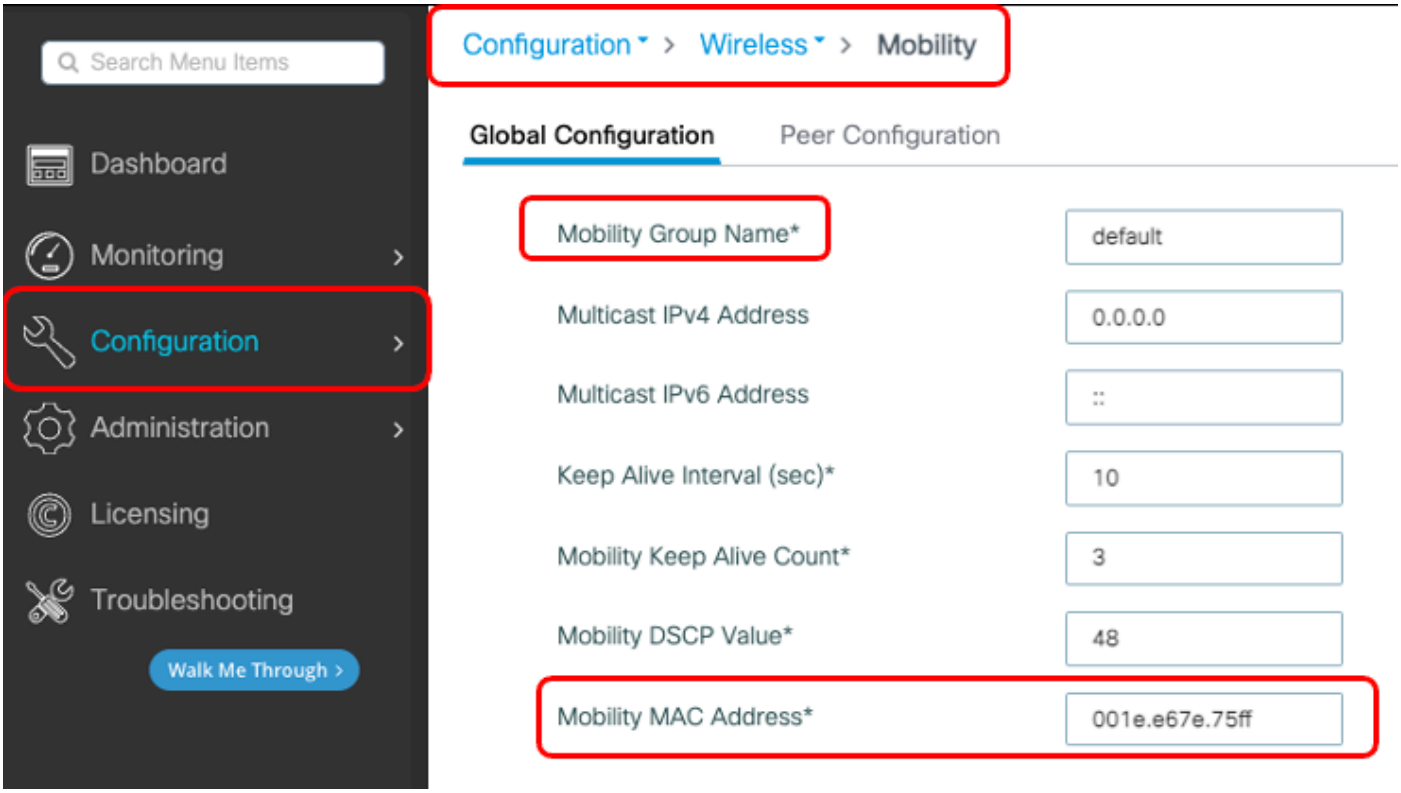
AireOS WLC-Konfiguration

Wenn Ihre 9800-Controller **High Availability**, stellen Sie sicher, dass Sie die MAC-Adresse für die Mobilität konfiguriert haben.

Schritt 1: Erfassen Sie 9800 WLC-Mobilitätsinformationen.

Über die grafische Benutzeroberfläche:

Navigieren Sie zu **Configuration > Wireless > Mobility > Global Configuration** und nimmt seine **Mobility Group Name** und **Mobility MAC Address**.



Über die Kommandozeile:

```
#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
```

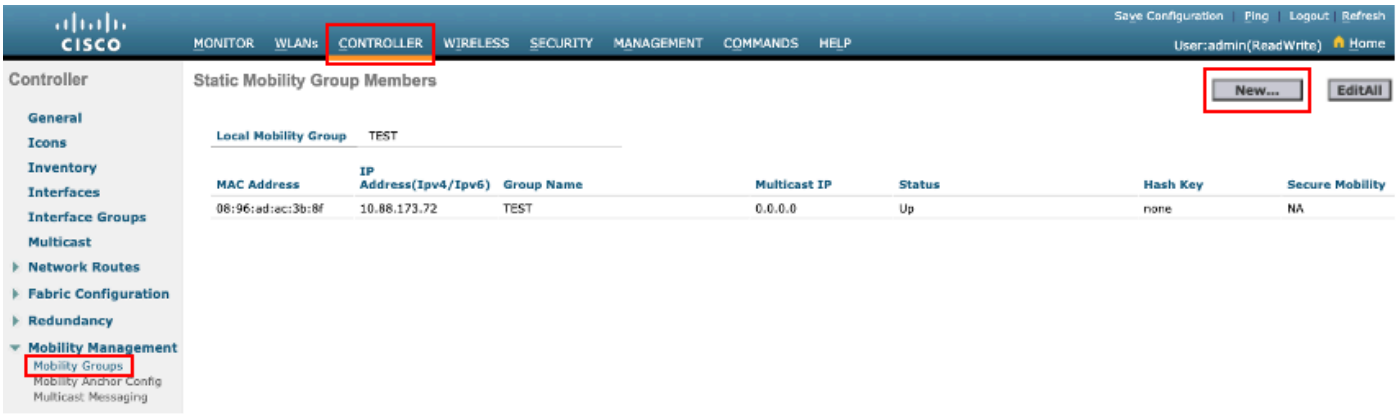
Schritt 2: Erfassen Sie den Hash-Wert des 9800 WLC.

```
# 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
```

Schritt 3: Fügen Sie die 9800 WLC-Informationen dem AireOS WLC hinzu.

Über die grafische Benutzeroberfläche:

Navigieren Sie zu **CONTROLLER > Mobility Management > Mobility Groups > New.**



Geben Sie die Werte ein und klicken Sie auf **Apply.**

> **Hinweis**: Hash ist nur erforderlich, wenn der 9800 ein selbstsigniertes Zertifikat wie den C9800-CL verwendet. Hardware-Appliances verfügen über ein SUDI-Zertifikat und benötigen keinen Hash (z. B. 9800-40, 9800-L usw.).

Über die Kommandozeile:

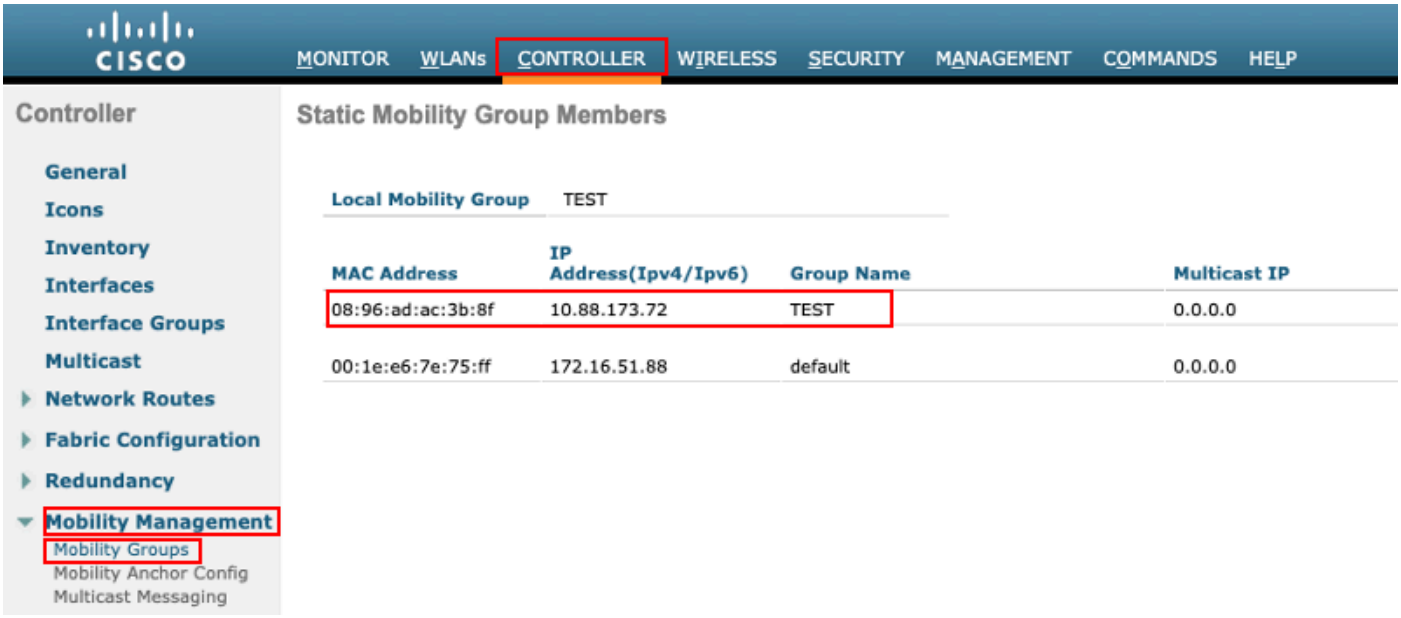
```
>config mobility group member add <9800 mac-address> <9800 WLC-IP> <group-name> encrypt enable
>config mobility group member hash <9800 WLC-IP> <9800 WLC-Hash>
>config mobility group member data-dtls <9800 mac-address> disable
```

9800 WLC-Konfiguration

Schritt 1: Sammeln von AireOS-Mobilitätsinformationen

Über die grafische Benutzeroberfläche:

Melden Sie sich bei der AireOS-Benutzeroberfläche an, und navigieren Sie zu **CONTROLLER > Mobility Management > Mobility Groups** und notieren Sie sich MAC-Adresse, IP-Adresse und Gruppennamen.



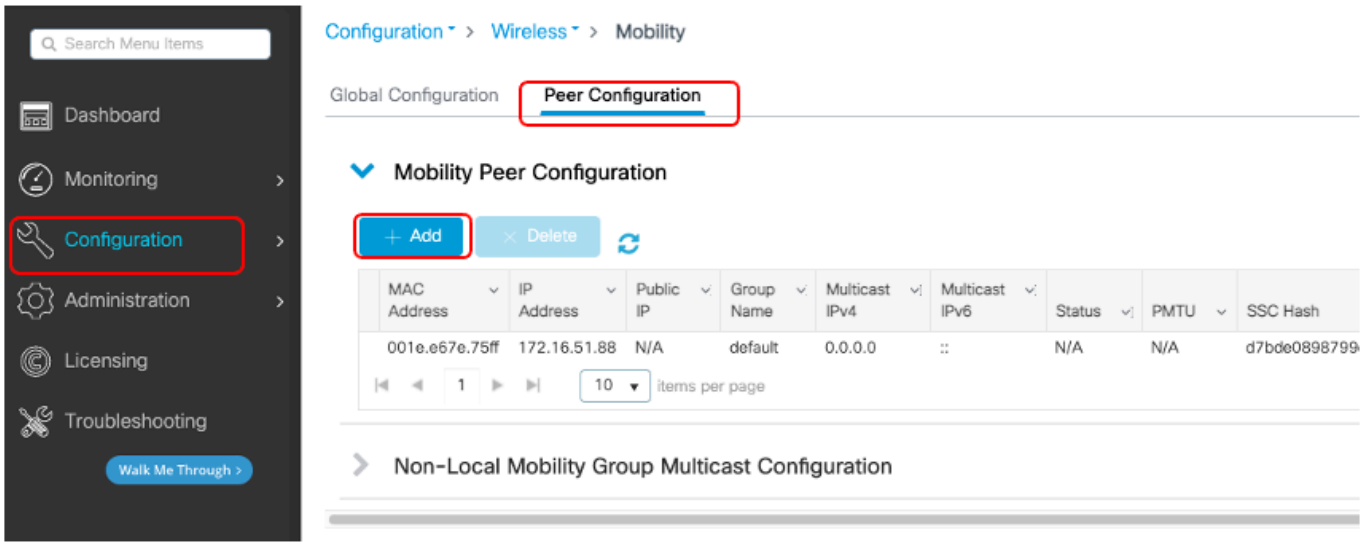
Über die Kommandozeile:

```
>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
```

Schritt 2: Hinzufügen der AireOS WLC-Informationen zum 9800 WLC

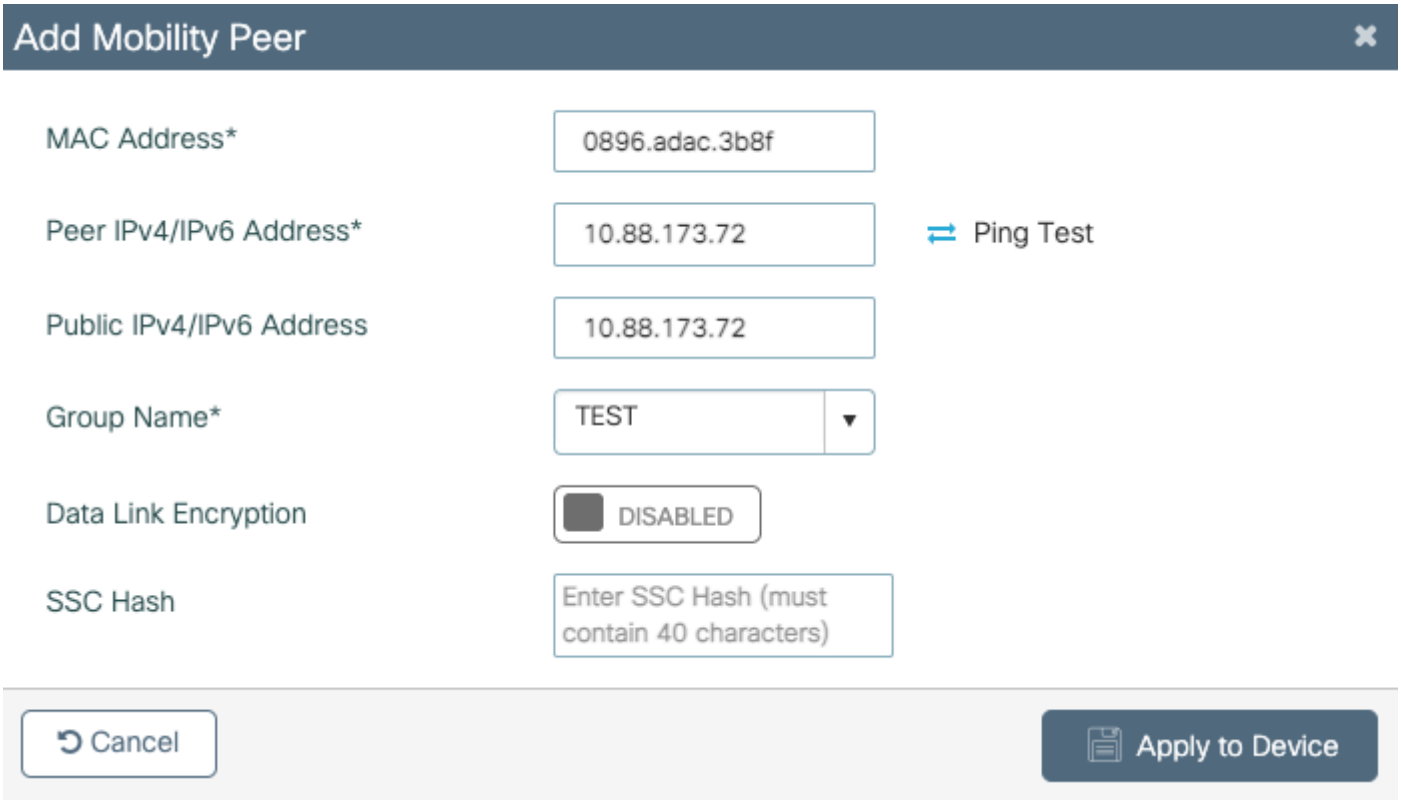
Über die grafische Benutzeroberfläche:

Navigieren Sie zu **Configuration > Wireless > Mobility > Peer Configuration > Add**



Geben Sie die AireOS-WLC-Informationen ein.

> **Hinweis**: Auf dem 9800 WLC ist die Verschlüsselung der Kontrollebene immer aktiviert, d. h. Sie müssen auf der AireOS-Seite die sichere Mobilität aktiviert haben. Die Verschlüsselung von Datenverbindungen ist jedoch optional. Wenn Sie die Funktion auf der 9800-Seite aktivieren, aktivieren Sie sie in AireOS mit: **config mobility group member data-dtls enable**



Über die Kommandozeile:

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

Überprüfung

Verwenden Sie diesen Abschnitt, um zu überprüfen, ob Ihre Konfiguration ordnungsgemäß funktioniert.

AireOS WLC-Überprüfung

```
>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 - Verifizierung

```
#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
```

Fehlerbehebung

In diesem Abschnitt finden Sie Informationen zur Fehlerbehebung bei Ihrer Konfiguration.

Verwenden Sie zum Debuggen des Prozesses die folgenden Befehle, um Probleme mit der Mobility-Tunnelimplementierung zu beheben:

AireOS-WLC

Schritt 1: Aktivieren Sie die Mobilitätsdebugs.

```
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
```

Schritt 2: Reproduzieren der Konfiguration und Überprüfen der Ausgabe

Beispiel für die erfolgreiche Erstellung eines Mobility-Tunnels auf einem AirOS WLC.

```
*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

Standardmäßig protokollieren die 9800-Controller fortlaufend Prozessinformationen, ohne dass ein spezielles Debugverfahren erforderlich ist.

Stellen Sie zur Fehlerbehebung einfach eine Verbindung mit dem Controller her, und rufen Sie die Protokolle der Wireless-Komponenten ab.

Die Protokolle können Tage umfassen, je nachdem, wie beschäftigt der Controller ist.

Um die Analyse zu vereinfachen, ziehen Sie die Protokolle mit einem Zeitbereich oder für die letzte Anzahl von Minuten (die Standardzeit ist 10 Minuten), und Sie können nach IP- oder MAC-Adressen filtern.

Schritt 1: Überprüfen Sie die aktuelle Uhrzeit auf dem Controller, damit Sie die Protokolle bis zum Auftreten des Problems nachverfolgen können.

```
# show clock
```

Schritt 2: Sammeln Sie die Controller-Protokolle, falls auf Cisco IOS-Ebene Informationen vorhanden sind, die mit dem Problem in Zusammenhang stehen könnten.

```
# show logging
```

Schritt 3: Sammeln Sie die stets verfügbaren Ablaufverfolgungen auf Benachrichtigungsebene für eine bestimmte Adresse. Sie können die Mobilitäts-Peer-IP oder -MAC zum Filtern verwenden.

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

Dieser Befehl generiert Protokolle für die letzten 10 Minuten. Es ist möglich, diese Zeit mit dem Befehl **show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt.**

Sie können den Inhalt der Sitzung anzeigen oder die Datei auf einen externen TFTP-Server kopieren.

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

or

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

Radio Active Tracing

Wenn die stets verfügbaren Protokolle nicht genügend Informationen bereitstellen, um zu wissen, welche Probleme bei der Tunnelkonfiguration ausgelöst wurden, können Sie bedingtes Debuggen und Erfassung aktivieren. **Radio Active (RA)** Spuren, die eine detailliertere Prozessaktivität liefern.

Schritt 1: Vergewissern Sie sich, dass noch keine Debugbedingungen aktiviert sind.

```
# show debugging
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop


IOSXE Packet Tracing Configs:




Packet Infra debugs:

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

Wenn eine Bedingung angezeigt wird, die nicht mit der zu überwachenden Adresse in Zusammenhang steht, deaktivieren Sie sie.

So entfernen Sie eine bestimmte Adresse:

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

So entfernen Sie alle Bedingungen (empfohlene Methode):

```
# clear platform condition all
```

Schritt 2: Fügen Sie die Debugbedingung für eine Adresse hinzu, die überwacht werden soll.

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

> **Hinweis**: Wenn Sie mehrere Mobilitäts-Peers gleichzeitig überwachen möchten, verwenden Sie einen **debug platform condition feature wireless mac** -Befehl pro MAC-Adresse aus.

Schritt 3: Lassen Sie den 9800 WLC die Überwachung der angegebenen Adressenaktivität starten.

```
# debug platform condition start
```

> **Hinweis**: Die Ausgabe der Mobilitätsaktivität wird nicht angezeigt, da alles intern gepuffert wird, um später erfasst zu werden.

Schritt 4: Reproduzieren Sie das Problem oder das Verhalten, das Sie überwachen möchten.

Schritt 5: Beenden Sie die Debugs.

```
# debug platform condition stop
```

Schritt 6: Sammeln Sie die Ausgabe der Adressaktivität.

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

Mit diesem Befehl werden die Protokolle der letzten 10 Minuten erstellt. Es ist möglich, diese Zeit mit dem Befehl **show logging profile wireless letzten 1 Stunde Filter MAC AAAA.BBBB.CCCC zu-Datei bootflash:ra-AAAA.BBB.CCCC.txt** anpassen.

Sie können die **FILENAME.txt** auf einen externen Server oder zeigen die Ausgabe direkt auf dem Bildschirm an.

Datei auf externen Server kopieren:

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

Inhalt anzeigen:

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

Schritt 7. Wenn Sie immer noch nicht in der Lage sind, den Grund für einen Fehler zu finden, erfassen Sie die interne Ebene der Protokolle.

(Sie müssen den Client nicht erneut debuggen. Verwenden Sie die Protokolle, die bereits intern gespeichert wurden, aber sammeln Sie eine größere Anzahl von ihnen).

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

Sie können die **FILENAME.txt** auf einen externen Server oder zeigen die Ausgabe direkt auf dem Bildschirm an.

Datei auf externen Server kopieren:

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

Inhalt anzeigen:

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

Schritt 8: Entfernen Sie die Debug-Bedingungen.

```
# clear platform condition all
```

> **Hinweis**: Entfernen Sie nach einer Fehlerbehebungssitzung immer die Fehlerbehebungsbedingungen.

Beispiel für die erfolgreiche Erstellung eines Mobility-Tunnels auf einem 9800 WLC.

```
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
```

Integrierte Paketerfassung

In den meisten Fällen ist es sehr nützlich, um Pakete zu überprüfen, die zwischen WLCs ausgetauscht werden. Es ist besonders nützlich, Aufnahmen mit **Access Control Lists (ACLs)** um den erfassten Datenverkehr zu begrenzen.

Dies ist eine Konfigurationsvorlage für eingebettete Erfassungen in der CLI.

Schritt 1: Erstellen Sie die Filter-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
```

Schritt 2: Definieren Sie die Erfassungsparameter:

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

> **Hinweis**: Verwaltungsoberfläche für Parameter INTERFACE_NAME auswählen

Schritt 3: Erfassung starten:

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

Schritt 4: Erfassung beenden:

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

Schritt 5: Navigieren Sie zu **Troubleshooting > Packet Capture on GUI (Fehlerbehebung > Paketerfassung in GUI), um die Paketerfassungsdatei zu sammeln.**

Häufige Fehlerbehebungsszenarien

Die nächsten Beispiele bestehen aus Tunneln, die zwischen 9800 WLCs gebildet werden.

Steuerung und Datenpfad aufgrund von Verbindungsproblemen ausgefallen

Aktivieren **Always-On-Logs** und **Embedded packet captures** um zusätzliche Informationen zur Fehlerbehebung bereitzustellen:

```
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
```

Paketerfassungen sind nützlich, um das Verhalten zu bestätigen.

```
 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
```

Sowohl debug als auch WLC zeigen, dass keine Antwort auf die Steuerungs- oder Daten-Pings vorliegt. Ein allgemeines Szenario zeigt, dass IP-Verbindungen zulässig sind, die Ports 16666 oder 16667 jedoch nicht über das Netzwerk kommunizieren dürfen.

Konfigurationskonflikt zwischen WLCs

In diesem Fall haben wir die Konnektivität für alle Ports zwischen WLCs bestätigt, aber es gibt weiterhin Fälle, in denen Keepalives fehlen.

Aktivieren **Always-On-Logs** und **Embedded packet captures** um zusätzliche Informationen zur Fehlerbehebung bereitzustellen:

```
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
```

Interne Protokolle auf Peer 172.16.55.28 helfen uns dabei, eine Konfigurationsdiskrepanz zu bestätigen

```
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
```

Häufige Konfigurationskonflikte: falscher Gruppenname, falsche Übereinstimmung bei **Data Link Encryption** und falsche Mobility-MAC-Adresse.

Gruppenkonfliktprotokoll:

```
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-Adresskonfliktprotokoll:

```
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-Handshake-Probleme

Diese Art von Problem ist mit DTLS-Tunneleinrichtungen zwischen WLCs verbunden. Dies kann der Fall sein, wenn der Datenpfad aktiv ist, der Steuerungspfad jedoch erhalten bleibt. **DOWN**.

Aktivieren **Always-On-Logs** und **Embedded packet captures** um zusätzliche Informationen zur Fehlerbehebung bereitzustellen:

```
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
```

Nutzung **show wireless management trustpoint** und **show crypto pki trustpoints commands** , um Ihre Zertifikatsinformationen zu überprüfen.

HA SSO-Szenario

Wenn Sie Controller in Hochverfügbarkeits-SSO-Paaren haben, gibt es einen wichtigen Haken zu wissen. Die MAC-Adresse für die Mobilität ist nicht standardmäßig konfiguriert. Sie kann bei einem Failover dazu führen, dass der Mobility-Tunnel ausfällt.

Die **Zusammenfassung zur Wireless-Mobilität anzeigen** enthält die aktuelle verwendete Mobility-MAC, ist jedoch nicht unbedingt konfiguriert. Überprüfen Sie, ob die Mobilitäts-MAC für die Konfiguration **"show run"** konfiguriert ist. **| i Mobilität**

Wenn die Mobilitäts-MAC nicht in der aktuellen Konfiguration konfiguriert ist, wird sie beim Failover auf den Standby-WLC geändert, was zu einem Ausfall der Mobility-Tunnel führt.

Die einfache Lösung besteht darin, auf der Webbenutzeroberflächenseite **Configuration > Wireless > Mobility (Konfiguration > Wireless > Mobilität**) zu navigieren und auf **Apply (Anwenden)** zu klicken. Dadurch wird die aktuelle MAC-Adresse für Mobilität in der Konfiguration gespeichert. Die MAC-Adresse bleibt dann bei Failover unverändert, und die Mobility-Tunnel bleiben erhalten.

Dieses Problem tritt hauptsächlich auf, wenn Sie Ihre Mobilitätskonfiguration über die Befehlszeile durchführen und die Konfiguration der Mobility-MAC-Adresse vergessen. Die Webbenutzeroberfläche speichert automatisch eine MAC-Adresse für die Mobilität, wenn Sie die Einstellungen übernehmen.

Zugehörige Informationen

- Konfigurieren der Mobilitätsfunktion von WLAN Anchor auf dem Catalyst 9800
- Technischer Support und Dokumentation für Cisco Systeme

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.