Configurazione delle topologie di mobilità sui controller WLC (Wireless LAN Controller) Catalyst 9800

Sommario

Introduzione

In questo documento vengono descritti gli scenari di configurazione della mobilità che riguardano le topologie tra i Wireless LAN Controller (WLC) Catalyst 9800 e i WLC AireOS.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Accesso CLI o GUI ai controller wireless.

Componenti usati

- **AireOS WLC** versione 8.10 MR1 o successiva. È inoltre possibile utilizzare **Inter Release Controller Mobility (IRCM)** immagini speciali 8.5
- 9800 WLC, Cisco IOS® XE v17.3.4

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Configurazione

Esempio di rete

Linee guida e limitazioni

1. **Mobility Group** name on 9800 out of the box è "default".

Nota:
1) Nei casi in cui i WLC si trovano in subnet diverse, verificare che le porte UDP 1666 e 1667 siano aperte tra loro.
2) Si consiglia che entrambi i WLC 9800 eseguano la stessa versione in modo che i client che eseguono il roaming abbiano un'esperienza coerente sia nel roaming di layer 3 che negli scenari di ancoraggio guest.

Tunnel per la mobilità tra due Catalyst 9800 WLC

In questo esempio di base viene descritto come configurare la mobilità tra due controller 9800. Viene in genere utilizzato per l'accesso guest (ancoraggio) o per consentire ai client di spostarsi tra i controller e mantenere l'identità del client.

Quando si configura la mobilità su C9800, è innanzitutto necessario scegliere il nome del gruppo di mobilità. Il nome del gruppo di mobilità precompilato è un valore predefinito, ma è possibile personalizzarlo in base al valore desiderato.

Ènecessario configurare lo stesso nome del gruppo di mobilità tra i controller quando un roaming veloce di layer 2 è simile a **Fast Transition (FT)** o **Cisco Centralized Key Management (CCKM)** è in uso.

Per impostazione predefinita, l'indirizzo MAC Ethernet di base dello chassis come mostrato nella **show version** viene riflessa sulla GUI per l'indirizzo MAC della mobilità.

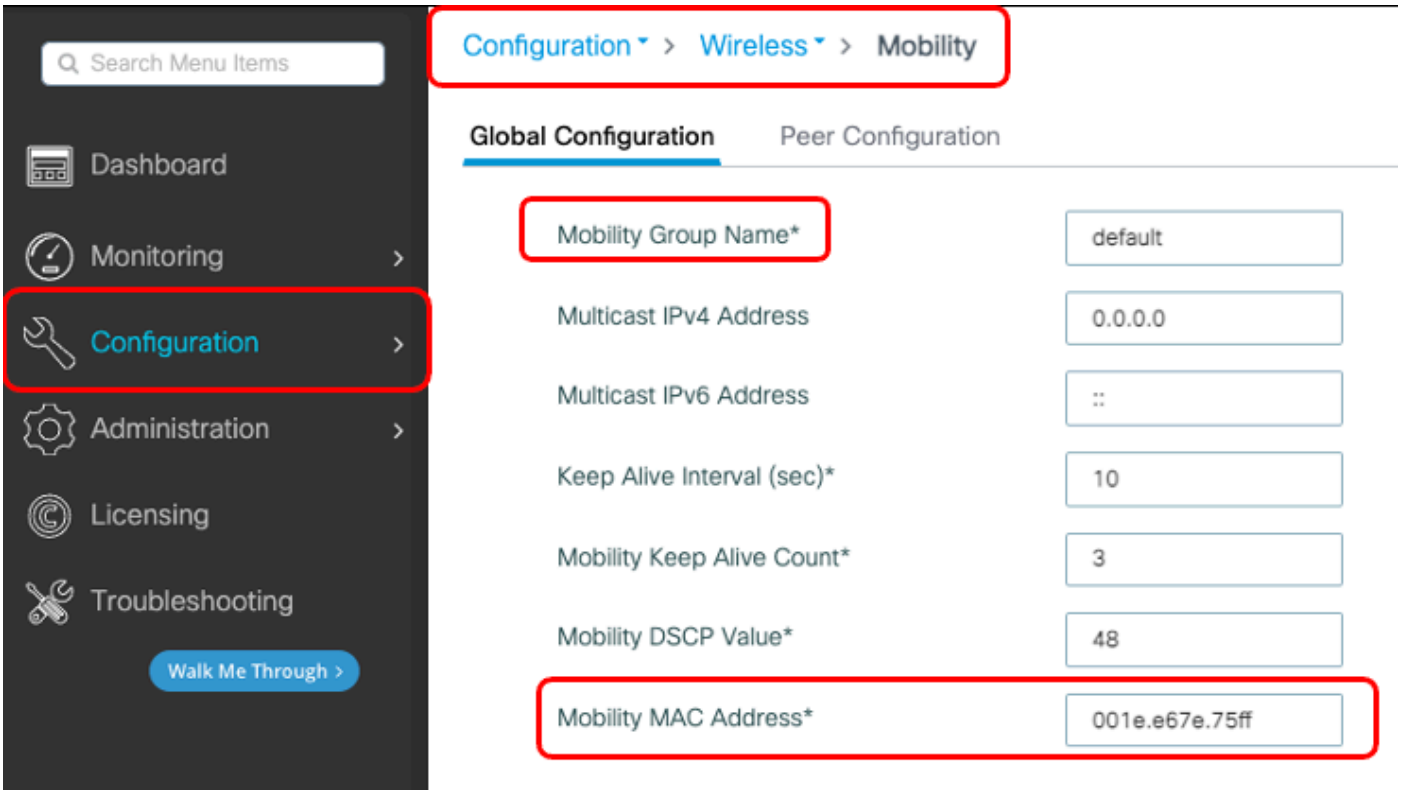
Dalla CLI, per impostazione predefinita, il mac per la mobilità è 0000.0000.000, come mostrato nella **show run all | inc mobility mac-address**

Nei casi in cui 9800 vengono accoppiati per **High Availability (HA) Stateful Switchover (SSO):**

Se la configurazione viene lasciata invariata e l'indirizzo MAC dello chassis viene utilizzato per formare il tunnel di mobilità, lo chassis attivo e il tunnel di mobilità avranno esito negativo in caso di failover.

Pertanto, è necessario configurare un indirizzo MAC di mobilità per la coppia C9800 HA.

Passaggio 1: Sulla GUI, passare a **Configuration > Wireless > Mobility > Global Configuration.**



Dalla CLI:

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

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

Passaggio 1. Raccogliere la configurazione della mobilità di entrambi i WLC.

Per entrambi i WLC, passare a **Configuration > Wireless > Mobility > Global Configuration** e prende atto della **Mobility Group Name** e **Mobility MAC Address.**

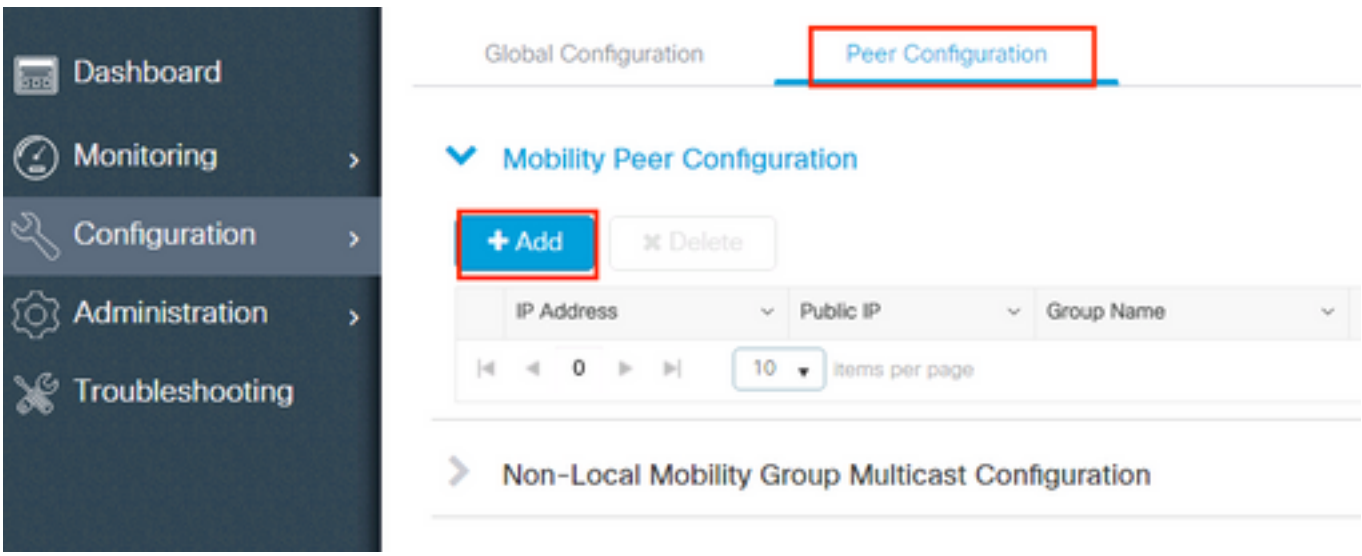
Dalla CLI:

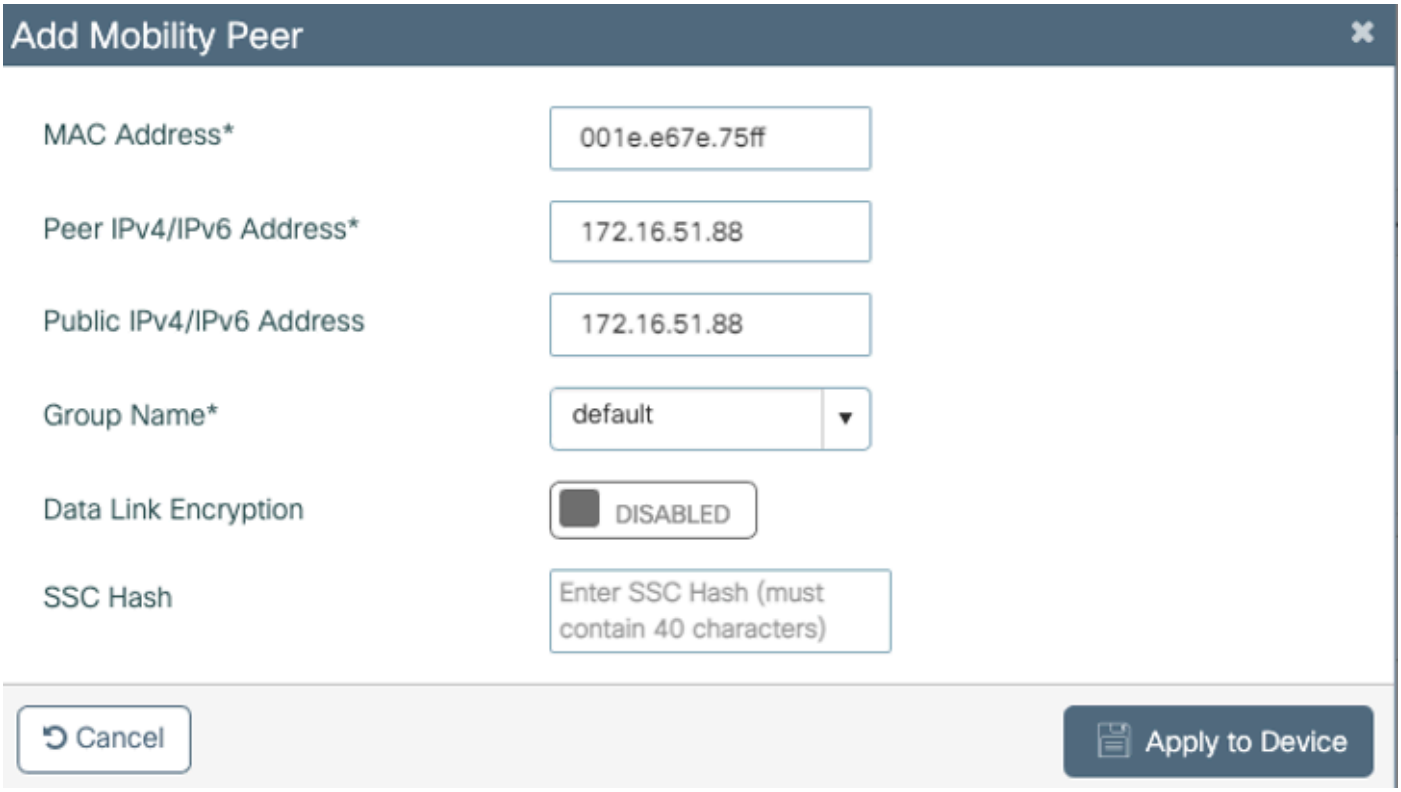
```
#show wireless mobility summary

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
```

Passaggio 2. Aggiungi configurazione peer

Passa a **Configuration > Wireless > Mobility > Peer Configuration** e immettere le informazioni sul controller peer. Fate lo stesso per entrambi i WLC 9800.

Dalla GUI:





Dalla CLI:

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

Nota: è possibile abilitare la crittografia dei collegamenti dati.

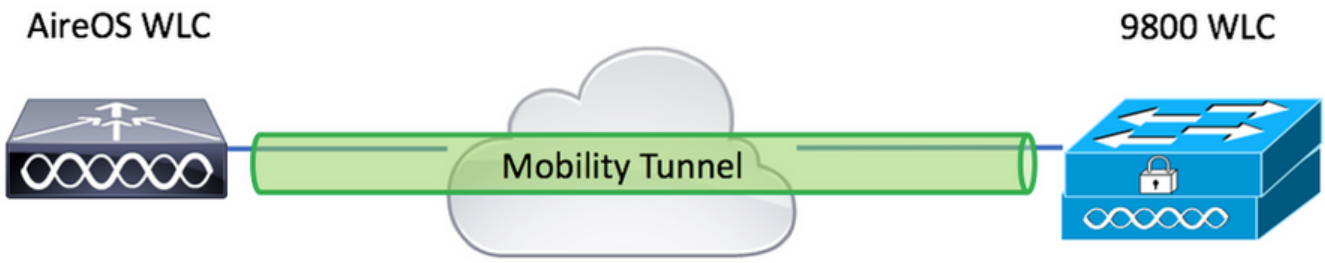
Tunnel di mobilità tra i controller AireOS WLC e 9800-CL

Questo scenario è normale per **brownfield** o durante la migrazione dei controller, in cui la rete viene suddivisa in un'area di punti di accesso (AP) controllata da un controller AireOS e un'altra da un 9800.

Si consiglia di distribuire gli access point tra i controller per area fisica o RF, in modo che i client eseguano il roaming solo tra i controller quando si spostano.

Evitare **salt and pepper** implementazione. Facoltativamente, questa topologia di mobilità può essere utilizzata anche per **guest anchor** dove 9800 agisce come esterno e AireOS come controller di ancoraggio.

Esempio di rete



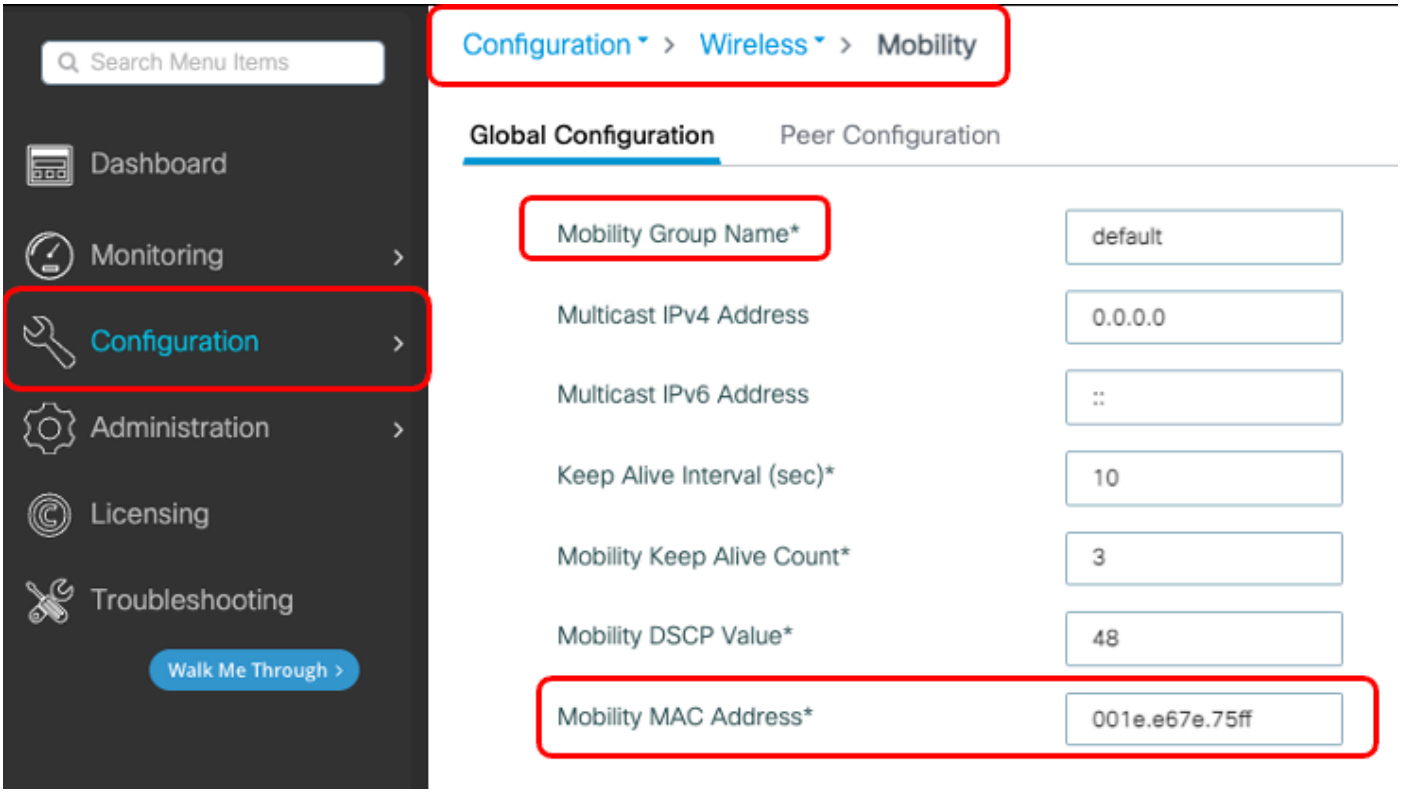
Configurazione AireOS WLC

Se i controller 9800 sono in **High Availability**verificare di aver configurato l'indirizzo MAC per la mobilità.

Passaggio 1. Raccogli informazioni sulla mobilità del WLC 9800.

Dalla GUI:

Passa a **Configuration > Wireless > Mobility > Global Configuration** e prende atto della **Mobility Group Name** e **Mobility MAC Address.**



Dalla 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
```

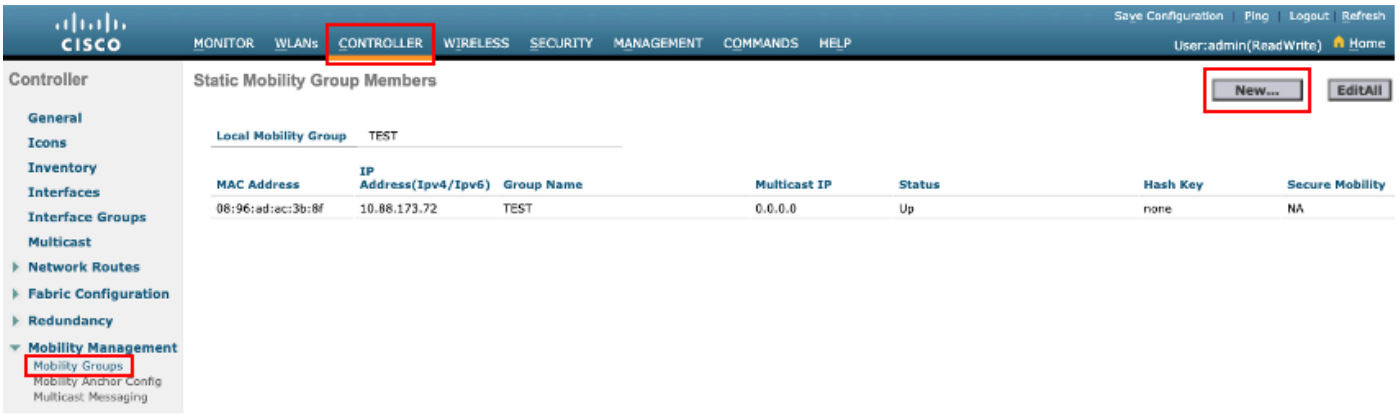
Passaggio 2. Raccogliere il valore Hash dal WLC 9800

```
# 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
```

Passaggio 3. Aggiungere le informazioni sul WLC di 9800 nel WLC di AireOS.

Dalla GUI:

Passa a **CONTROLLER > Mobility Management > Mobility Groups > New.**



Immettere i valori e fare clic su **Apply.**

> **Nota**: l'hash è richiesto solo nei casi in cui il modello 9800 utilizza un certificato autofirmato come il C9800-CL. I dispositivi hardware dispongono di un certificato SUDI e non richiedono un hash (ad esempio, 9800-40, 9800-L e così via).

Dalla CLI:

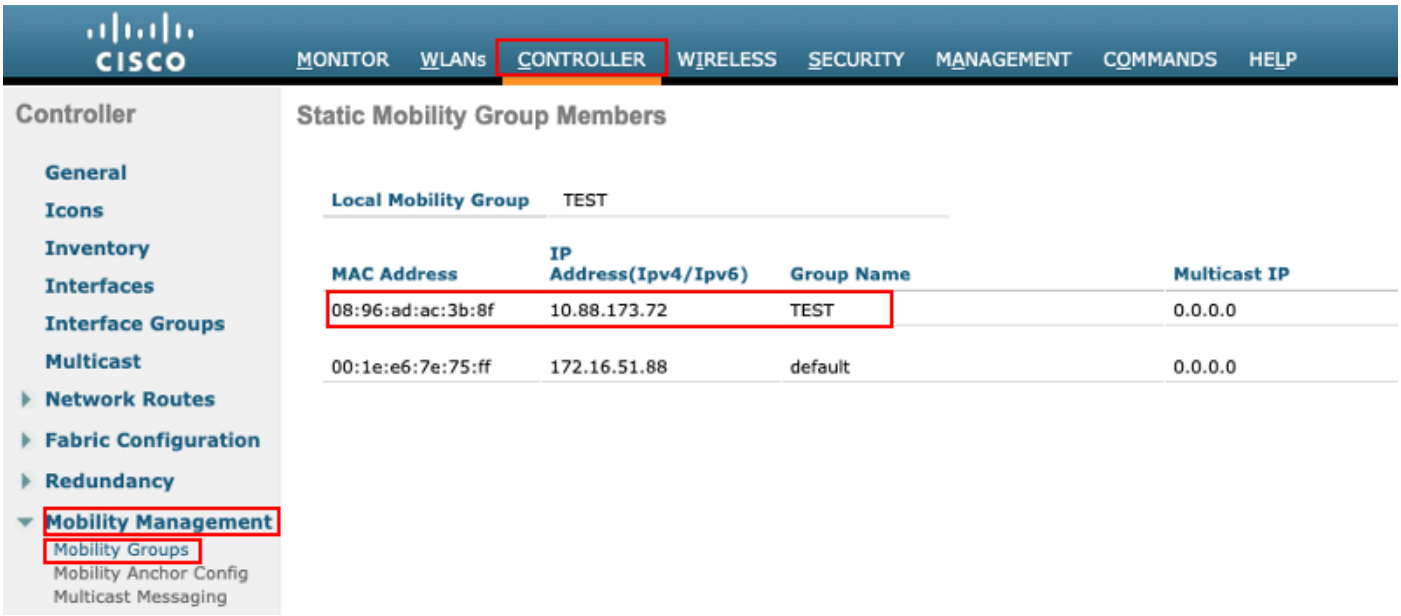
```
>config mobility group member add <9800 mac-address> <9800 WLC-IP> <group-name> encrypt enable
>config mobility group member hash <9800 WLC-IP> <9800 WLC-Hash>
>config mobility group member data-dtls <9800 mac-address> disable
```

Configurazione 9800 WLC

Passaggio 1. Raccogliere le informazioni sulla mobilità AireOS.

Dalla GUI:

Accedere alla GUI di AireOS e selezionare **CONTROLLER > Mobility Management > Mobility Groups** e annotare Indirizzo MAC, Indirizzo IP e Nome gruppo.



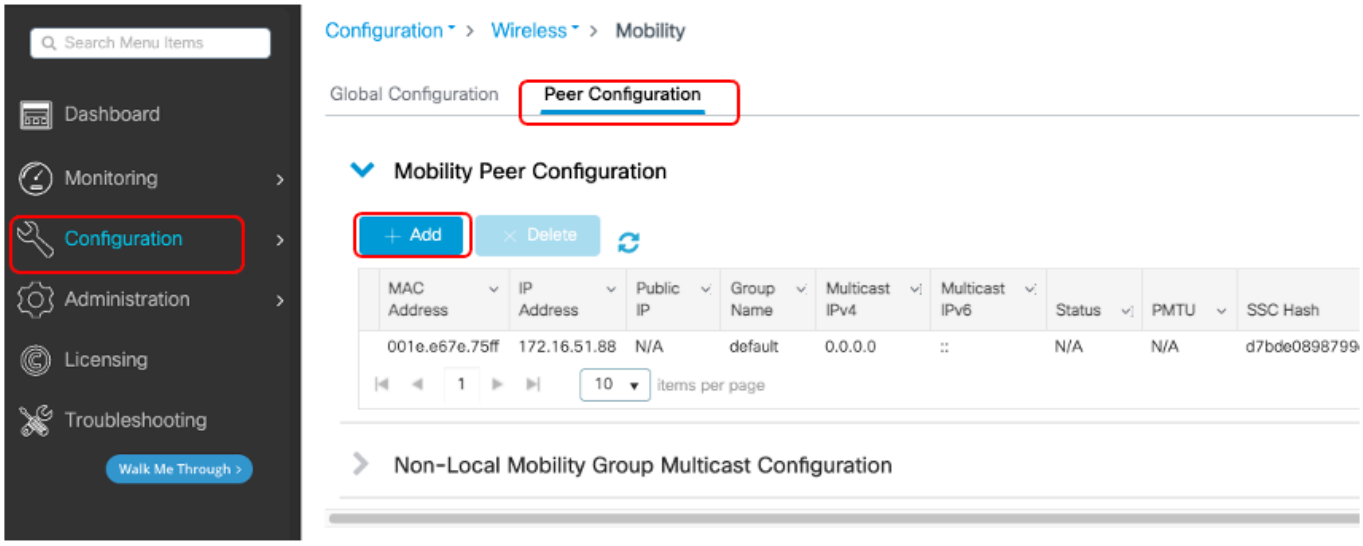
Dalla CLI:

```
>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
```

Passaggio 2. Aggiunta delle informazioni sul WLC di AireOS nel WLC 9800

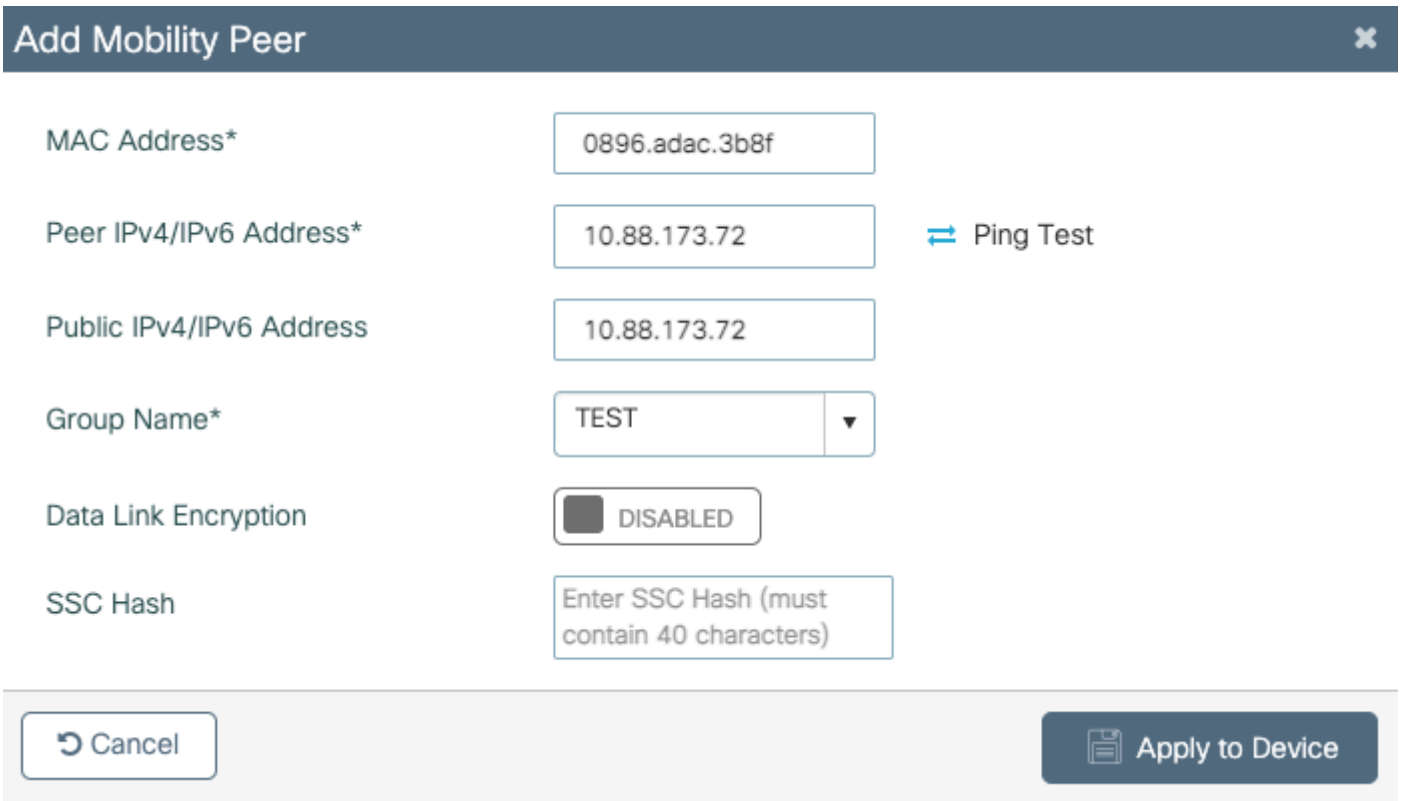
Dalla GUI:

Passa a **Configuration > Wireless > Mobility > Peer Configuration > Add**



Immettere le informazioni sul WLC di AireOS.

> **Nota**: sul WLC 9800, la crittografia del control plane è sempre abilitata, il che significa che è necessario avere una mobilità sicura abilitata sul lato AireOS. Tuttavia, la crittografia del collegamento dati è facoltativa. Se la si abilita sul lato 9800, abilitarla su AireOS con: **config mobility group member data-dtls enable**



Dalla CLI:

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

Verifica

Fare riferimento a questa sezione per verificare che la configurazione funzioni correttamente.

Verifica AireOS WLC

```
>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 Verification

```
#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
```

Risoluzione dei problemi

Le informazioni contenute in questa sezione permettono di risolvere i problemi relativi alla configurazione.

Per risolvere i problemi relativi all'implementazione del tunnel per la mobilità, utilizzare i seguenti comandi per eseguire il debug del processo:

AireOS WLC

Passaggio 1. Abilitare i debug relativi alla mobilità.

```
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
```

Passaggio 2. Riprodurre la configurazione e verificare l'output

Esempio di creazione riuscita di un tunnel per la mobilità su un WLC AirOS.

```
*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

Per impostazione predefinita, i controller 9800 registrano continuamente le informazioni di processo senza bisogno di una procedura di debug speciale.

Èsufficiente connettersi al controller e recuperare i registri associati a qualsiasi componente wireless per risolvere il problema.

I registri possono durare giorni, a seconda di quanto è occupato il controller.

Per semplificare l'analisi, tirare i log con un intervallo di tempo o per l'ultimo numero di minuti (l'ora predefinita è impostata su 10 minuti) ed è possibile filtrare in base agli indirizzi IP o MAC.

Passaggio 1. Controllare l'ora corrente del controller in modo da poter tenere traccia dei log nel tempo fino a quando si è verificato il problema.

```
# show clock
```

Passaggio 2. Raccogliere i log del controller, nel caso in cui vi siano informazioni a livello di Cisco IOS che potrebbero essere correlate al problema.

```
# show logging
```

Passaggio 3. Raccogliere le tracce del livello di avviso sempre attive per un indirizzo specifico. Per filtrare è possibile utilizzare l'indirizzo IP o MAC del peer mobile.

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

Questo comando genera registri per gli ultimi 10 minuti. È possibile regolare questo tempo con il comando **show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt**.

Èpossibile visualizzare il contenuto nella sessione o copiare il file in un server TFTP esterno.

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

or

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

Traccia attiva radio

Se i log sempre attivi non forniscono informazioni sufficienti per identificare i problemi che sono stati innescati durante la configurazione del tunnel, è possibile abilitare i debug condizionali e l'acquisizione **Radio Active (RA)** tracce, che forniscono un'attività di processo più dettagliata.

Passaggio 1. Verificare che non vi siano condizioni di debug già abilitate.

```
# show debugging
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop


IOSXE Packet Tracing Configs:




Packet Infra debugs:

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

Se viene visualizzata una condizione non correlata all'indirizzo che si desidera monitorare, disattivarla.

Per rimuovere un indirizzo specifico:

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

Per rimuovere tutte le condizioni (metodo consigliato):

```
# clear platform condition all
```

Passaggio 2. Aggiungere la condizione di debug per un indirizzo che si desidera monitorare.

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

> **Nota**: se si desidera monitorare più di un dispositivo peer mobile contemporaneamente, utilizzare un **debug platform condition feature wireless mac** per indirizzo MAC.

Passaggio 3. Avere il WLC 9800 per avviare il monitoraggio dell'attività dell'indirizzo specificata.

```
# debug platform condition start
```

Nota: l'output dell'attività di mobilità non viene visualizzato in quanto tutto viene memorizzato internamente nel buffer per essere raccolto in un secondo momento.

Passaggio 4. Riprodurre il problema o il comportamento che si desidera monitorare.

Passaggio 5. Interrompere i debug.

```
# debug platform condition stop
```

Passaggio 6. Raccoglie l'output dell'impegno di tipo indirizzo.

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

Questo comando genera registri per gli ultimi 10 minuti. È possibile regolare questo tempo con il comando **show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt**.

Èpossibile copiare **FILENAME.txt** su un server esterno o visualizzare l'output direttamente sullo schermo.

Copiare il file su un server esterno:

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

Visualizzare il contenuto:

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

Passaggio 7. Se non è ancora possibile individuare la causa di un errore, raccogliere il livello interno dei registri.

Non è necessario eseguire di nuovo il debug del client. Utilizzare i registri già archiviati internamente, ma raccoglierne una gamma più ampia).

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

Èpossibile copiare **FILENAME.txt** su un server esterno o visualizzare l'output direttamente sullo schermo.

Copiare il file su un server esterno:

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

Visualizzare il contenuto:

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

Passaggio 8. Rimuovere le condizioni di debug.

```
# clear platform condition all
```

> **Nota**: rimuovere sempre le condizioni di debug dopo una sessione di risoluzione dei problemi.

Esempio di creazione riuscita di un tunnel per la mobilità su un WLC 9800.

```
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

Nella maggior parte dei casi, è molto utile per controllare i pacchetti scambiati tra i WLC. È particolarmente utile filtrare le acquisizioni con **Access Control Lists (ACLs)** per limitare il traffico acquisito.

Questo è un modello di configurazione per le acquisizioni incorporate nella CLI.

Passaggio 1. Creare l'ACL di filtro:

```
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
```

Passaggio 2. Definire i parametri di acquisizione:

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

> **Nota**: selezionare l'interfaccia di gestione per il parametro INTERFACE_NAME

Passaggio 3. Avviare l'acquisizione:

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

Passaggio 4. Interrompere l'acquisizione:

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

Passaggio 5. Per raccogliere il file di acquisizione dei pacchetti, selezionare **Risoluzione dei problemi > Packet Capture** sulla GUI.

Scenari comuni di risoluzione dei problemi

Gli esempi seguenti sono i tunnel formati tra 9800 WLC.

Percorso dati e di controllo inattivo a causa di problemi di connettività

Abilita **Always-On-Logs** e **Embedded packet captures** per fornire ulteriori informazioni per la risoluzione dei problemi:

```
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
```

Le acquisizioni dei pacchetti sono utili per confermare il comportamento.

```
 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
```

Si noti che sia il debug sia il WLC mostrano che non è presente alcuna risposta ai ping dei dati o dei controlli. In uno scenario comune la connettività IP è consentita, ma le porte 1666 o 1667 non possono comunicare attraverso la rete.

Mancata corrispondenza della configurazione tra WLC

In questo caso abbiamo confermato la connettività per tutte le porte tra i WLC, ma continuiamo a notare la mancata corrispondenza dei pacchetti keepalive.

Abilita **Always-On-Logs** e **Embedded packet captures** per fornire ulteriori informazioni per la risoluzione dei problemi:

```
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
```

I registri interni sul peer 172.16.55.28 ci aiutano a confermare la mancata corrispondenza della configurazione

```
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
```

Configurazione non corrispondente comune: nome di gruppo errato, mancata corrispondenza in **Data Link Encryption** e un indirizzo MAC di mobilità non corretto.

Registro di gruppo non corrispondente:

```
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.
```

Registro indirizzi MAC non corrispondenti:

```
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.
```

Problemi di handshake DTLS

Questo tipo di problema è correlato agli stabilimenti del tunnel DTLS tra i WLC. È possibile che il percorso dati sia attivo ma il percorso di controllo rimane **DOWN**.

Abilita **Always-On-Logs** e **Embedded packet captures** per fornire ulteriori informazioni per la risoluzione dei problemi:

```
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
```

Utilizzo **show wireless management trustpoint** e **show crypto pki trustpoints commands** per verificare le informazioni sul certificato.

Lo scenario HA SSO

Se si dispone di controller nella coppia High Availability SSO, è importante essere informati. L'indirizzo MAC di mobilità non è configurato per impostazione predefinita e può causare l'interruzione del tunnel di mobilità in caso di failover.

Il **riepilogo mostra mobilità wireless** fornisce l'indirizzo MAC per la mobilità corrente in uso, ma non è necessariamente configurato. Verificare se per la configurazione è stato configurato l'indirizzo MAC per la mobilità con **show run | i mobilità**

Se il mac per la mobilità non è configurato nella configurazione in esecuzione, cambia in seguito al failover sul WLC in standby e ciò causa il malfunzionamento dei tunnel per la mobilità.

La soluzione semplice consiste nell'accedere alla pagina **Configurazione > Wireless > Mobility** Web UI e selezionare **Applica**. In questo modo l'attuale MAC per PC portatili viene salvato nella configurazione. Il MAC rimane quindi lo stesso anche dopo il failover e i tunnel di mobilità vengono mantenuti.

Questo problema si verifica principalmente se si esegue la configurazione della mobilità tramite la riga di comando e si dimentica di configurare l'indirizzo MAC della mobilità. L'interfaccia utente Web salva automaticamente un indirizzo MAC per dispositivi mobili quando si applicano le impostazioni.

Informazioni correlate

- Configurazione della funzione WLAN Anchor Mobility su Catalyst 9800
- Documentazione e supporto tecnico – Cisco Systems

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).