

在Catalyst 9800無線LAN控制器(WLC)上設定行動拓撲

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[設定](#)

[網路圖表](#)

[准則和限制](#)

[兩個Catalyst 9800 WLC之間的行動通道](#)

[—](#)
[步驟1.收集兩個9800 WLC的行動化組態。](#)

[步驟2.新增對等配置](#)

[AireOS WLC和9800-CL控制器之間的移動隧道](#)

[網路圖表](#)

[AireOS WLC配置](#)

[步驟1.收集9800 WLC行動資訊。](#)

[步驟2.從9800 WLC收集雜湊值](#)

[步驟3.將9800 WLC資訊新增到AireOS WLC中。](#)

[9800 WLC組態](#)

[步驟1.收集AireOS移動性資訊。](#)

[步驟2.將AireOS WLC資訊新增到9800 WLC中](#)

[驗證](#)

[AireOS WLC驗證](#)

[Catalyst 9800 WLC驗證](#)

[疑難排解](#)

[AireOS WLC](#)

[Catalyst 9800 WLC](#)

[無線電活動跟蹤](#)

[嵌入式封包擷取](#)

[常見故障排除場景](#)

[由於連線問題導致的控制和資料路徑中斷](#)

[WLC之間的組態不相符](#)

[DTLS握手問題](#)

[HA SSO案例](#)

[相關資訊](#)

簡介

本檔案介紹涵蓋Catalyst 9800無線LAN控制器(WLC)和AireOS WLC之間拓撲的行動化組態案例。

必要條件

需求

思科建議瞭解以下主題：

- 對無線控制器的CLI或GUI訪問。

採用元件

- AireOS WLC 8.10 MR1或更高版本。您還可以使用 Inter Release Controller Mobility (IRCM) 特殊的8.5形象
- 9800 WLC , Cisco IOS® XE v17.3.4

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

設定

網路圖表



准則和限制

1. **Mobility Group** 現成的9800的名稱是「default」。

附註：

- 1)如果WLC位於不同的子網中，請確保它們之間的埠UDP 16666和16667處於開啟狀態。
- 2)建議兩個9800 WLC執行相同的版本，以使漫遊的使用者端在第3層漫遊和訪客錨點情況下都具有一致的體驗。

兩個Catalyst 9800 WLC之間的行動通道

以下基本範例說明如何在兩個9800控制器之間設定行動化。這通常用於訪客接入（錨點），或允許客戶端在控制器間漫遊並保持客戶端身份。

在C9800上配置移動性時，首先要選擇移動性組名稱。預填充的移動組名稱是預設值，但您可以將其自定義為所需的值。

當快速第2層漫遊時，必須在控制器間配置相同的移動組名稱 **Fast Transition (FT)** 或 **Cisco Centralized Key Management (CCKM)** 正在使用中。

預設情況下，機箱的基本乙太網MAC地址，如所示 `show version` 在GUI上反映移動MAC地址。

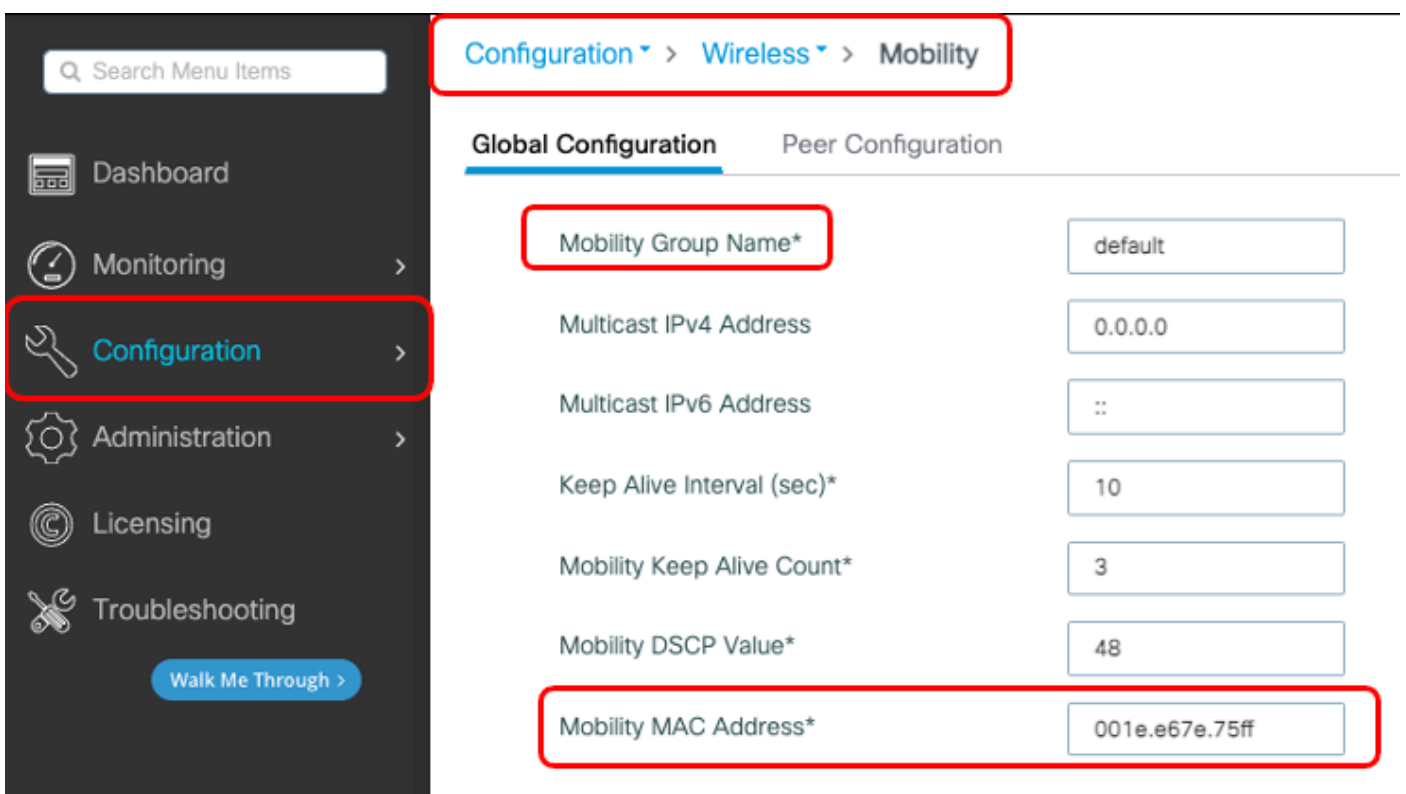
預設情況下，在CLI上，移動mac為0000.0000.000，如中所示 `show run all | inc mobility mac-address`

如果配對了9800個 **High Availability (HA) Stateful Switchover (SSO)**:

如果配置保留為預設值，並且機箱MAC地址用於形成移動隧道，則發生故障轉移時，主用機箱和移動隧道將失敗。

因此，需要為C9800 HA對配置移動MAC地址。

第1步：在GUI上，導航至 **Configuration > Wireless > Mobility > Global Configuration**.



通過CLI:

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

步驟1.收集兩個9800 WLC的行動化組態。

對於兩個9800 WLC，導覽至 **Configuration > Wireless > Mobility > Global Configuration** 並注意其 **Mobility Group Name** 和 **Mobility MAC Address**.

通過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
```

步驟2.新增對等配置

導航至 **Configuration > Wireless > Mobility > Peer Configuration** 並輸入對等控制器資訊。對兩個9800 WLC執行相同操作。

通過GUI:

The screenshot shows the Cisco GUI interface. On the left is a dark sidebar with navigation options: Dashboard, Monitoring, Configuration (highlighted), Administration, and Troubleshooting. The main content area has two tabs: 'Global Configuration' and 'Peer Configuration' (highlighted with a red box). Below the tabs is a section titled 'Mobility Peer Configuration' with a blue dropdown arrow. Under this section, there are two buttons: '+ Add' (highlighted with a red box) and '✕ Delete'. Below the buttons is a table with columns: 'IP Address', 'Public IP', and 'Group Name'. At the bottom of the table area, there is a pagination control showing '0' items and '10 items per page'. Below the table area is a link for 'Non-Local Mobility Group Multicast Configuration'.

Add Mobility Peer ✕

MAC Address*	<input type="text" value="001e.e67e.75ff"/>
Peer IPv4/IPv6 Address*	<input type="text" value="172.16.51.88"/>
Public IPv4/IPv6 Address	<input type="text" value="172.16.51.88"/>
Group Name*	<input type="text" value="default"/> ▼
Data Link Encryption	<input type="checkbox"/> DISABLED
SSC Hash	<input type="text" value="Enter SSC Hash (must contain 40 characters)"/>

通過CLI:

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

注意：（可選）您可以啟用資料鏈路加密。

AireOS WLC和9800-CL控制器之間的移動隧道

此案例對於以下專案是正常的 **brownfield** 在部署或控制器遷移期間，我們將網路劃分到由AireOS控制器控制的接入點區域(AP)中，另一個由9800控制。

建議按物理或RF區域在控制器之間分配AP，以便客戶端在控制器之間移動時僅在控制器之間漫遊。

避免 **salt and pepper** 部署. 或者，此移動拓撲還可用於 **guest anchor** 其中9800是外部控制器，AireOS是錨點控制器。

網路圖表



AireOS WLC配置

如果您的9800控制器位於 High Availability中，確保已配置移動MAC地址。

步驟1.收集9800 WLC行動資訊。

通過GUI:

導航至 Configuration > Wireless > Mobility > Global Configuration 並注意其 Mobility Group Name 和 Mobility MAC Address.

Configuration > Wireless > Mobility

Global Configuration Peer Configuration

Mobility Group Name*	default
Multicast IPv4 Address	0.0.0.0
Multicast IPv6 Address	::
Keep Alive Interval (sec)*	10
Mobility Keep Alive Count*	3
Mobility DSCP Value*	48
Mobility MAC Address*	001e.e67e.75ff

通過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
```

步驟2.從9800 WLC收集雜湊值

```
# show wireless management trustpoint
```

```
Trustpoint Name : Jay-9800_WLC_TP
```

Certificate Info : Available

Certificate Type : SSC

Certificate Hash : d7bde0898799dbfeffd4859108727d3372d3a63d

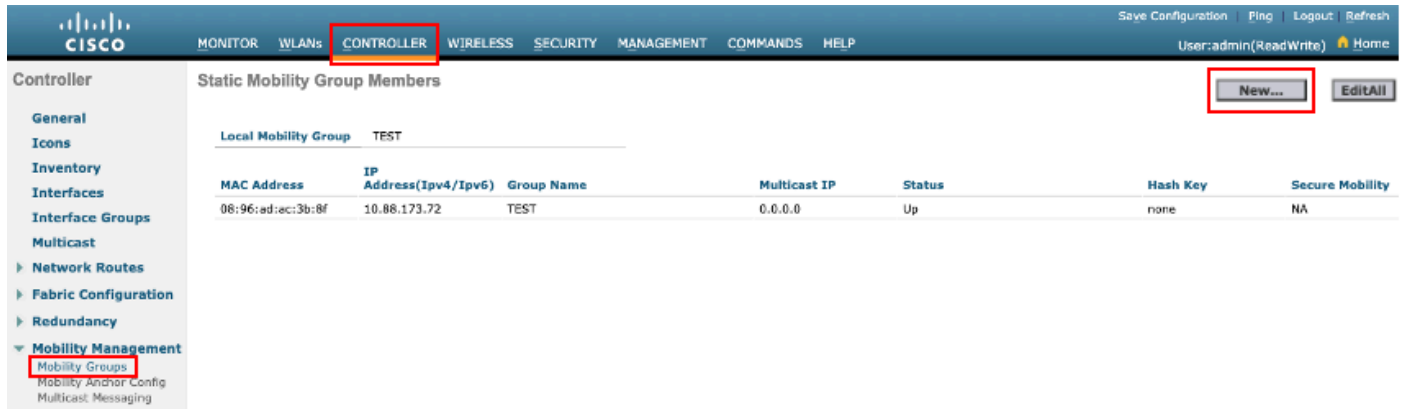
Private key Info : Available

FIPS suitability : Not Applicable

步驟3.將9800 WLC資訊新增到AireOS WLC中。

通過GUI:

導航至 **CONTROLLER > Mobility Management > Mobility Groups > New.**



輸入值，然後按一下 **Apply.**



註：僅當9800使用自簽名證書（如C9800-CL）時需要雜湊。硬體裝置具有SUDI證書，不需要雜湊（例如9800-40、9800-L等）。

通過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
```

9800 WLC組態

步驟1.收集AireOS移動性資訊。

通過GUI:

登入到AireOS GUI並導航至 **CONTROLLER > Mobility Management > Mobility Groups** 並記下MAC地址、IP地址和組名。

Local Mobility Group	TEST		
MAC Address	IP Address(Ipv4/Ipv6)	Group Name	Multicast IP
08:96:ad:ac:3b:8f	10.88.173.72	TEST	0.0.0.0
00:1e:e6:7e:75:ff	172.16.51.88	default	0.0.0.0

通過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
08:96:ad:ac:3b:8f	10.88.173.72	TEST	0.0.0.0

Up

步驟2.將AireOS WLC資訊新增到9800 WLC中

通過GUI:

導航至 **Configuration > Wireless > Mobility > Peer Configuration > Add**

Configuration > Wireless > Mobility

Global Configuration **Peer Configuration**

▼ Mobility Peer Configuration

+ Add **× Delete**

MAC Address	IP Address	Public IP	Group Name	Multicast IPv4	Multicast IPv6	Status	PMTU	SSC Hash
001e.e67e.75ff	172.16.51.88	N/A	default	0.0.0.0	::	N/A	N/A	d7bde0898799

1 10 items per page

► Non-Local Mobility Group Multicast Configuration

輸入AireOS WLC資訊。

註：在9800 WLC上，控制平面加密始終處於啟用狀態，這意味著您需要在AireOS端啟用安全移動。但是，資料鏈路加密是可選的。如果在9800端啟用它，請在AireOS上使用 `config mobility group member data-dtls enable` 啟用它

Add Mobility Peer ✕

MAC Address*

Peer IPv4/IPv6 Address* ⇄ Ping Test

Public IPv4/IPv6 Address

Group Name* ▼

Data Link Encryption DISABLED

SSC Hash

通過CLI:

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

驗證

使用本節內容，確認您的組態是否正常運作。

AireOS WLC驗證

```
>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	Status	Group Name
Multicast IP			
00:1e:e6:7e:75:ff	172.16.51.88	Up	default
0.0.0.0			
08:96:ad:ac:3b:8f	10.88.173.72	Up	TEST
0.0.0.0			

Catalyst 9800 WLC驗證

```
#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 IPv6	Public Ip	Group Name Status	Multicast IPv4 PMTU	Multicast
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			

疑難排解

本節提供的資訊用於對組態進行疑難排解。

要對移動隧道實施進行故障排除，請使用以下命令對過程進行調試：

AireOS WLC

步驟1.啟用移動調試。

```
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
```

步驟2.重現配置並檢驗輸出

在AirOS WLC上成功建立行動通道範例。

```
*capwapPingSocketTask: Feb 07 09:53:38.507: Client initiating connection on 172.16.0.5:16667 <-> 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.507: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.508: Received DTLS packet from mobility peer 172.16.0.21 bytes: 48
*capwapPingSocketTask: Feb 07 09:53:38.508: mm_dtls2_process_data_rcv_msg:1207 rcvBufLen 48 clr_pkt_len 2048 peer ac100015
*capwapPingSocketTask: Feb 07 09:53:38.508: Record      : type=22, epoch=0, seq=0
*capwapPingSocketTask: Feb 07 09:53:38.508:      Hndshk : type=3, len=23 seq=0, frag_off=0, frag_len=23
*capwapPingSocketTask: Feb 07 09:53:38.508: Handshake in progress for link 172.16.0.5:16667 <-> 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.508: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.508: DTLS consumed packet from mobility peer 172.16.0.21 bytes: 48
!
!<--output-omited-->
!
*capwapPingSocketTask: Feb 07 09:53:38.511: dtls2_cert_verify_callback: Forcing Certificate validation as success
*capwapPingSocketTask: Feb 07 09:53:38.511: Peer certificate verified.
*capwapPingSocketTask: Feb 07 09:53:38.511: Handshake in progress for link 172.16.0.5:16667 <-> 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.511: Nothing to send on link 172.16.0.5:16667 <-> 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.511: DTLS consumed packet from mobility peer 172.16.0.21 bytes: 503
*capwapPingSocketTask: Feb 07 09:53:38.511: Received DTLS packet from mobility peer 172.16.0.21 bytes: 56
*capwapPingSocketTask: Feb 07 09:53:38.511: mm_dtls2_process_data_rcv_msg:1207 rcvBufLen 56 clr_pkt_len 2048 peer ac100015
*capwapPingSocketTask: Feb 07 09:53:38.511: Record      : type=22, epoch=0, seq=6
*capwapPingSocketTask: Feb 07 09:53:38.511:      Hndshk : type=13, len=6 seq=3, frag_off=0, frag_len=6
*capwapPingSocketTask: Feb 07 09:53:38.523: Handshake in progress for link 172.16.0.5:16667 <-> 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.523: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.524: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.524: Sending packet to 172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.524: DTLS consumed packet from mobility peer 172.16.0.21 bytes: 56
*capwapPingSocketTask: Feb 07 09:53:38.527: Received DTLS packet from mobility peer 172.16.0.21 bytes: 91
*capwapPingSocketTask: Feb 07 09:53:38.527: mm_dtls2_process_data_rcv_msg:1207 rcvBufLen 91 clr_pkt_len 2048 peer ac100015
```

```
*capwapPingSocketTask: Feb 07 09:53:38.527: Record      : type=20, epoch=0, seq=8
*capwapPingSocketTask: Feb 07 09:53:38.527: Connection established for link 172.16.0.5:16667 <->
172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.527: ciperspec 1
*capwapPingSocketTask: Feb 07 09:53:38.527: Nothing to send on link 172.16.0.5:16667 <->
172.16.0.21:16667
*capwapPingSocketTask: Feb 07 09:53:38.527: DTLS consumed packet from mobility peer 172.16.0.21
bytes: 91
*mmMobility: Feb 07 09:53:38.527: DTLS Action Result message received
*mmMobility: Feb 07 09:53:38.527: Key plumb succeeded
*mmMobility: Feb 07 09:53:38.527: mm_dtls2_callback: Connection established with
172.16.0.21:16667
*mmMobility: Feb 07 09:53:38.527: mm_dtls2_db_status_up:895 Connections status up for entry
172.16.0.21:16667
*mmMobility: Feb 07 09:53:38.527: mm_dtls2_callback: DTLS Connection established with
172.16.0.21:16667, Sending update msg to mobility HB
```

Catalyst 9800 WLC

預設情況下，9800控制器會持續記錄程式資訊，不需要任何特殊偵錯程式。

只需連線到控制器，並檢索與任何無線元件相關的日誌即可進行故障排除。

日誌可以跨越數天；這取決於控制器有多忙。

為簡化分析，請獲取時間範圍或最後分鐘數的日誌（預設時間設定為10分鐘），您可以按IP或MAC地址進行過濾。

步驟1.檢查控制器上的當前時間，這樣您就可以跟蹤問題發生時之前的日誌。

```
# show clock
```

步驟2.收集控制器日誌，以防Cisco IOS級別出現可能與問題相關的任何資訊。

```
# show logging
```

步驟3.收集特定地址的始終可用的通知級別跟蹤。您可以使用行動對等IP或MAC進行過濾。

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

此命令生成過去10分鐘的日誌，可以使用此命令調整此時間 **show logging profile wireless last 1 hour filter mac AAAA.BBBB.CCCC to-file bootflash:ra-AAAA.BBBB.CCCC.txt.**

您可以顯示作業階段上的內容，也可以將檔案複製到外部TFTP伺服器。

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

or

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

無線電活動跟蹤

如果永遠線上日誌沒有提供足夠的資訊來瞭解隧道配置期間觸發了什麼問題，則可以啟用條件調試

和捕獲 Radio Active (RA) 跟蹤，提供更為詳細的流程活動。

步驟1.驗證是否未啟用調試條件。

```
# show debugging
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop
```

```
IOSXE Packet Tracing Configs:
```

```
Packet Infra debugs:
```

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

如果您看到與要監控的地址不相關的任何條件，請將其禁用。

要刪除特定地址，請執行以下操作：

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

要刪除所有條件（建議方式）：

```
# clear platform condition all
```

步驟2.為要監控的地址新增調試條件。

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

註：如果要同時監視多個移動對等體，請使用 `debug platform condition feature wireless mac` 命令的 MAC地址。

步驟3.讓9800 WLC開始監控指定的地址活動。

```
# debug platform condition start
```

注意：移動活動的輸出不顯示，因為所有內容都在內部緩衝，稍後收集。

步驟4.重現您要監控的問題或行為。

步驟5.停止調試。

```
# debug platform condition stop
```

步驟6.收集地址活動的輸出。

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

此命令會生成過去10分鐘的日誌。可以使用**show logging profile wireless last 1 hour filter mac AAAA.BBB.CCCC to-file bootflash:ra-AAAA.BBB.CCCC.txt**命令調整此時間。

您可以複製 **FILENAME.txt** 或直接在螢幕上顯示輸出。

將檔案複製到外部伺服器：

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

顯示內容：

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

步驟7.如果仍無法找到故障的原因，請收集內部日誌級別。

(無需再次調試客戶端。使用已內部儲存的日誌，但收集範圍更廣的日誌)。

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

您可以複製 **FILENAME.txt** 或直接在螢幕上顯示輸出。

將檔案複製到外部伺服器：

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

顯示內容：

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

步驟8.移除偵錯條件。

```
# clear platform condition all
```

注意：在排除會話故障後始終刪除調試條件。

在9800 WLC上成功建立行動通道範例。

```
2021/09/28 10:20:50.497612 {mobilityd_R0-0}{1}: [errmsg] [26516]: (info): %MM_NODE_LOG-6-MEMBER_ADDED: Adding Mobility member (IP: IP: 172.16.55.28: default)
2021/09/28 10:20:52.595483 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_data of XID (0) to (ipv4: 172.16.55.28 )
2021/09/28 10:20:52.595610 {mobilityd_R0-0}{1}: [mm-pmtu] [26516]: (debug): Peer IP:
172.16.55.28 PMTU size is 1385 and calculated additional header length is 148
2021/09/28 10:20:52.595628 {mobilityd_R0-0}{1}: [mm-client] [26516]: (debug): MAC:
0000.0000.0000 Sending keepalive_ctrl_req of XID (80578) to (ipv4: 172.16.55.28 )
2021/09/28 10:20:52.595686 {mobilityd_R0-0}{1}: [mm-keepalive] [26516]: (note): Peer IP:
172.16.55.28 keepalive data packet missed, 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 missed, 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 missed, 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

```

嵌入式封包擷取

大多數情況下，它對於檢查WLC之間交換的資料包非常有用。過濾捕獲特別有用，Access Control Lists (ACLs) 以限制捕獲的流量。

這是CLI上嵌入捕獲的配置模板。

步驟1.建立過濾器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

```

步驟2.定義捕獲引數：

```

monitor capture <CAPTURE_NAME> access-list <ACL_NAME> buffer size 10 control-plane both

```

```
interface <INTERFACE_NAME> both limit duration 300
```

註：為INTERFACE_NAME引數選擇管理介面

步驟3.開始捕獲：

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

步驟4.停止捕獲：

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

步驟5.在GUI上導覽至Troubleshooting > Packet Capture，以收集封包擷取檔案。

常見故障排除場景

下一個範例包括在9800個WLC之間形成的通道。

由於連線問題導致的控制和資料路徑中斷

啟用 Always-On-Logs 和 Embedded packet captures 提供故障排除的其他資訊：

```
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 missed, 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 missed, 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 missed, total missed packet = 30
```

封包擷取對於確認行為非常有用。

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

請注意，debug和WLC都顯示沒有響應控制或資料ping。常見的情況顯示允許IP連線，但不16666許端16667連線埠透過網路通訊。

WLC之間的組態不相符

在此案例中，我們確認了WLC之間所有連線埠的連線，但繼續發現keepalive遺漏。

啟用 Always-On-Logs 和 Embedded packet captures 提供故障排除的其他資訊：


```
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 missed, 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 missed, total missed packet = 3
```

對等172.16.55.28上的內部日誌可幫助我們確認配置不匹配

```
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
```

常見配置不匹配包括：組名稱不正確，上的不匹配 **Data Link Encryption** 和錯誤的移動mac地址。

組不匹配日誌：

```
2021/09/28 17:33:22.963 {mobilityd_R0-0}{1}: [errmsg] [27081]: (ERR): %MM_INFRA_LOG-3-
MSG_PROC_FAILED_GROUP_NAME_HASH: Pkt group name hash: 82FE070E6E9A37A543CEBED96DB0388F Peer
group name hash: 3018E2A00F10176849AC824E0190AC86 Failed to validate endpoint. reason: Group
name hash mismatch.
```

MAC地址不匹配日誌：

```
2021/09/28 19:09:33.455 {mobilityd_R0-0}{1}: [errmsg] [27081]: (ERR): %MM_INFRA_LOG-3-
MSG_PROC_FAILED_MAC_ADDR: Pkt MAC: 001e.e67e.75fa Peer MAC: 001e.e67e.75ff Failed to validate
endpoint. reason: MAC address mismatch.
```

DTLS握手問題

此類問題與WLC之間的DTLS隧道建立相關。這可能是在資料路徑為UP但控制路徑保留的情況下DOWN。

啟用 **Always-On-Logs** 和 **Embedded packet captures** 提供故障排除的其他資訊：

```
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
```

使用 **show wireless management trustpoint** 和 **show crypto pki trustpoints commands** 驗證憑證資訊。

HA SSO案例

如果您在高可用性SSO對中有控制器，有一個重要的提示需要知道。移動MAC地址在預設情況下未配置，如果發生故障轉移，可能導致移動隧道關閉。

show wireless mobility summary為您提供了當前使用的移動MAC，但不需要進行配置。檢查組態是

否使用show run設定了行動化MAC | i移動性

如果在運行配置中未配置移動mac，則在故障切換到備用WLC時它會更改，這會導致移動隧道失敗。

簡單的解決方案是導覽至**Configuration > Wireless > Mobility** Web UI頁面並按**Apply**。這會將目前的行動化MAC儲存到組態中。然後，在保留故障切換和移動隧道時，MAC保持不變。

如果您通過命令列執行移動配置，但忘記配置移動MAC地址，則主要會發生此問題。當您應用設定時，Web UI會自動儲存移動MAC地址。

相關資訊

- [在Catalyst 9800上設定WLAN錨點行動功能](#)
- [技術支援與文件 - Cisco Systems](#)

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。