Configureer mobiliteitsplatforms op Catalyst 9800 draadloze LAN-controllers (WLC's)

Inhoud

Inleiding

Dit document beschrijft mobiliteitsconfiguratiescenario's die topologieën tussen Catalyst 9800

draadloze LAN-controllers (WLC's) en AireOS WLC's dekken.

Voorwaarden

Vereisten

Cisco raadt kennis van deze onderwerpen aan:

- CLI- of GUI-toegang tot de draadloze controllers.

Gebruikte componenten

- **AireOS WLC** versie 8.10 MR1 of hoger. U kunt ook **Inter Release Controller Mobility (IRCM)** speciale 8.5 afbeeldingen
- 980 WLC, Cisco IOS® XE v17.3.4

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Configureren

Netwerkdiagram

Richtsnoeren en beperkingen

1. **Mobility Group** de naam op 9800 uit het vak is "standaard".

Opmerking:
1) Zorg er in gevallen waarin WLC's in verschillende subnetten zijn, voor dat de poort UDP 16666 en 16667 ertussen open is.
2) Aanbevolen wordt dat beide 9800 WLC's dezelfde versie draaien, zodat clients die over roamen beschikken over een consistente ervaring in zowel Layer 3-roam- als gastenankerscenario's.

Mobiliteitstunnel tussen twee Catalyst 9800 WLC's

In dit basisvoorbeeld wordt beschreven hoe u mobiliteit kunt instellen voor twee 9800 controllers. Dit wordt vaak gebruikt voor de toegang van de Gast (anker), of om cliënten toe te staan om over controlemechanismen te zwerven en cliëntidentiteit te bewaren.

Wanneer u mobiliteit op C9800 configureert, is het eerste te kiezen ding de naam van de mobiliteitsgroep. De naam van de voorgevulde mobiliteitsgroep is een standaard, maar u kunt deze aan een gewenste waarde aanpassen.

U moet dezelfde naam van de mobiliteitsgroep configureren over controllers wanneer een snelle Layer 2-roam als **Fast Transition (FT)** of **Cisco Centralized Key Management (CCKM)** in gebruik is.

Standaard wordt het basis Ethernet MAC-adres van het chassis weergegeven in **show version** wordt weerspiegeld op GUI voor het adres van de mobiliteitsMAC.

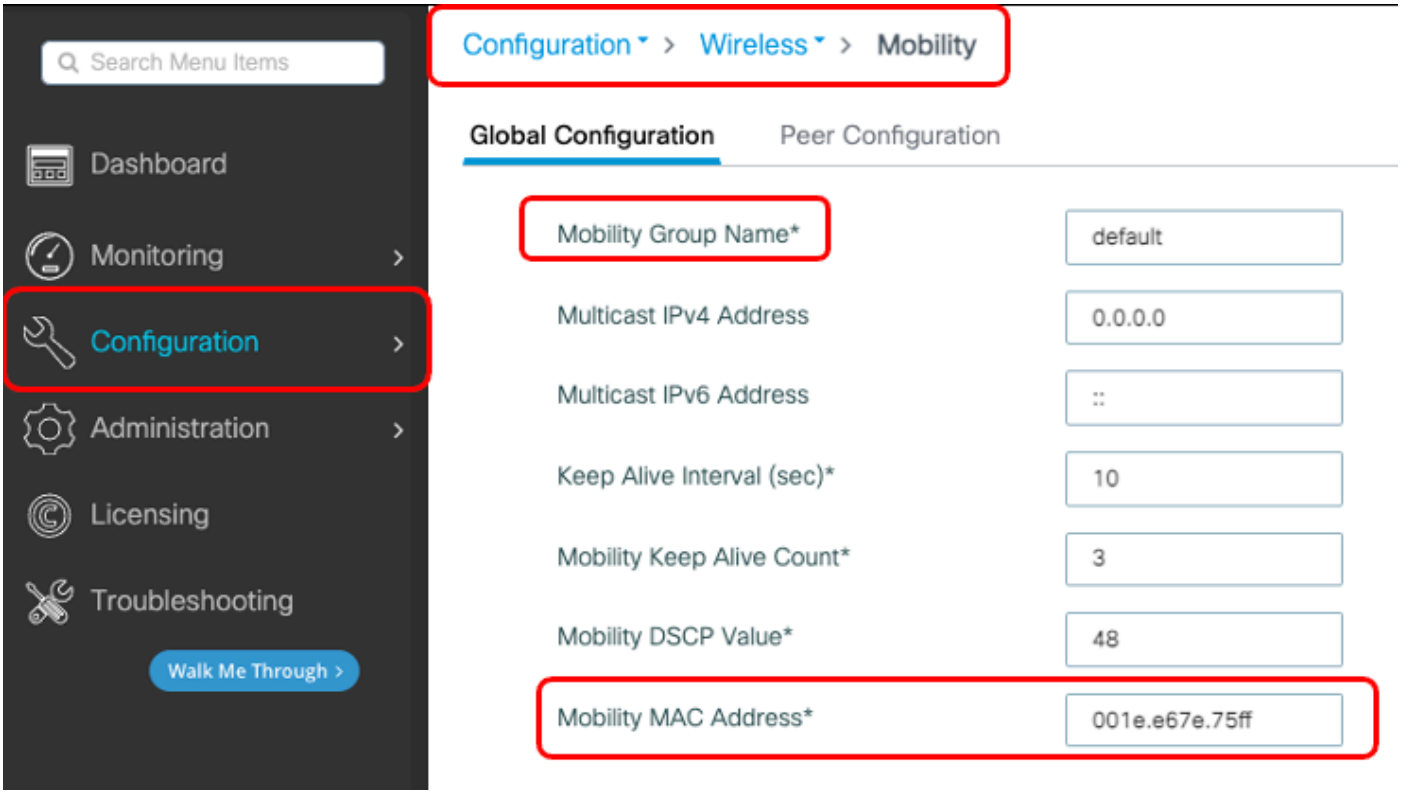
Op CLI is de mobiliteitskaart standaard 0000.000.000 zoals weergegeven in **show run all | inc mobility mac-address**

Wanneer 9800's zijn gekoppeld voor **High Availability (HA) Stateful Switchover (SSO):**

Als de configuratie standaard wordt gelaten en het chassis MAC-adres wordt gebruikt om een mobiliteitstunnel te vormen, falen het actieve chassis en de mobiliteitstunnel bij failover.

Daarom is het verplicht dat een mobiliteits-MAC-adres wordt geconfigureerd voor C9800 HA-paar.

Stap 1: Ga op GUI naar **Configuration > Wireless > Mobility > Global Configuration.**



Via de CLI:

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

```
# wireless mobility group name <mobility-group-name>
```

Stap 1. Verzamel mobiliteitsconfiguratie van beide 9800 WLCs.

Voor beide 9800 WLC's navigeer naar **Configuration > Wireless > Mobility > Global Configuration** en kennis te nemen van haar **Mobility Group Name** en **Mobility MAC Address**.

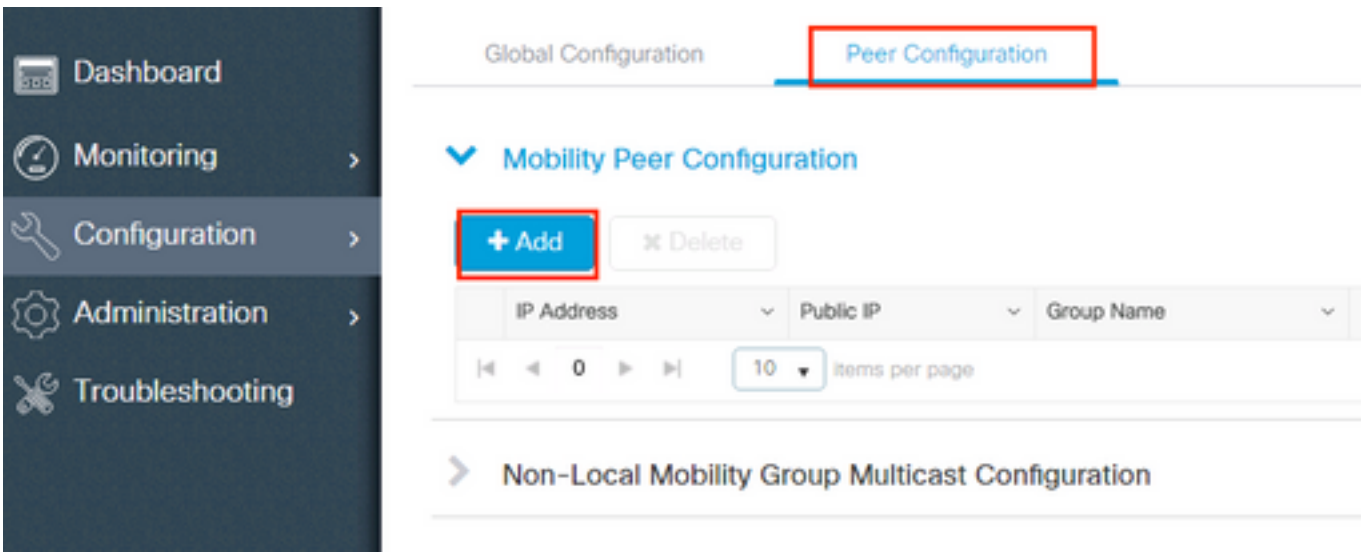
Via de CLI:

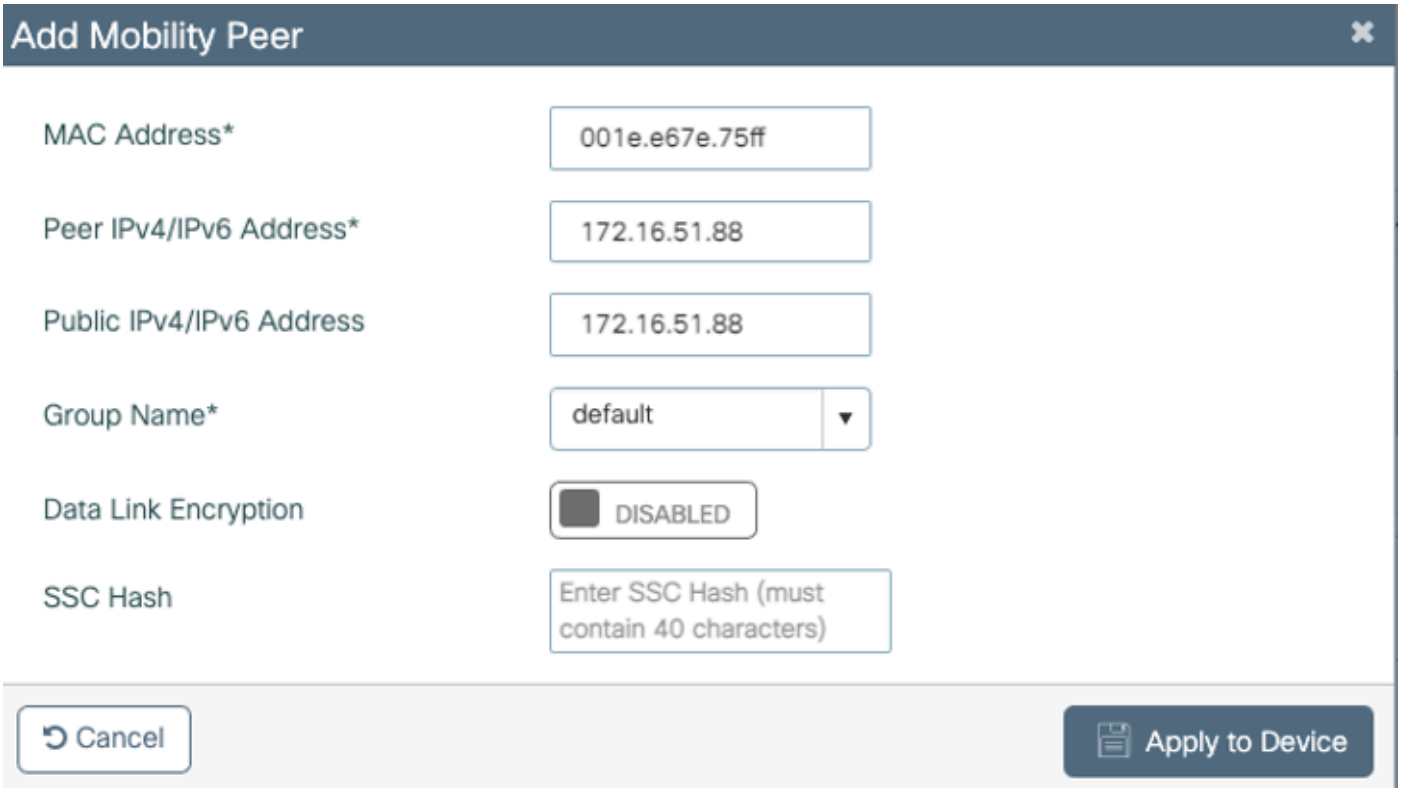
```
#show wireless mobility summary

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
```

Stap 2. Peer-configuratie toevoegen

Naar navigeren **Configuration > Wireless > Mobility > Peer Configuration** en voer de peer controller informatie in. Doe hetzelfde voor beide 9800 WLC's.

Via de GUI:





Via de CLI:

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

Opmerking: u kunt naar keuze Data Link Encryption inschakelen.

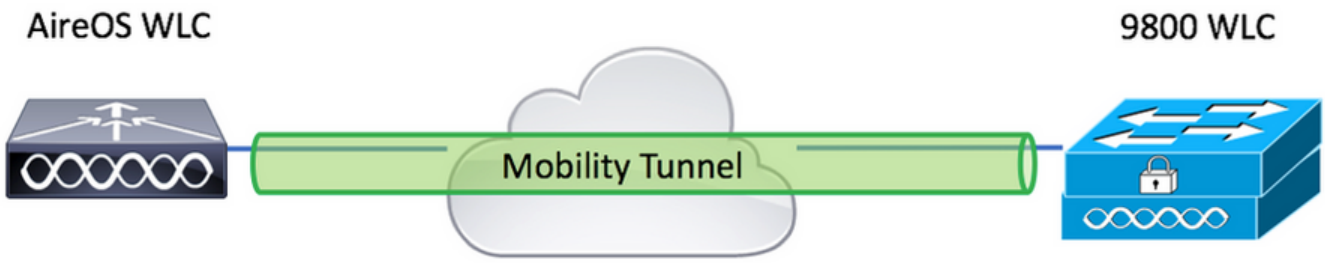
Mobiliteitstunnel tussen AireOS WLC- en 9800-CL-controllers

Dit scenario is normaal voor **brownfield** implementaties of tijdens controller-migratie, waarbij we het netwerk splitsen in een gebied van access points (AP's) die worden bestuurd door een AireOS-controller, en een ander door een 9800.

Het is aan te raden dat de AP's worden gedistribueerd over controllers per fysieke of RF-gebieden, zodat clients alleen tussen controllers zwerven wanneer ze bewegen.

vermijden **salt and pepper** implementatie. Optioneel kan deze mobiliteitstopologie ook worden gebruikt voor **guest anchor** waarbij 9800 optreedt als buitenlands en een AireOS als ankercontroller.

Netwerkdiagram



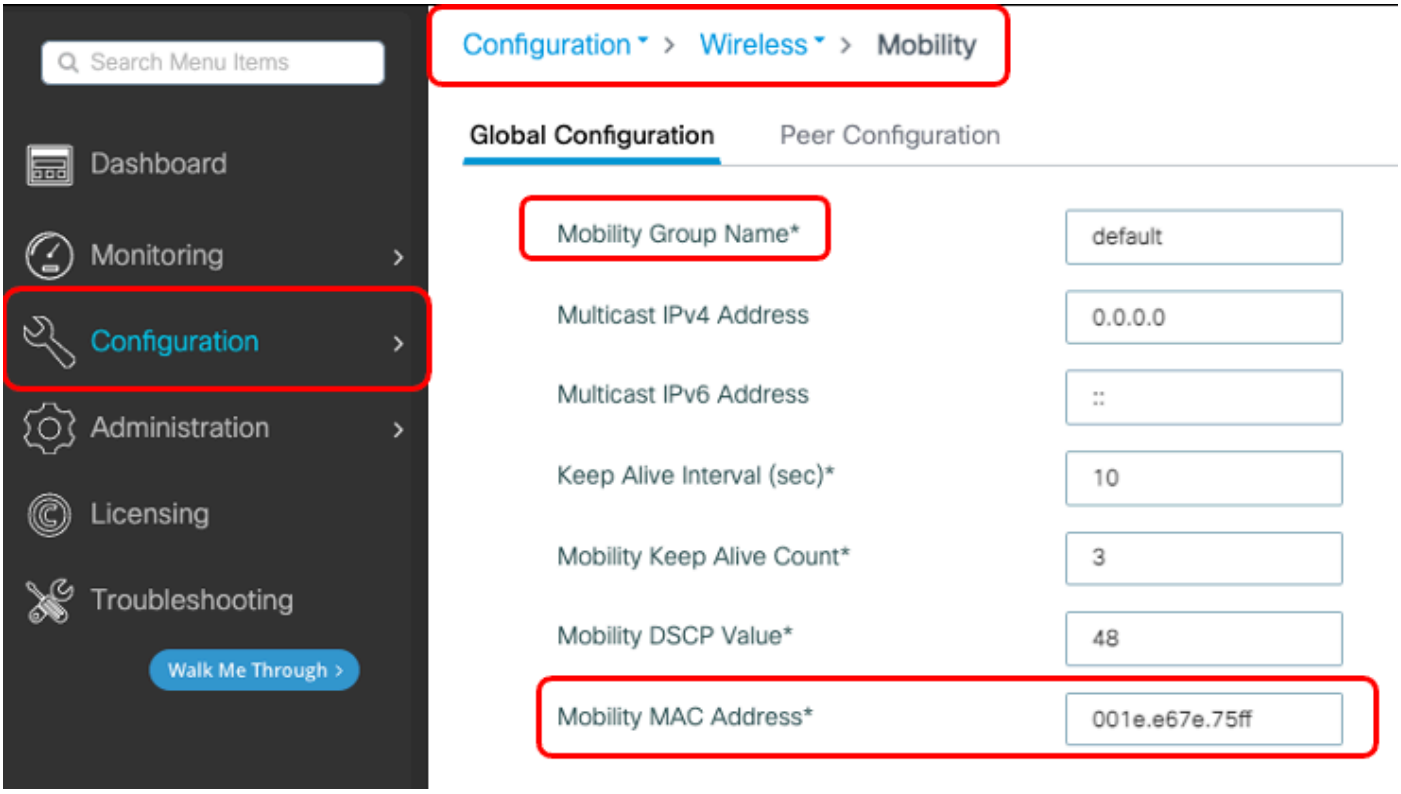
Configuratie AireOS WLC

Als uw 9800 controllers **High Availability**, zorg ervoor dat u het mobility MAC-adres hebt geconfigureerd.

Stap 1. Verzamel 9800 WLC-mobiliteitsinformatie.

Via de GUI:

Naar navigeren **Configuration > Wireless > Mobility > Global Configuration** en kennis te nemen van haar **Mobility Group Name** en **Mobility MAC Address.**



Via de 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
```

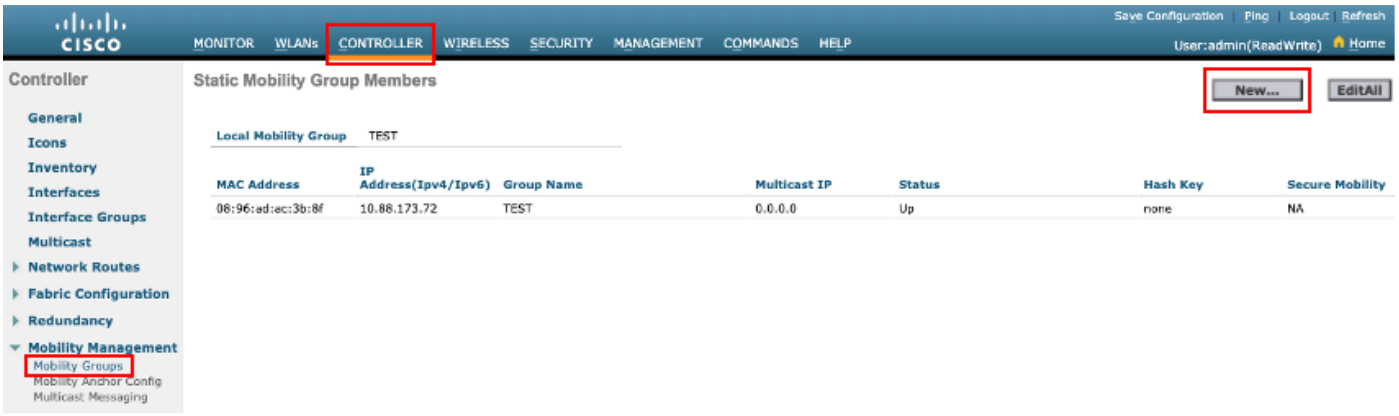
Stap 2. Verzamel de Hashwaarde van de 9800 WLC

```
# 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
```

Stap 3. Voeg de 9800 WLC-informatie toe aan de AireOS WLC.

Via de GUI:

Naar navigeren **CONTROLLER > Mobility Management > Mobility Groups > New.**



Voer de waarden in en klik op **Apply.**

> **Opmerking**: Hash is alleen vereist in gevallen waarin de 9800 een zelfondertekend certificaat gebruikt zoals de C9800-CL. Hardware-applicaties hebben een SUDI-certificaat en hebben geen hash nodig (bijvoorbeeld een 9800-40, 9800-L, enzovoort).

Via de CLI:

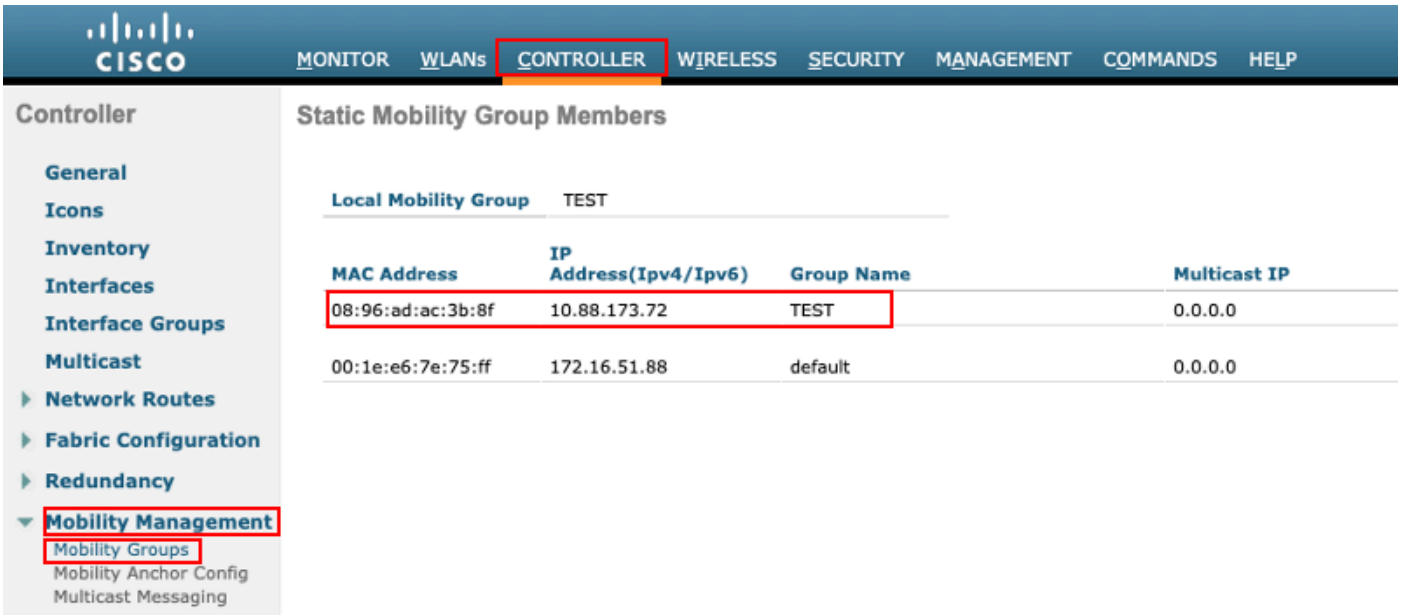
```
>config mobility group member add <9800 mac-address> <9800 WLC-IP> <group-name> encrypt enable
>config mobility group member hash <9800 WLC-IP> <9800 WLC-Hash>
>config mobility group member data-dtls <9800 mac-address> disable
```

Configuratie 9800 WLC

Stap 1. Verzamel AireOS-mobiliteitsinformatie.

Via de GUI:

Log in op AireOS GUI en navigeer naar **CONTROLLER > Mobility Management > Mobility Groups** en neem nota van MAC-adres, IP-adres en groepsnaam.



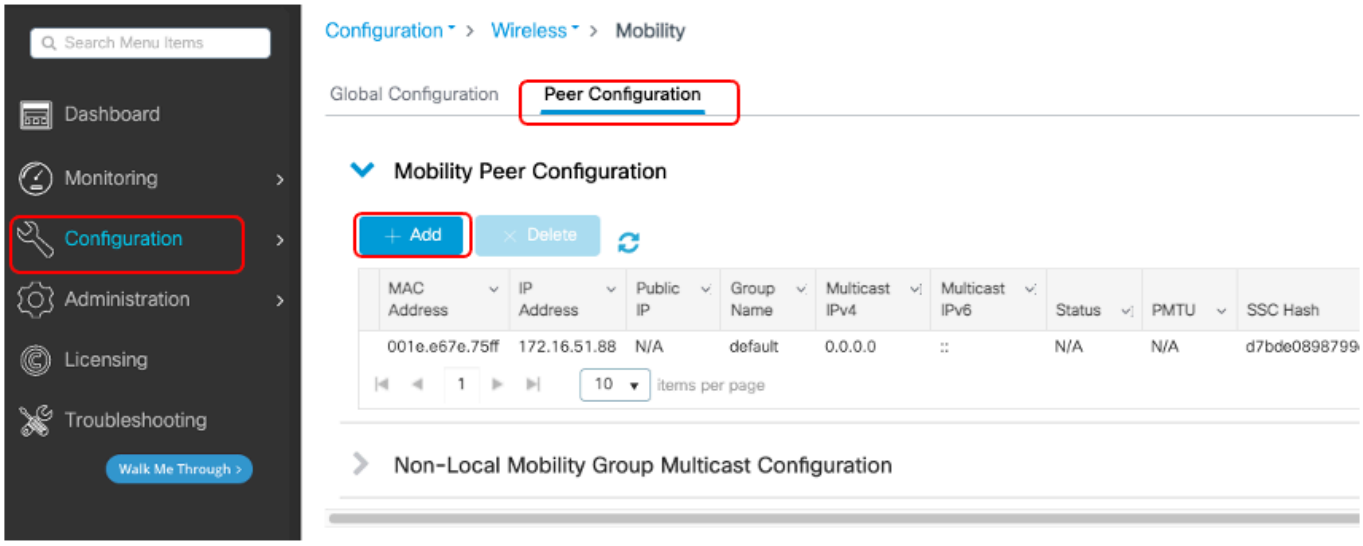
Via de CLI:

```
>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
```

Stap 2. Voeg de AireOS WLC-informatie toe aan de 9800 WLC

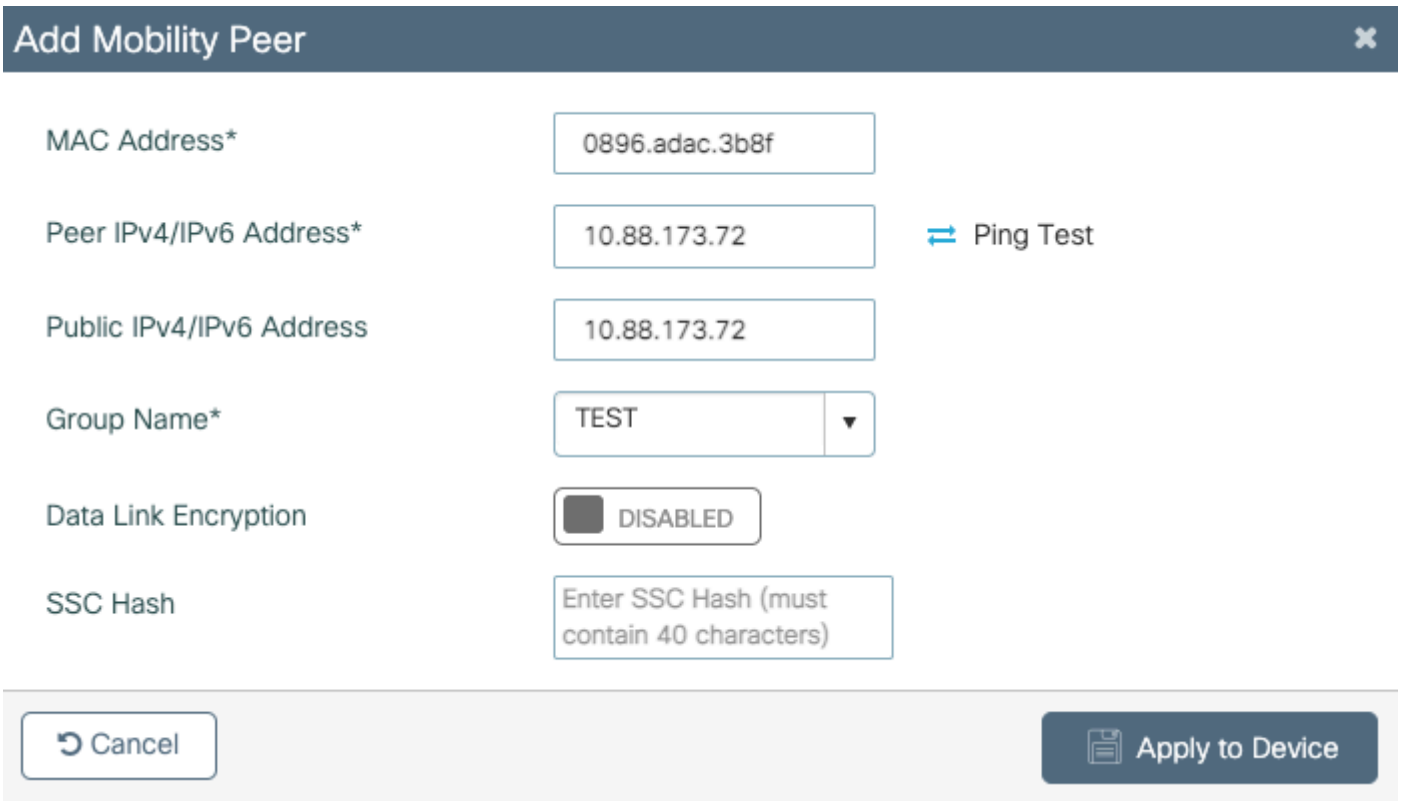
Via de GUI:

Naar navigeren **Configuration > Wireless > Mobility > Peer Configuration > Add**



Voer de informatie over AireOS WLC in.

> **Opmerking**: op de 9800 WLC is de besturingsplane-encryptie altijd ingeschakeld, wat betekent dat u beveiligde mobiliteit moet hebben ingeschakeld aan de kant van AireOS. De codering van de datalink is echter optioneel. Als u het inschakelt aan de 9800-kant, kunt u het inschakelen op AireOS met: **config Mobility Group data-dtls inschakelen**



Via de CLI:

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

Verifiëren

Gebruik deze sectie om te controleren of uw configuratie goed werkt.

AireOS WLC-verificatie

```
>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-verificatie

```
#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
```

Problemen oplossen

Deze sectie verschaft informatie die wordt gebruikt om problemen met uw configuratie op te lossen.

Om de implementatie van de mobiliteitstunnel problemen op te lossen, gebruikt u deze opdrachten om het proces te debuggen:

AireOS WLC

Stap 1. Schakel het debuggen van de mobiliteit in.

```
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
```

Stap 2. Reproduceer de configuratie en controleer de uitvoer

Voorbeeld van een succesvolle mobiliteitstunnel op een 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

Standaard logt de 9800 controllers continu procesinformatie in zonder dat er een speciale debug procedure nodig is.

Maak eenvoudig verbinding met de controller en haal de logbestanden op die zijn gekoppeld aan een draadloze component voor probleemoplossingsdoeleinden.

De logbestanden kunnen dagen duren; dat hangt af van hoe druk de controller is.

Om analyse te vereenvoudigen, trek de logbestanden met een tijdbereik of voor het laatste aantal minuten (de standaardtijd is ingesteld op 10 minuten) en u kunt filteren op IP- of MAC-adressen.

Stap 1. Controleer de huidige tijd op de controller, zodat u de logbestanden kunt volgen in de tijd terug naar toen het probleem zich voordeed.

```
# show clock
```

Stap 2. Verzamel de controllerlogboeken, voor het geval dat er om het even welke informatie op Cisco IOS niveau is die met het probleem zou kunnen worden verwant.

```
# show logging
```

Stap 3. Verzamel de altijd-op berichtniveau sporen voor een specifiek adres. U kunt de mobiliteit peer IP of MAC gebruiken om te filteren.

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

Met deze opdracht worden logbestanden gegenereerd voor de laatste 10 minuten. Deze tijd kan met de opdracht worden aangepast **show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt.**

U kunt de inhoud op de sessie weergeven of het bestand naar een externe TFTP-server kopiëren.

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

or

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

Radio actief overtrekken

Als de altijd ingeschakelde logbestanden niet voldoende informatie bieden om te weten wat de geactiveerde problemen tijdens tunnelconfiguratie zijn, kunt u voorwaardelijke debugs inschakelen en vastleggen **Radio Active (RA)** sporen, die een meer gedetailleerde procesactiviteit geven.

Stap 1. Controleer of er geen debug voorwaarden zijn die al zijn ingeschakeld.

```
# show debugging
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop


IOSXE Packet Tracing Configs:




Packet Infra debugs:

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

Als u een voorwaarde ziet die niet gerelateerd is aan het adres dat u wilt controleren, schakelt u deze uit.

Zo verwijdert u een specifiek adres:

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

Zo verwijdert u alle omstandigheden (aanbevolen manier):

```
# clear platform condition all
```

Stap 2. Voeg de debug-voorwaarde toe voor een adres dat u wilt controleren.

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

> **Opmerking**: als u meer dan één mobiliteitspeer tegelijkertijd wilt bewaken, gebruikt u een **debug platform condition feature wireless mac** opdracht per MAC-adres.

Stap 3. Heb de 9800 WLC om monitor van de gespecificeerde adresactiviteit te beginnen.

```
# debug platform condition start
```

Opmerking: Uitvoer van de mobiliteitsactiviteit wordt niet weergegeven, omdat alles intern wordt gebufferd om later te worden verzameld.

Stap 4. Reproduceer het probleem of het gedrag dat u wilt controleren.

Stap 5. Stop de debugs.

```
# debug platform condition stop
```

Stap 6. Verzamel de output van de adresactiviteit.

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

Met deze opdracht worden logbestanden gegenereerd voor de laatste 10 minuten. Het is mogelijk om deze tijd aan te passen met de opdracht **toon logboekprofiel draadloos laatste 1 uur filter mac AAA.BBB.CCCC to-file bootflash:ra-AAA.BBB.CCCC.txt**.

U kunt de **FILENAME.txt** naar een externe server of de uitvoer rechtstreeks op het scherm weergeven.

Kopieert het bestand naar een externe server:

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

Geef de inhoud weer:

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

Stap 7. Als u nog steeds niet in staat bent om de reden van een fout te vinden, verzamel dan het interne niveau van de logbestanden.

(U hoeft de client niet opnieuw te debuggen. Gebruik de logbestanden die al intern waren opgeslagen, maar verzamel een breder bereik van hen).

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

U kunt de **FILENAME.txt** naar een externe server of de uitvoer rechtstreeks op het scherm weergeven.

Kopieert het bestand naar een externe server:

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

Geef de inhoud weer:

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

Stap 8. Verwijder de debug-voorwaarden.

```
# clear platform condition all
```

Opmerking: verwijder altijd de debug-voorwaarden na een probleemoplossingssessie.

Voorbeeld van een succesvolle mobiliteitstunnel op een 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
```

Ingesloten pakketvastlegging

Meestal is het zeer nuttig om pakketten te controleren die tussen WLCs worden uitgewisseld. Het

is vooral handig om opnamen te filteren met **Access Control Lists (ACLs)** om opgenomen verkeer te beperken.

Dit is een configuratiesjabloon voor ingesloten opnamen op CLI.

Stap 1. Maak de 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
```

Stap 2. Definieer de opnameparameters:

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

> **Opmerking**: Selecteer beheerinterface voor INTERFACE_NAME parameter

Stap 3. Start de vastlegging:

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

Stap 4. Stop de vastlegging:

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

Stap 5. Ga naar **Problemen oplossen > Packet Capture op GUI om het pakketopnamebestand te verzamelen.**

Gemeenschappelijke probleemoplossingsscenario's

De volgende voorbeelden bestaan uit tunnels gevormd tussen 9800 WLCs.

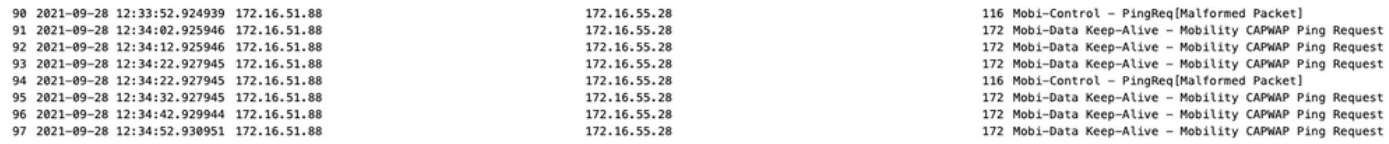
Beheer en gegevenspad omlaag vanwege connectiviteitsproblemen

Inschakelen **Always-On-Logs** en **Embedded packet captures** extra informatie geven voor probleemoplossing:

```
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**

Packet-opnamen zijn nuttig om gedrag te bevestigen.



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

Het bericht dat zowel debug als WLC tonen dat er geen reactie op de Controle of de Gegevens pings is. Een gemeenschappelijk scenario toont aan IP de connectiviteit wordt toegestaan maar de havens 16666 of 16667 niet om over het netwerk worden toegestaan te communiceren.

Configuratie-mismatch tussen WLC's

In dit geval bevestigden we connectiviteit voor alle poorten tussen WLC's, maar blijven zien keepalives missen.

Inschakelen **Always-On-Logs** en **Embedded packet captures** extra informatie geven voor probleemoplossing:

```
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 logs op peer 172.16.55.28 helpen ons te bevestigen configuratie mismatch

```
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
```

Veelvoorkomende configuratie mismatch omvatten: onjuiste groepsnaam, mismatch op **Data Link Encryption** en onjuist Mobility Mac-adres.

Logboek voor groepsmismatch:

```
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.
```

Logboek voor MAC-adresmismatch:

```
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.
```

Problemen met DTLS-handdruk

Dit soort kwestie houdt verband met DTLS-tunnelinrichtingen tussen WLC's. Het kan het geval zijn dat het gegevenspad UP is, maar het controlepad blijft **DOWN**.

Inschakelen **Always-On-Logs** en **Embedded packet captures** extra informatie geven voor probleemoplossing:

```
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
```

Gebruik **show wireless management trustpoint** en **show crypto pki trustpoints commands** om uw certificaatinformatie te verifiëren.

Het HA SSO-scenario

Als u controllers in High Availability SSO-paar hebt, is er een belangrijke vangst om te weten. Het mobiele MAC-adres wordt standaard niet geconfigureerd en kan ervoor zorgen dat de mobiliteitstunnel omlaag gaat als er een failover gebeurt.

De **show draadloze mobiliteitssamenvatting** geeft u de huidige mobiliteit MAC in gebruik, maar het is niet noodzakelijk geconfigureerd. Controleer of de configuratie de mobiliteit MAC geconfigureerd heeft met **show run | i mobiliteit**

Als de mobiliteitscamera niet in de lopende configuratie is geconfigureerd, verandert deze bij failover in de standby WLC en dit zorgt ervoor dat mobiliteitstunnels falen.

De simpele oplossing is om naar de **Configuration > Wireless > Mobility** web UI pagina te navigeren en te klikken **zijn van toepassing**. Hiermee slaat u de huidige mobiliteitsMAC op in de configuratie. De MAC blijft dan hetzelfde bij failover en de mobiliteitstunnels blijven behouden.

Deze kwestie gebeurt vooral als u uw mobiliteitsconfiguratie door de opdrachtregel doet en vergeet het mobiliteits-MAC-adres te configureren. De web UI slaat automatisch een mobiliteits MAC adres op wanneer u de instellingen toepast.

Gerelateerde informatie

- WLAN-ankermobiliteit op Catalyst 9800 configureren
- Technische ondersteuning en documentatie – Cisco Systems

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.