Configuration des topologies de mobilité sur les contrôleurs LAN sans fil (WLC) Catalyst 9800

Table des matières

Introduction

Ce document décrit les scénarios de configuration de mobilité qui couvrent les topologies entre les

contrôleurs LAN sans fil (WLC) Catalyst 9800 et les WLC AireOS.

Conditions préalables

Exigences

Cisco recommande de connaître les sujets suivants :

- Accès CLI ou GUI aux contrôleurs sans fil.

Composants utilisés

- **AireOS WLC** version 8.10 MR1 ou ultérieure. Vous pouvez également utiliser **Inter Release Controller Mobility (IRCM)** images spéciales 8.5
- WLC 9800, Cisco IOS® XE v17.3.4

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Configurer

Diagramme du réseau

Directives et restrictions

1. **Mobility Group** le nom du 9800 est « default » (par défaut).

Note:
1) Dans les cas où les WLC sont dans des sous-réseaux différents, assurez-vous que le port UDP 16666 et 16667 est ouvert entre eux.
2) Il est recommandé que les deux WLC 9800 exécutent la même version afin que les clients qui se déplacent à travers aient une expérience cohérente dans les scénarios d'ancrage d'invité et d'itinérance de couche 3.

Tunnel de mobilité entre deux WLC Catalyst 9800

Cet exemple de base décrit comment configurer la mobilité entre deux contrôleurs 9800. Il est

généralement utilisé pour l'accès invité (ancre) ou pour permettre aux clients de se déplacer sur les contrôleurs et de préserver l'identité du client.

Lorsque vous configurez la mobilité sur C9800, la première chose à choisir est le nom du groupe de mobilité. Le nom du groupe de mobilité prérempli est une valeur par défaut, mais vous pouvez le personnaliser selon une valeur souhaitée.

Vous devez configurer le même nom de groupe de mobilité sur les contrôleurs lorsqu'une couche2 rapide se déplace comme **Fast Transition (FT)** ou **Cisco Centralized Key Management (CCKM)** est en cours d'utilisation.

Par défaut, l'adresse MAC Ethernet de base du châssis, comme indiqué dans la **show version** est reflété sur l'interface utilisateur graphique pour l'adresse MAC de mobilité.

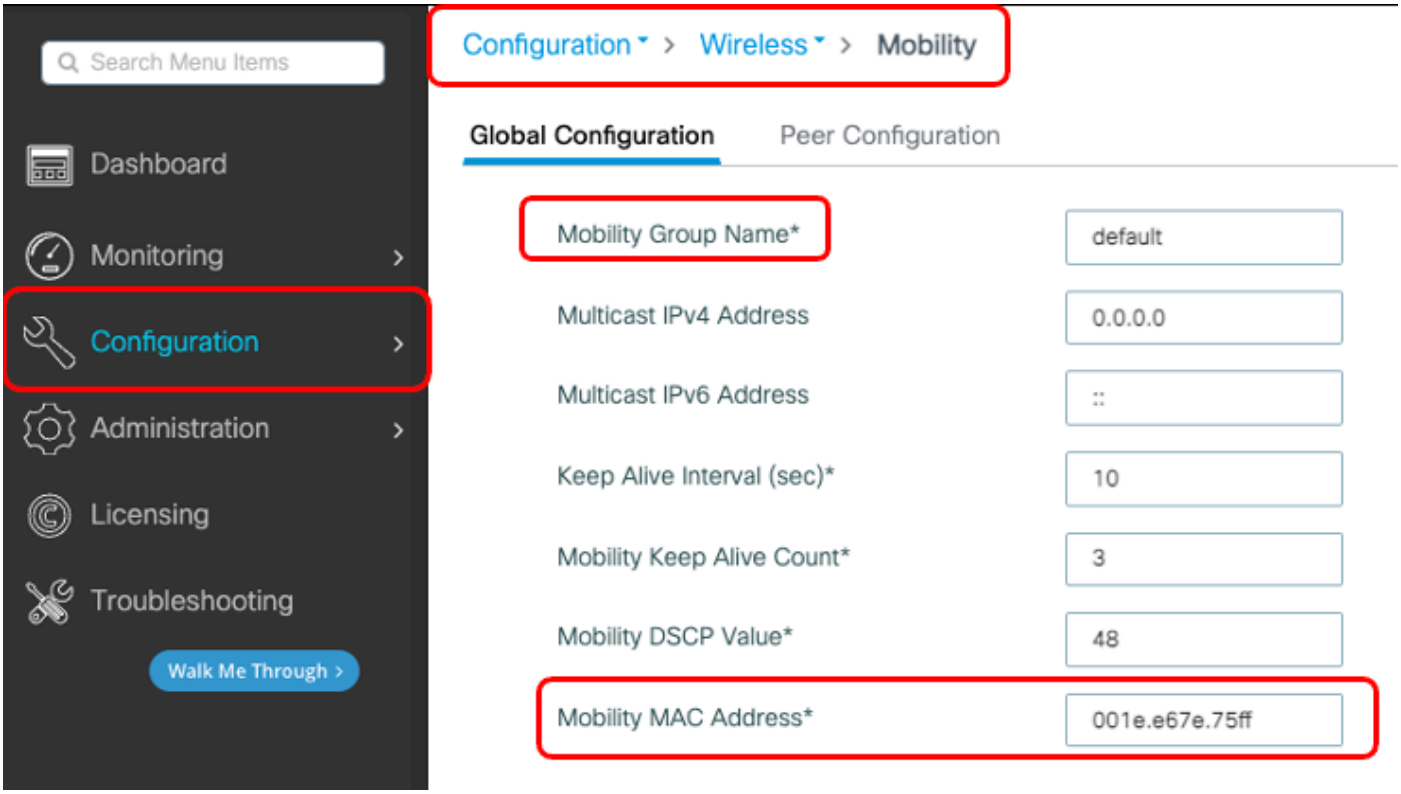
Sur l'interface de ligne de commande, par défaut, le mac de mobilité est 0000.000.000, comme indiqué dans **show run all | inc mobility mac-address**

Dans les cas où les 9800 sont associés pour **High Availability (HA) Stateful Switchover (SSO):**

Si la configuration est conservée par défaut et que l'adresse MAC du châssis est utilisée pour former le tunnel de mobilité, le châssis actif et le tunnel de mobilité échouent en cas de basculement.

Par conséquent, il est obligatoire de configurer une adresse MAC de mobilité pour la paire HA C9800.

Étape 1 : sur l'interface utilisateur graphique, accédez à **Configuration > Wireless > Mobility > Global Configuration.**



Via la CLI :

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

Étape 1. Collecter la configuration de mobilité des deux WLC 9800.

Pour les deux WLC 9800, accédez à **Configuration > Wireless > Mobility > Global Configuration** et prend acte de son **Mobility Group Name** et **Mobility MAC Address.**

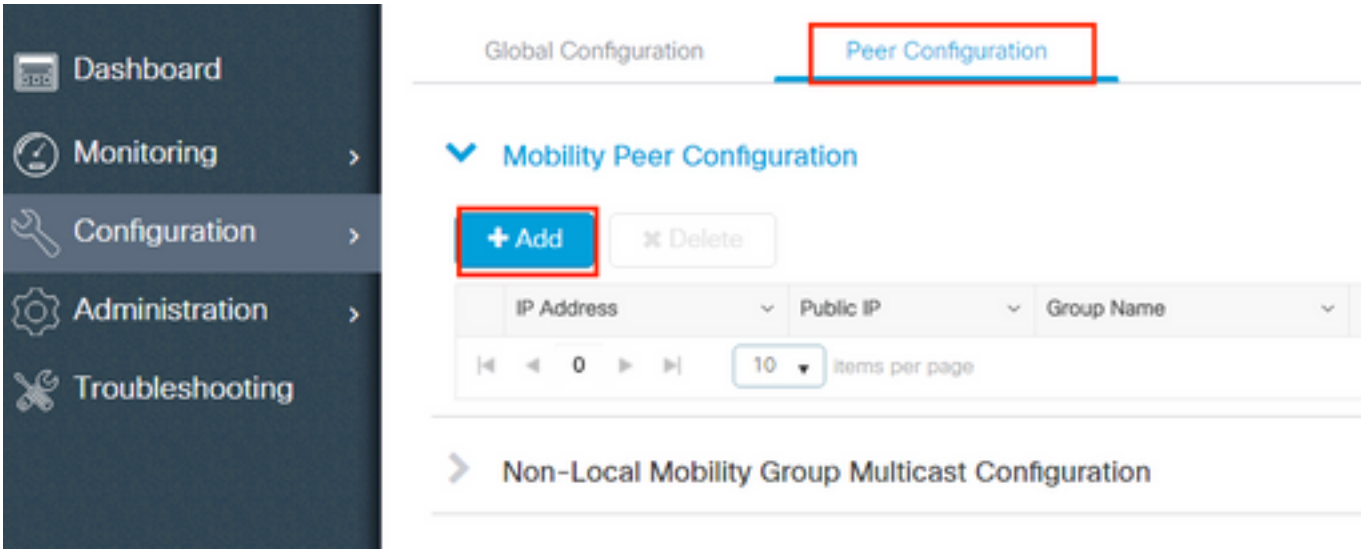
Via la CLI :

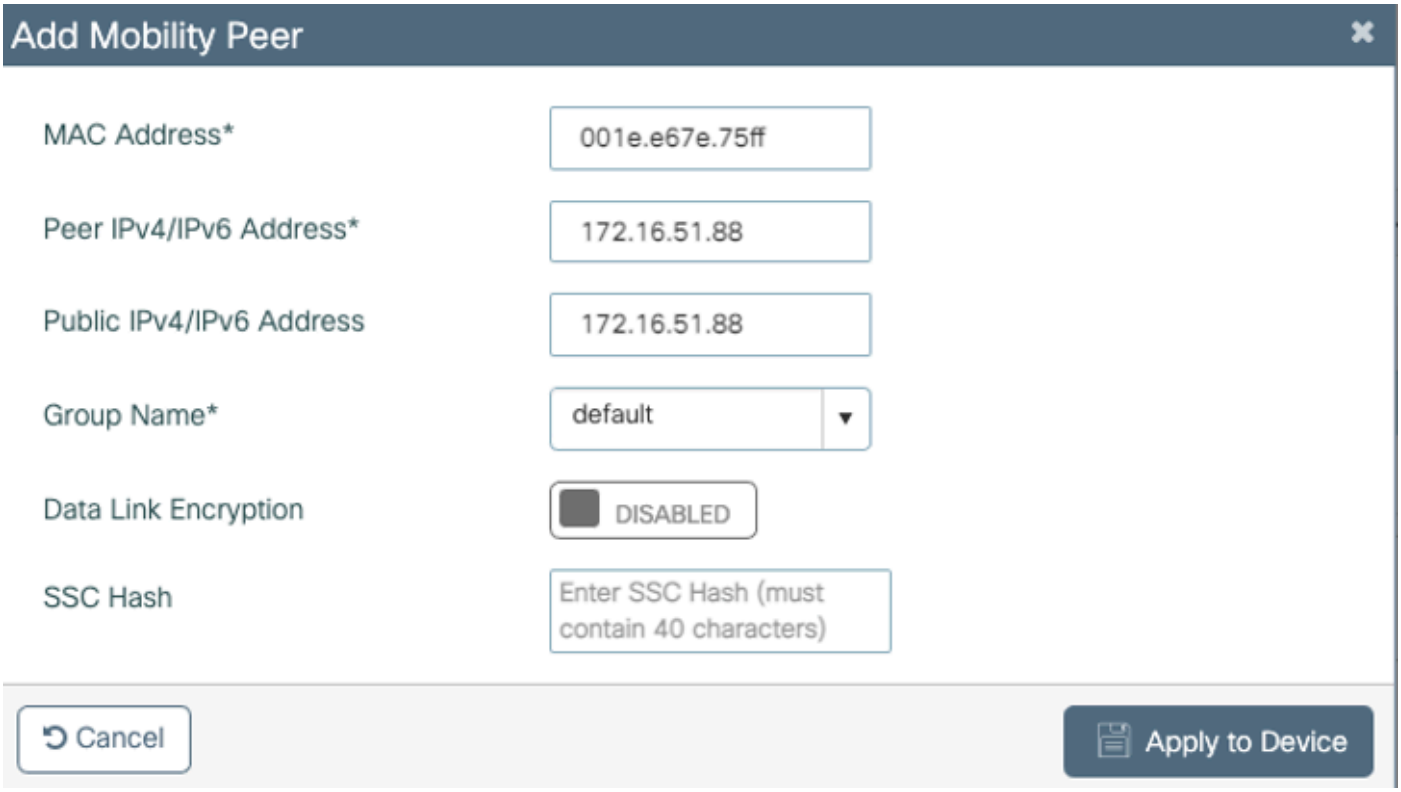
```
#show wireless mobility summary

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
```

Étape 2. Ajouter une configuration homologue

Naviguez jusqu'à **Configuration > Wireless > Mobility > Peer Configuration** et entrez les informations du contrôleur homologue. Faites la même chose pour les deux WLC 9800.

Via l'interface utilisateur graphique :





Via la CLI :

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

Remarque : vous pouvez éventuellement activer le cryptage de liaison de données.

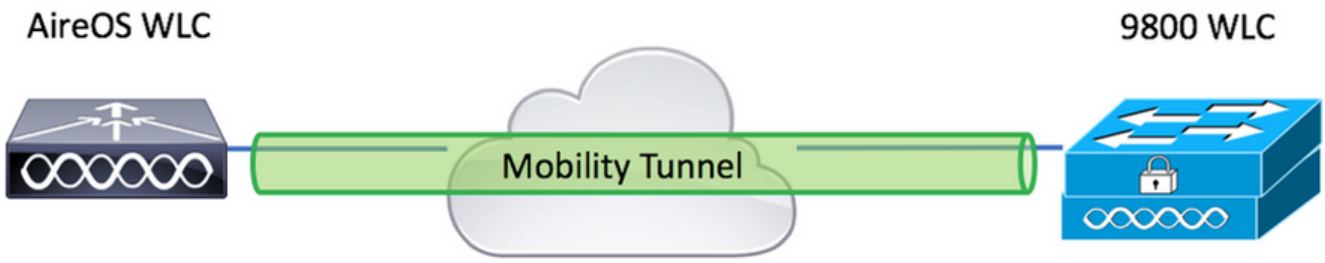
Tunnel de mobilité entre les contrôleurs AireOS WLC et 9800-CL

Ce scénario est normal pour **brownfield** ou lors de la migration du contrôleur, où nous divisons le réseau en une zone de points d'accès (AP) contrôlée par un contrôleur AireOS et une autre par un 9800.

Il est conseillé que les points d'accès soient répartis entre les contrôleurs par zone physique ou RF, de sorte que les clients ne se déplacent qu'entre les contrôleurs lorsqu'ils se déplacent.

Éviter **salt and pepper** déploiement. Cette topologie de mobilité peut également être utilisée pour **guest anchor** où 9800 agit en tant que contrôleur étranger et un AireOS en tant que contrôleur d'ancrage.

Diagramme du réseau



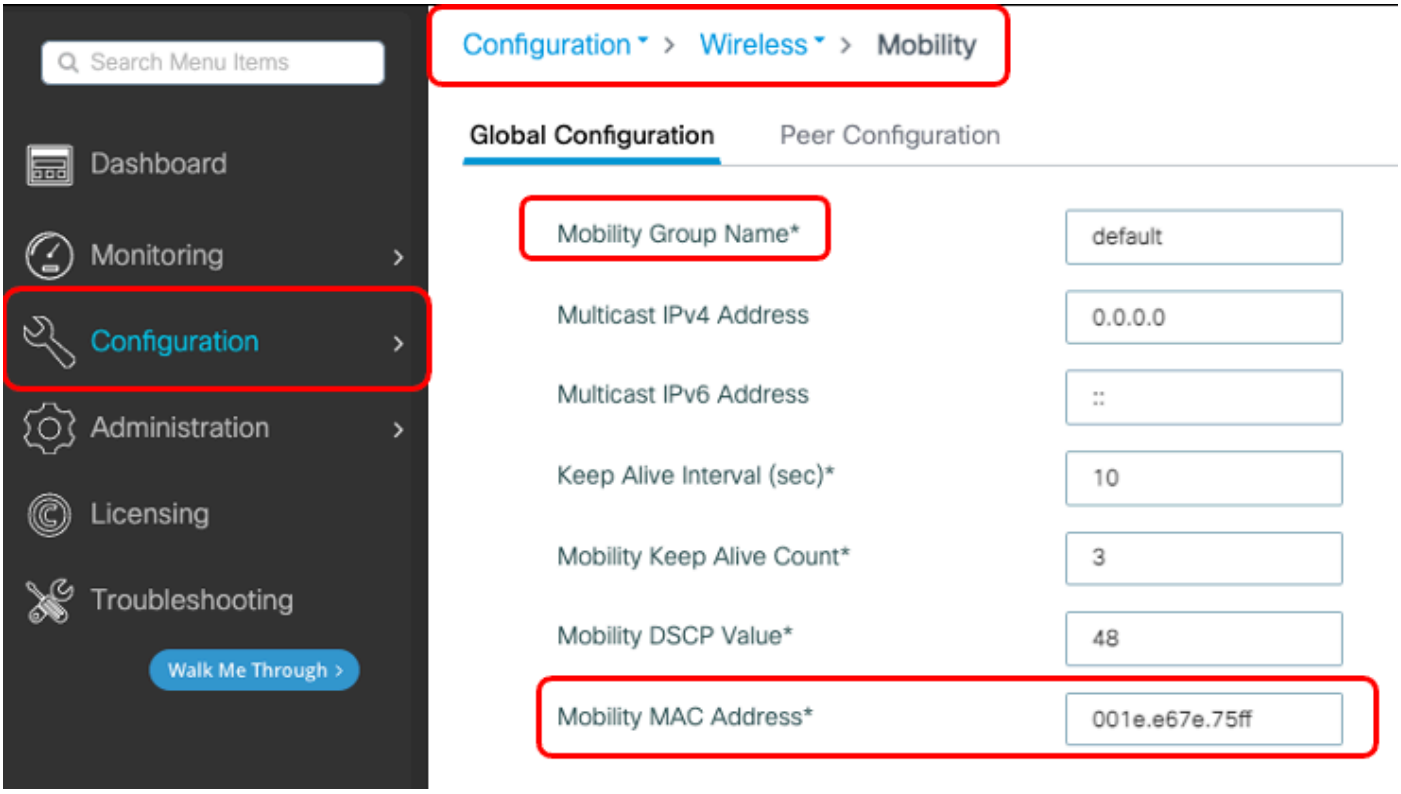
Configuration WLC AireOS

Si vos contrôleurs 9800 sont **High Availability**, vérifiez que vous avez configuré l'adresse MAC de mobilité.

Étape 1. Collecter les informations de mobilité du WLC 9800.

Via l'interface utilisateur graphique :

Naviguez jusqu'à **Configuration > Wireless > Mobility > Global Configuration** et prend acte de son **Mobility Group Name** et **Mobility MAC Address.**



Via la 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
```

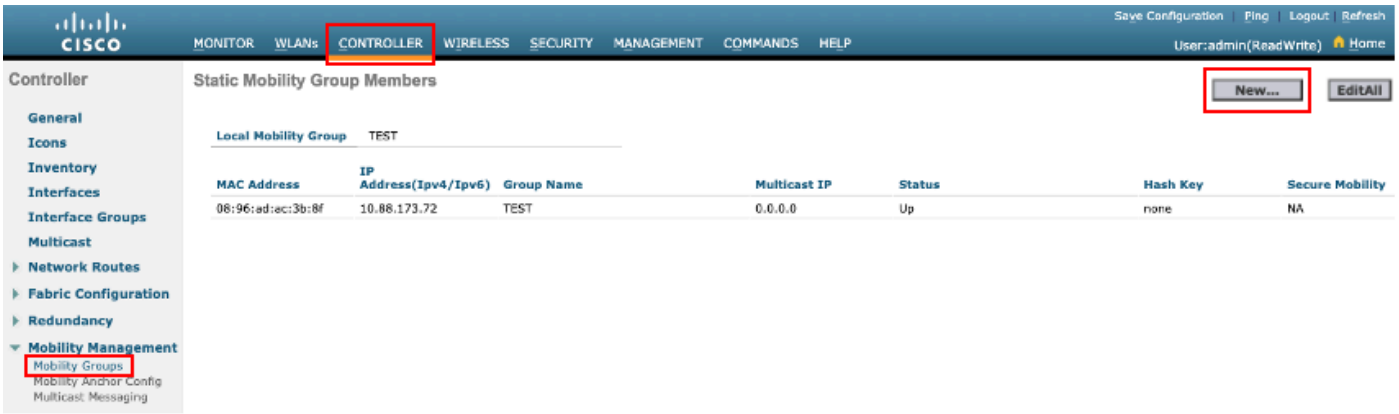
Étape 2. Collecter la valeur de hachage du WLC 9800

```
# 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
```

Étape 3. Ajoutez les informations du WLC 9800 dans le WLC AireOS.

Via l'interface utilisateur graphique :

Naviguez jusqu'à **CONTROLLER > Mobility Management > Mobility Groups > New.**



Entrez les valeurs et cliquez sur **Apply.**

> **Remarque** : le hachage n'est requis que dans les cas où le 9800 utilise un certificat auto-signé tel que le C9800-CL. Les appliances matérielles disposent d'un certificat SUDI et n'ont pas besoin d'un hachage (par exemple, un 9800-40, un 9800-L, etc.).

Via la CLI :

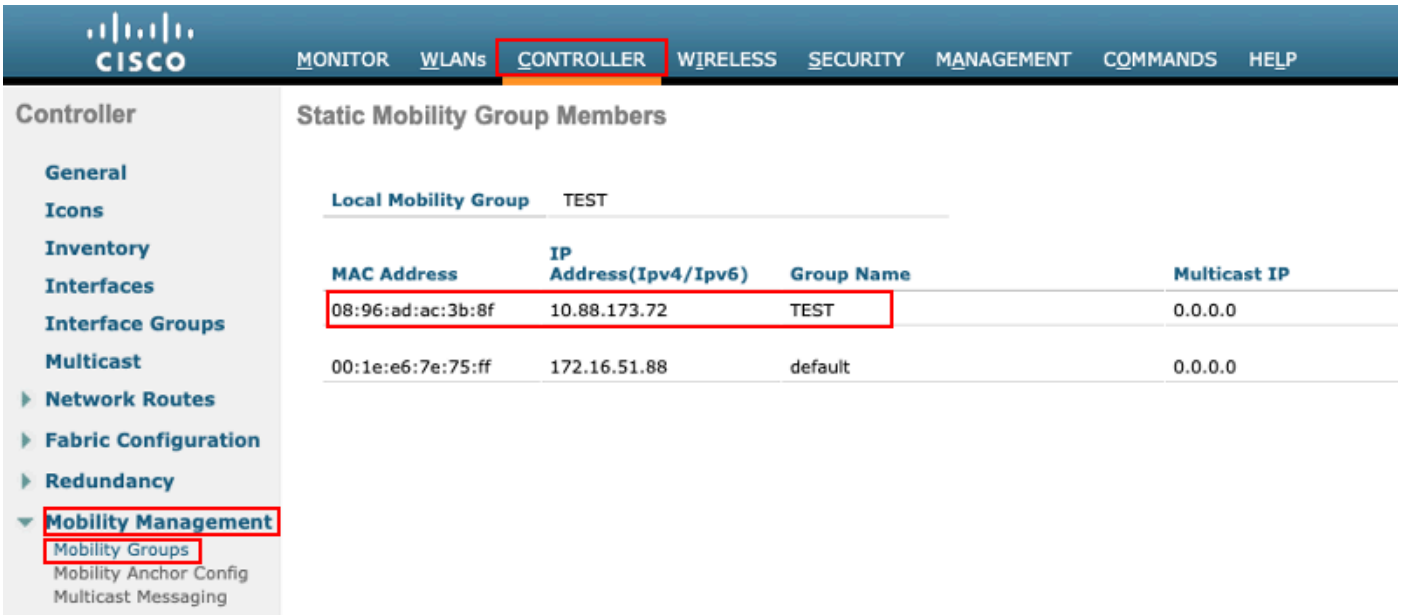
```
>config mobility group member add <9800 mac-address> <9800 WLC-IP> <group-name> encrypt enable
>config mobility group member hash <9800 WLC-IP> <9800 WLC-Hash>
>config mobility group member data-dtls <9800 mac-address> disable
```

Configuration du WLC 9800

Étape 1. Collecter des informations sur la mobilité AireOS.

Via l'interface utilisateur graphique :

Connectez-vous à l'interface AireOS et accédez à **CONTROLLER > Mobility Management > Mobility Groups** et prenez note de l'adresse MAC, de l'adresse IP et du nom du groupe.



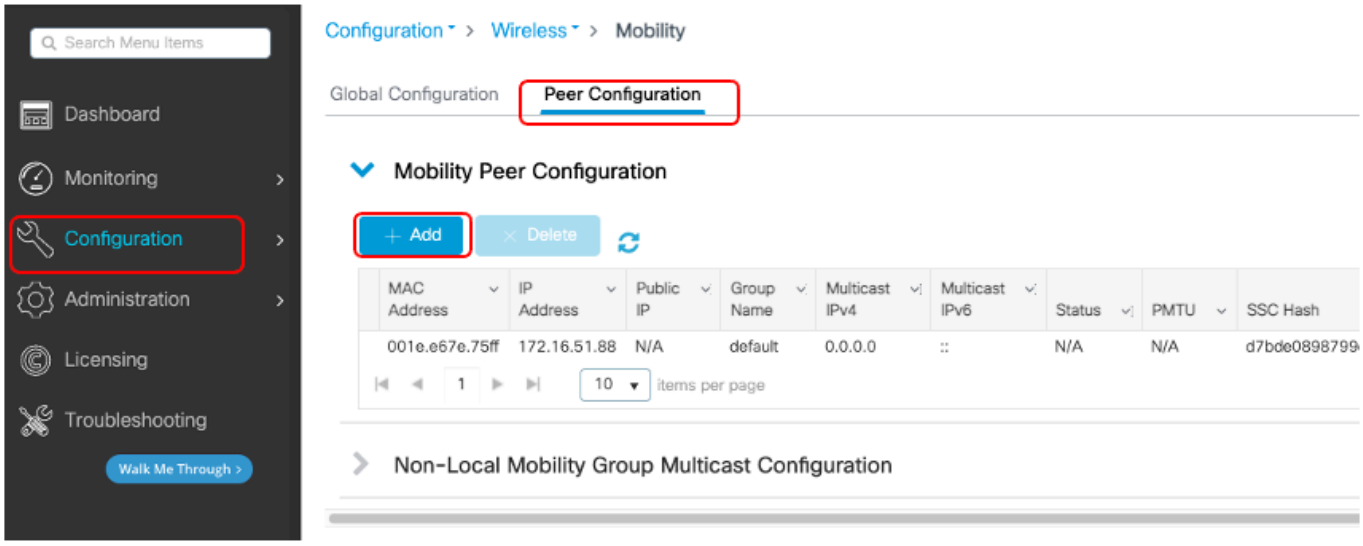
Via la CLI :

```
>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
```

Étape 2. Ajoutez les informations du WLC AireOS au WLC 9800

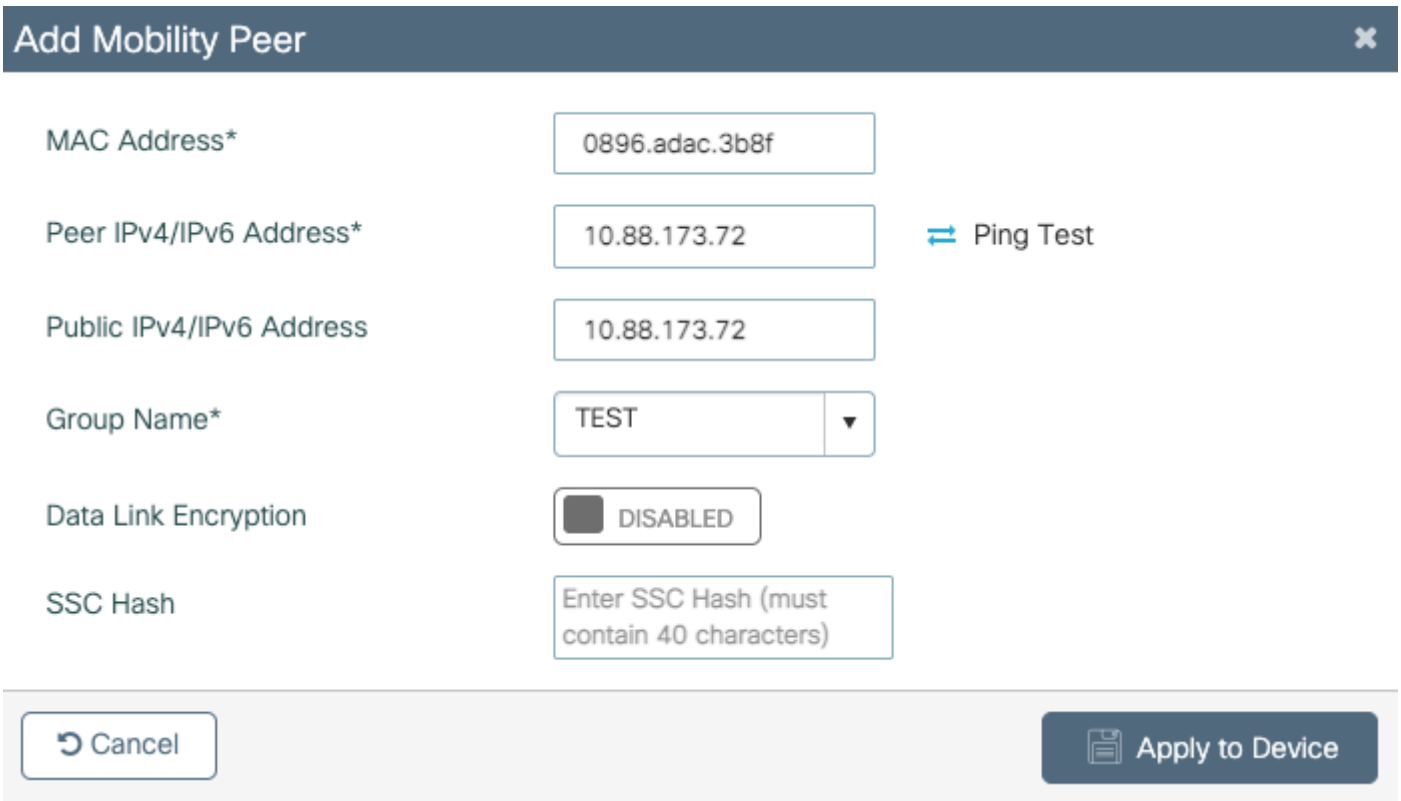
Via l'interface utilisateur graphique :

Naviguez jusqu'à **Configuration > Wireless > Mobility > Peer Configuration > Add**



Saisissez les informations du WLC AireOS.

> **Remarque** : sur le WLC 9800, le cryptage du plan de contrôle est toujours activé, ce qui signifie que vous devez avoir une mobilité sécurisée activée du côté d'AireOS. Cependant, le chiffrement de la liaison de données est facultatif. Si vous l'activez sur le côté 9800, activez-le sur AireOS avec : **config mobility group member data-dtls enable**



Via la CLI :

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

Vérifier

Utilisez cette section pour confirmer que votre configuration fonctionne correctement.

Vérification WLC AireOS

```
>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
```

Vérification WLC du Catalyst 9800

```
#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
```

Dépannage

Cette section fournit des informations permettant de dépanner votre configuration.

Pour dépanner l'implémentation du tunnel de mobilité, utilisez ces commandes pour déboguer le processus :

WLC AireOS

Étape 1. Activez les débogages de mobilité.

```
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
```

Étape 2. Reproduire la configuration et vérifier le résultat

Exemple de création réussie d'un tunnel de mobilité sur 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
```

WLC Catalyst 9800

Par défaut, les contrôleurs 9800 consignent en permanence les informations de processus sans avoir besoin d'une procédure de débogage spéciale.

Il vous suffit de vous connecter au contrôleur et de récupérer les journaux associés à tout composant sans fil à des fins de dépannage.

Les journaux peuvent s'étendre sur plusieurs jours ; cela dépend du niveau d'occupation du contrôleur.

Pour simplifier l'analyse, extrayez les journaux avec une plage de temps ou pour le dernier nombre de minutes (le temps par défaut est défini sur 10 minutes) et vous pouvez filtrer par adresses IP ou MAC.

Étape 1. Vérifiez l'heure actuelle sur le contrôleur afin de pouvoir suivre les journaux dans le temps jusqu'au moment où le problème s'est produit.

```
# show clock
```

Étape 2. Collectez les journaux du contrôleur, au cas où il y aurait des informations au niveau de Cisco IOS qui pourraient être liées au problème.

```
# show logging
```

Étape 3. Collecter les traces de niveau de notification toujours actives pour une adresse spécifique. Vous pouvez utiliser l'adresse IP ou MAC de l'homologue de mobilité pour filtrer.

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

Cette commande génère des journaux pour les 10 dernières minutes, il est possible d'ajuster cette durée avec la commande **show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt.**

Vous pouvez soit afficher le contenu de la session, soit copier le fichier sur un serveur TFTP

externe.

```
# 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

Si les journaux toujours actifs ne fournissent pas suffisamment d'informations pour savoir quels problèmes déclenchés lors de la configuration du tunnel, vous pouvez activer les débogages conditionnels et la capture **Radio Active (RA)** traces, qui donnent une activité de processus plus détaillée.

Étape 1. Vérifiez qu'aucune condition de débogage n'est déjà activée.

```
# show debugging
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop


IOSXE Packet Tracing Configs:




Packet Infra debugs:

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

Si vous voyez une condition qui n'est pas liée à l'adresse que vous voulez surveiller, désactivez-la.

Pour supprimer une adresse spécifique :

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

Pour supprimer toutes les conditions (méthode recommandée) :

```
# clear platform condition all
```

Étape 2. Ajoutez la condition de débogage pour une adresse que vous souhaitez surveiller.

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

> **Remarque** : si vous souhaitez surveiller plusieurs homologues de mobilité en même temps, utilisez un **debug platform condition feature wireless mac** par adresse MAC.

Étape 3. Demandez au WLC 9800 de démarrer la surveillance de l’activité d’adresse spécifiée.

```
# debug platform condition start
```

> **Remarque** : le résultat de l'activité de mobilité n'est pas affiché car tout est mis en mémoire tampon en interne pour être collecté ultérieurement.

Étape 4. Reproduisez le problème ou le comportement que vous souhaitez surveiller.

Étape 5. Arrêtez les débogages.

```
# debug platform condition stop
```

Étape 6. Collectez le résultat de l'activité d'adressage.

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

Cette commande génère des journaux pendant les 10 dernières minutes. Il est possible d'ajuster cette heure avec la commande **show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt**.

Vous pouvez soit copier le **FILENAME.txt** sur un serveur externe ou afficher le résultat directement à l'écran.

Copiez le fichier sur un serveur externe :

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

Affichez-en le contenu :

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

Étape 7. Si vous ne parvenez toujours pas à trouver la cause d'une défaillance, collectez le niveau interne des journaux.

(Vous n'avez pas besoin de déboguer à nouveau le client. Utilisez les journaux déjà stockés en interne, mais collectez un plus grand nombre d'entre eux).

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

Vous pouvez soit copier le **FILENAME.txt** sur un serveur externe ou afficher le résultat directement à l'écran.

Copiez le fichier sur un serveur externe :

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

Affichez-en le contenu :

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

Étape 8. Supprimez les conditions de débogage.

```
# clear platform condition all
```

Remarque : supprimez toujours les conditions de débogage après une session de dépannage.

Exemple de création réussie d'un tunnel de mobilité sur 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

Capture de paquets intégrée

La plupart du temps, il est très utile de vérifier les paquets échangés entre les WLC. Il est particulièrement utile de filtrer les captures avec **Access Control Lists (ACLs)** afin de limiter le trafic capturé.

Il s'agit d'un modèle de configuration pour les captures intégrées sur CLI.

Étape 1. Créez la liste de contrôle d'accès filtre :

```
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
```

Étape 2. Définissez les paramètres de capture :

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

Remarque : sélectionnez l'interface de gestion pour le paramètre INTERFACE_NAME

Étape 3. Démarrez la capture :

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

Étape 4. Arrêtez la capture :

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

Étape 5. Accédez à **Troubleshooting > Packet Capture** sur GUI pour collecter le fichier de capture de paquets.

Scénarios de dépannage courants

Les exemples suivants consistent en des tunnels formés entre 9800 WLC.

Contrôle et chemin des données en panne en raison de problèmes de connectivité

Activer **Always-On-Logs** et **Embedded packet captures** pour fournir des informations supplémentaires de dépannage :

```
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
```

Les captures de paquets sont utiles pour confirmer le comportement.

```
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
```

Notez que les commandes debug et WLC montrent qu'il n'y a pas de réponse aux requêtes ping Control ou Data. Un scénario courant indique que la connectivité IP est autorisée, mais que les ports 16666 ou 16667 ne sont pas autorisés à communiquer sur le réseau.

Non-concordance de configuration entre les WLC

Dans ce cas, nous avons confirmé la connectivité pour tous les ports entre les WLC, mais continuez à remarquer que les keepalives manquent.

Activer **Always-On-Logs** et **Embedded packet captures** pour fournir des informations supplémentaires de dépannage :

```
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
```

Les journaux internes sur l'homologue 172.16.55.28 nous aident à confirmer la non-correspondance de configuration

```
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
```

Incompatibilité de configuration courante : nom de groupe incorrect, incompatibilité sur **Data Link Encryption** et adresse MAC de mobilité incorrecte.

Journal des incohérences de groupe :

```
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.
```

Journal de non-concordance des adresses MAC :

```
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.
```

Problèmes de connexion DTLS

Ce type de problème est lié aux établissements de tunnel DTLS entre les WLC. Il peut arriver que le chemin de données soit UP mais que le chemin de contrôle reste **DOWN**.

Activer **Always-On-Logs** et **Embedded packet captures** pour fournir des informations supplémentaires de dépannage :

```
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
```

Utilisation **show wireless management trustpoint** et **show crypto pki trustpoints commands** pour vérifier vos informations de certificat.

Scénario HA SSO

Si vous avez des contrôleurs dans la paire SSO haute disponibilité, il y a un accrochage important à savoir. L'adresse MAC de mobilité n'est pas configurée par défaut et peut entraîner la désactivation du tunnel de mobilité en cas de basculement.

Le **résumé show wireless mobility** vous donne l'adresse MAC de mobilité actuelle utilisée, mais elle n'est pas nécessairement configurée. Vérifiez si l'adresse MAC de mobilité est configurée avec **show run** dans la configuration **| i mobilité**

Si le MAC de mobilité n'est pas configuré dans la configuration en cours, il change lors du basculement vers le WLC de secours et cela entraîne l'échec des tunnels de mobilité.

La solution la plus simple consiste à accéder à la page Web **Configuration > Wireless > Mobility** UI et à cliquer sur **apply**. Cette opération enregistre l'adresse MAC de mobilité actuelle dans la configuration. L'adresse MAC reste la même lors du basculement et les tunnels de mobilité sont préservés.

Ce problème se produit principalement si vous effectuez votre configuration de mobilité via la ligne de commande et que vous oubliez de configurer l'adresse MAC de mobilité. L'interface utilisateur Web enregistre automatiquement une adresse MAC de mobilité lorsque vous appliquez les paramètres.

Informations connexes

- Configuration de la fonction WLAN Anchor Mobility sur Catalyst 9800
- Technical Support & Documentation - Cisco Systems

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.