

在9800无线局域网控制器中配置FlexConnect mDNS

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[网络图](#)

[配置](#)

[在WLC中全局启用mDNS](#)

[选择服务列表中的mDNS服务](#)

[将服务列表与服务策略合并](#)

[配置mDNS Flex配置文件](#)

[使用mDNS网关模式配置WLAN](#)

[将mDNS服务策略应用于WLAN策略](#)

[在FlexConnect策略中配置mDNS Flex Profile](#)

[验证](#)

[WLC Show命令](#)

[AP Show命令](#)

[故障排除](#)

[有线服务提供商](#)

[无线服务提供商](#)

[主mDNS AP](#)

[每个mDNS服务列表不允许的服务](#)

[服务定义中的自定义mDNS服务配置](#)

[FlexConnect mDNS桥接模式](#)

[Flexconnect mDNS丢弃模式](#)

简介

本文档介绍如何在9800无线LAN控制器中配置FlexConnect组播域名系统(mDNS)网关。

先决条件

要求

Cisco 建议您了解以下主题：

- 9800无线LAN控制器mDNS概念
- FlexConnect本地交换概念

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Catalyst 9800无线控制器系列(Catalyst 9800-L)、Cisco IOS® XE Cupertino 17.9.5
- 集成多业务路由器(ISR)、Cisco IOS® XE Gibraltar 17.6.5
- Catalyst 3560系列交换机，Cisco IOS® 15.2.4E10
- 接入点9117AXI-B，接入点9130AXI-B

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

组播域名系统(mDNS)是一种协议，可灵活地发现和共享服务提供商(SP)和服务用户（无线客户端）之间的服务。服务提供商是指提供打印机、智能电视、文件共享服务等服务的设备，服务用户可以利用这些设备。

mDNS协议基于UDP，对IPv4使用端口5353、Mac地址01:00:5E：00:00：FB和IP地址224.0.0.251，对IPv6使用端口FF02：：FB。

mDNS在WLC中有两种工作模式：桥接和网关。桥接模式仅在服务提供商和服务用户必须位于同一子网中的同一个VLAN（第2层）中工作。网关模式与相同或不同VLAN中的服务提供商和服务用户协同工作，WLC或AP执行Bonjour网关以缓存来自服务提供商的服务并与服务用户共享。

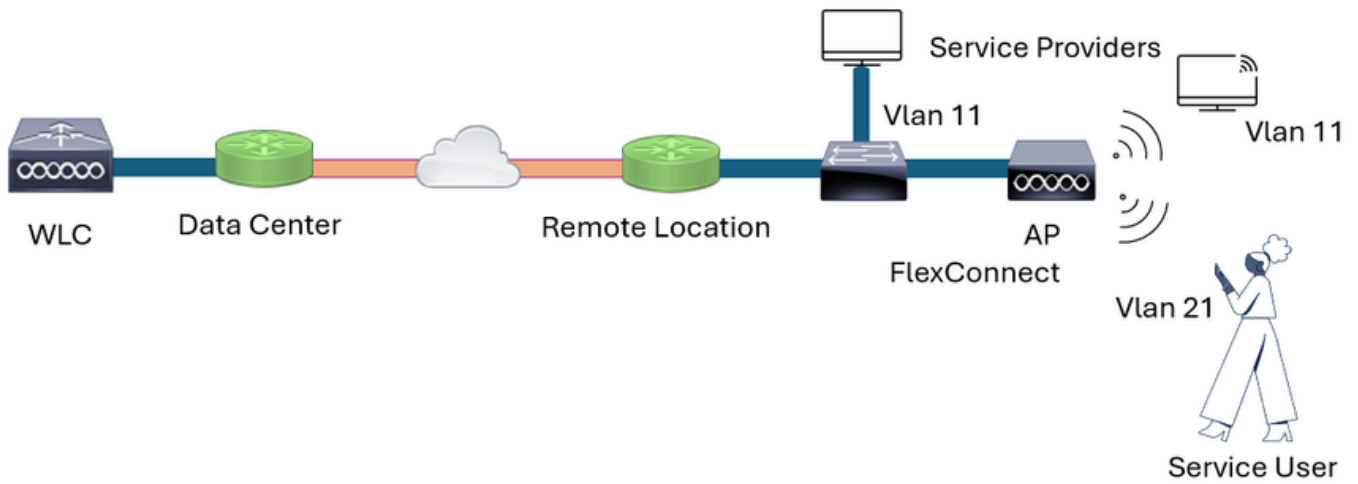
本文档仅基于mDNS FlexConnect本地交换，在这种情况下，AP充当mDNS网关以缓存服务提供商通告的服务，并与服务用户共享这些服务。



注意：有关集中交换mDNS配置，请参阅[了解Catalyst 9800无线控制器上的mDNS](#)

网络图

无线和有线服务提供商在FlexConnect本地交换环境中通告mDNS服务，以及使用mDNS服务的无线客户端（服务用户）。



配置

在WLC中全局启用mDNS

要使AP作为mDNS网关工作，需要全局启用mDNS网关来启用该功能。

WLC GUI

Configuration > Services > mDNS

Global Service Policy mDNS Flex Profile

mDNS Gateway **ENABLED**

Transport

Active-Query Timer (Minutes) *

mDNS-AP Service Policy [Clear](#)

mDNS全局配置

WLC CLI

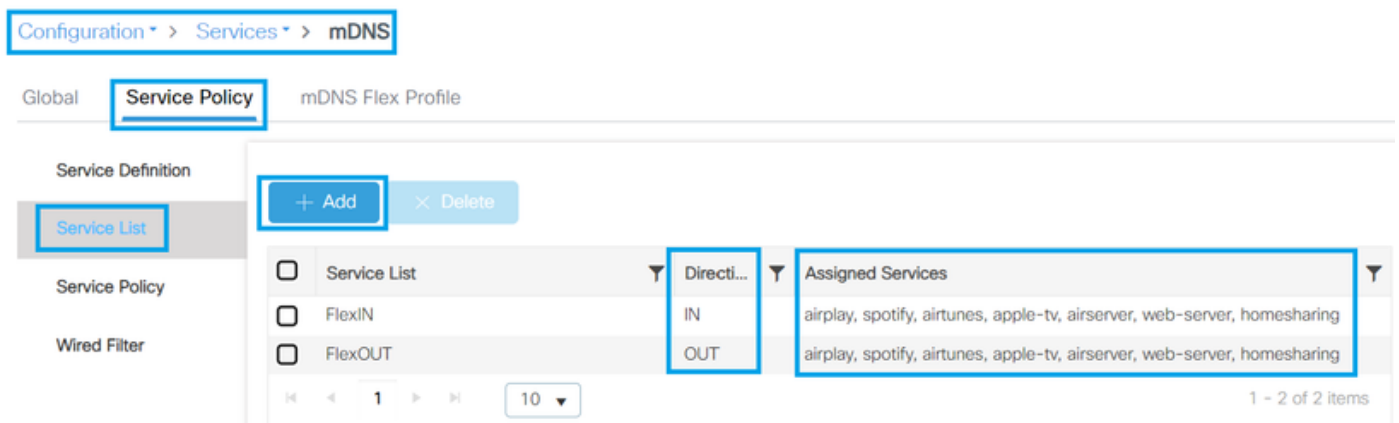
WLC#

```
WLC#conf t
WLC(config)#mdns-sd gateway
WLC(config-mdns-sd)#end
WLC#
```

选择服务列表中的mDNS服务

配置服务列表以允许首选的mDNS服务。该列表必须在两个方向上配置，即IN和OUT，以过滤充当mDNS网关的接入点允许哪些入口和出口服务。

WLC GUI



在Service List中选择所需的服务

WLC CLI

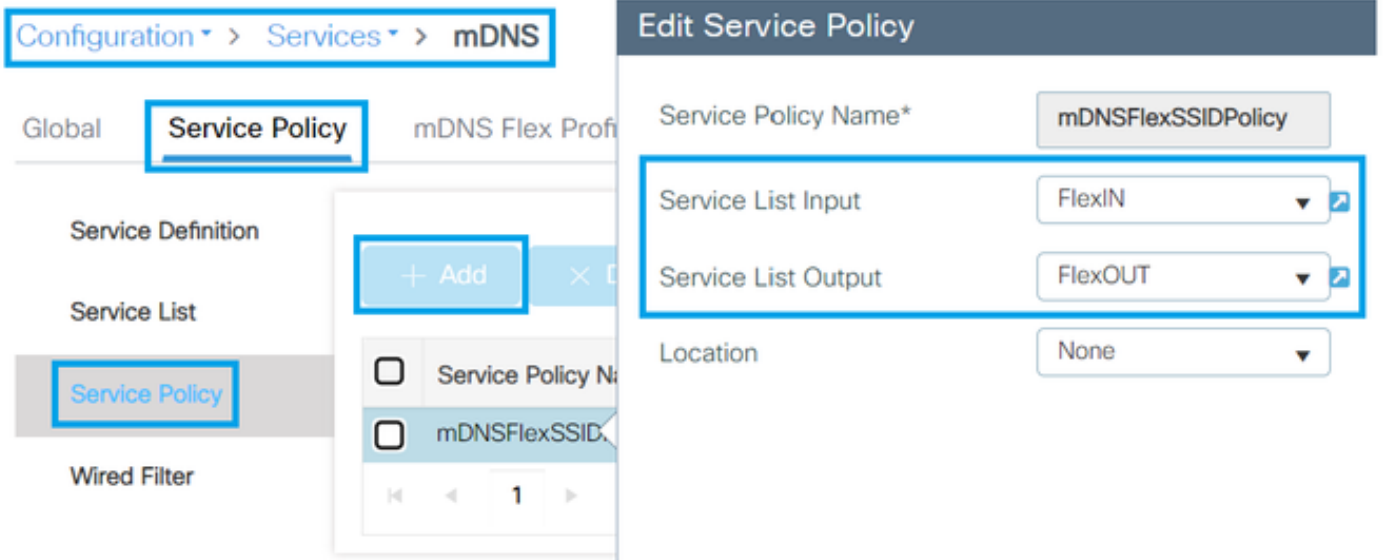
```
WLC#
WLC#conf t
WLC(config)#mdns-sd service-list FlexIN IN
WLC(config-mdns-sl-in)#match airplay
WLC(config-mdns-sl-in)#match spotify
WLC(config-mdns-sl-in)#exit

WLC(config)#mdns-sd service-list FlexOUT OUT
WLC(config-mdns-sl-out)#match airplay
WLC(config-mdns-sl-out)#match spotify
WLC(config-mdns-sl-out)#end
WLC#
```

将服务列表与服务策略合并

使用所需的服务配置了Service List IN和Service List OUT后，将使用服务策略进行合并。合并后，此服务策略可用于WLAN策略、FlexConnect配置文件和mDNS Flex策略。

WLC GUI



合并mDNS策略中的mDNS服务列表

WLC CLI

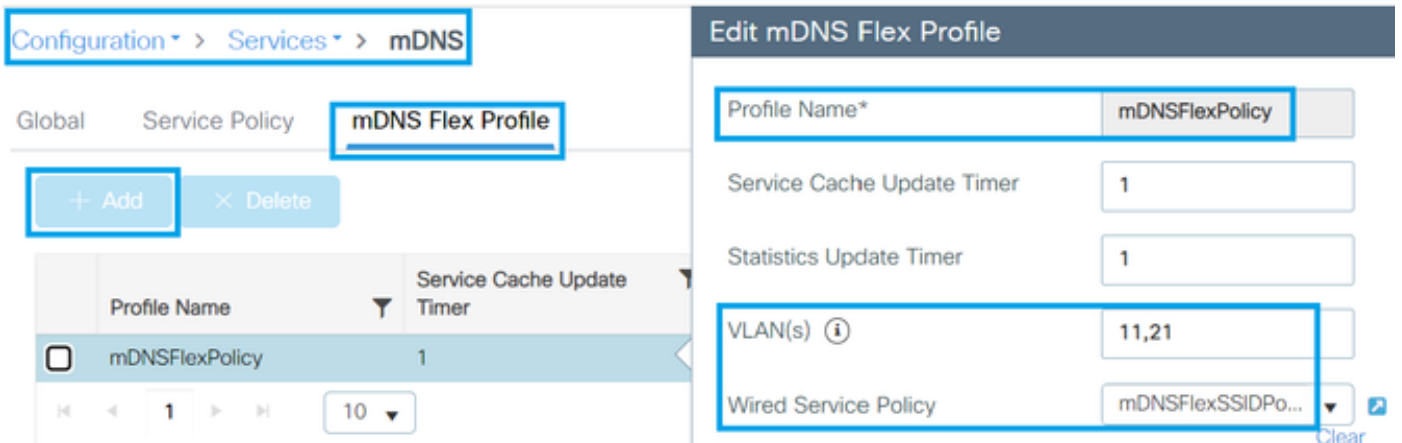
```

WLC#
WLC#conf t
WLC(config)#mdns-sd service-policy mDNSFlexSSIDPolicy
WLC(config-mdns-ser-pol)#service-list FlexIN IN
WLC(config-mdns-ser-pol)#service-list FlexOUT OUT
WLC(config-mdns-ser-pol)#end
WLC#
  
```

配置mDNS Flex配置文件

在mDNS Flex Profile中，需要将使用mDNS的FlexConnect本地交换Vlan添加到Flex Profile中，服务提供商和服务用户的Vlan必须添加到mDNS Flex Profile中，同时添加mDNS服务策略，以允许通过有线方式过滤服务。

WLC GUI



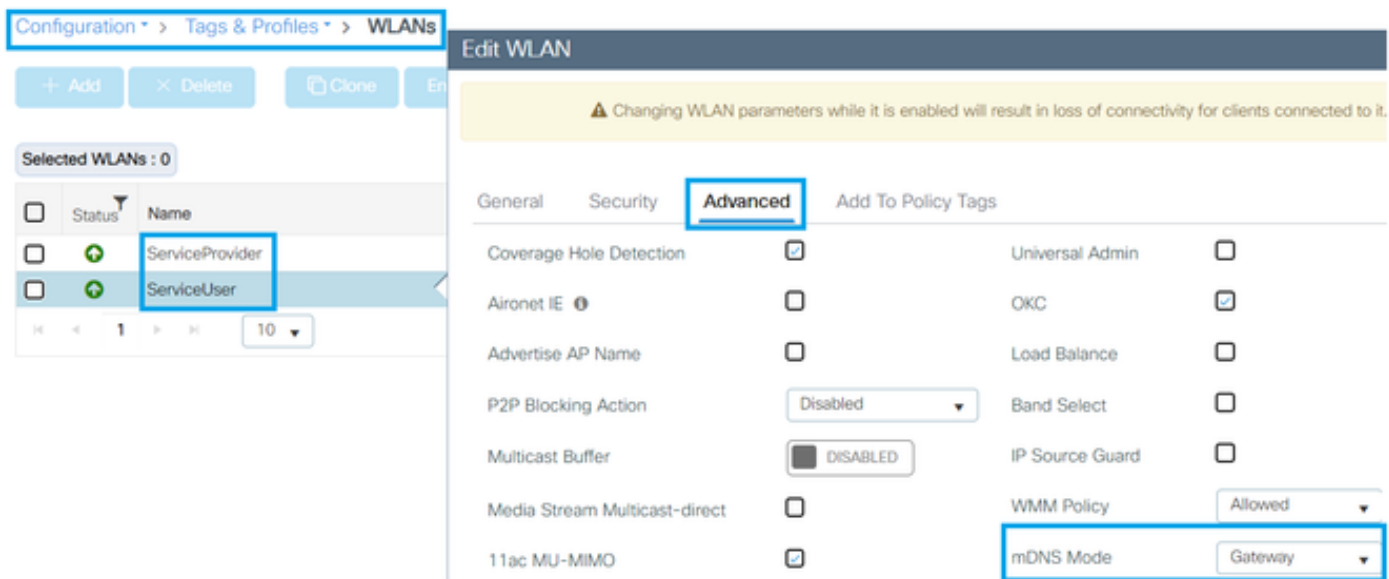
WLC CLI

```
WLC#  
WLC#conf t  
WLC(config)#mdns-sd flex-profile mDNSFlexPolicy  
WLC(config-mdns-flex-prof)#wired-vlan-range 11,21  
WLC(config-mdns-flex-prof)#wired-service-policy mDNSFlexSSIDPolicy  
WLC(config-mdns-flex-prof)#end  
WLC#
```

使用mDNS网关模式配置WLAN

默认情况下，每个WLAN都将mDNS模式用作桥接。为了让AP知道何时充当通过无线连接的服务提供商和服务用户的mDNS网关，必须使用mDNS作为网关模式配置WLAN。

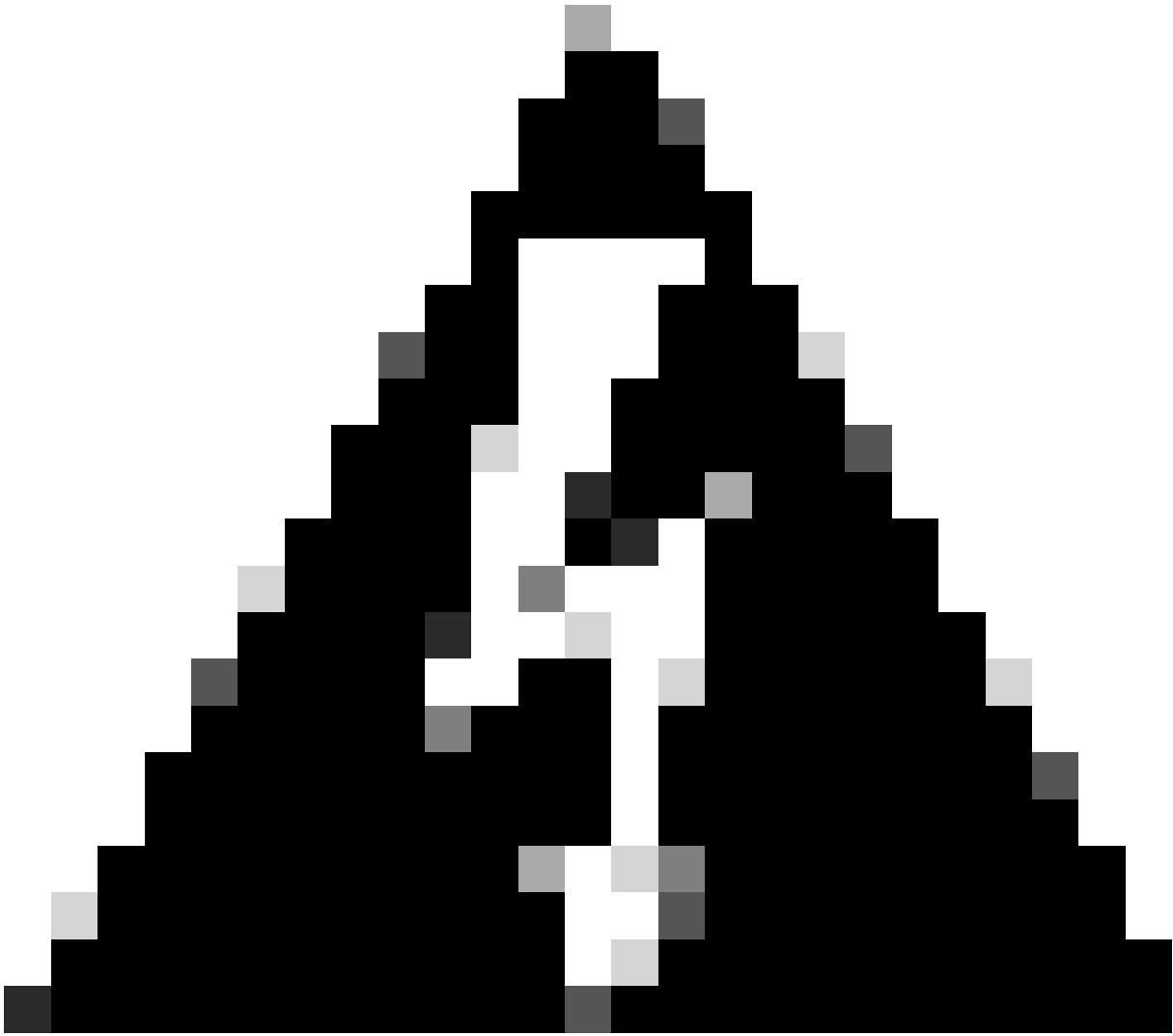
WLC GUI



在mDNS网关模式下配置SSID

WLC CLI

```
WLC#  
WLC#conf t  
WLC(config)#wlan ServiceUser  
WLC(config-wlan)#mdns-sd-interface gateway  
WLC(config-wlan)#end  
WLC#
```

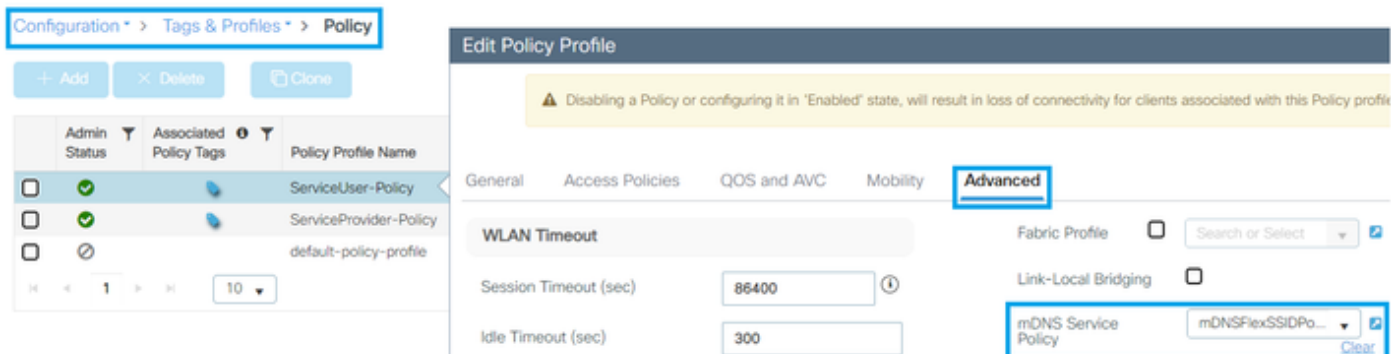


警告：WLAN中的配置更改会导致连接的无线客户端从SSID中删除。在生产期间，请谨慎更改WLAN的任何配置。

将mDNS服务策略应用于WLAN策略

对于无线服务提供商和无线用户提供商，在对WLAN的WLAN策略应用后，将使用先前配置的mDNS策略过滤mDNS服务。

WLC GUI



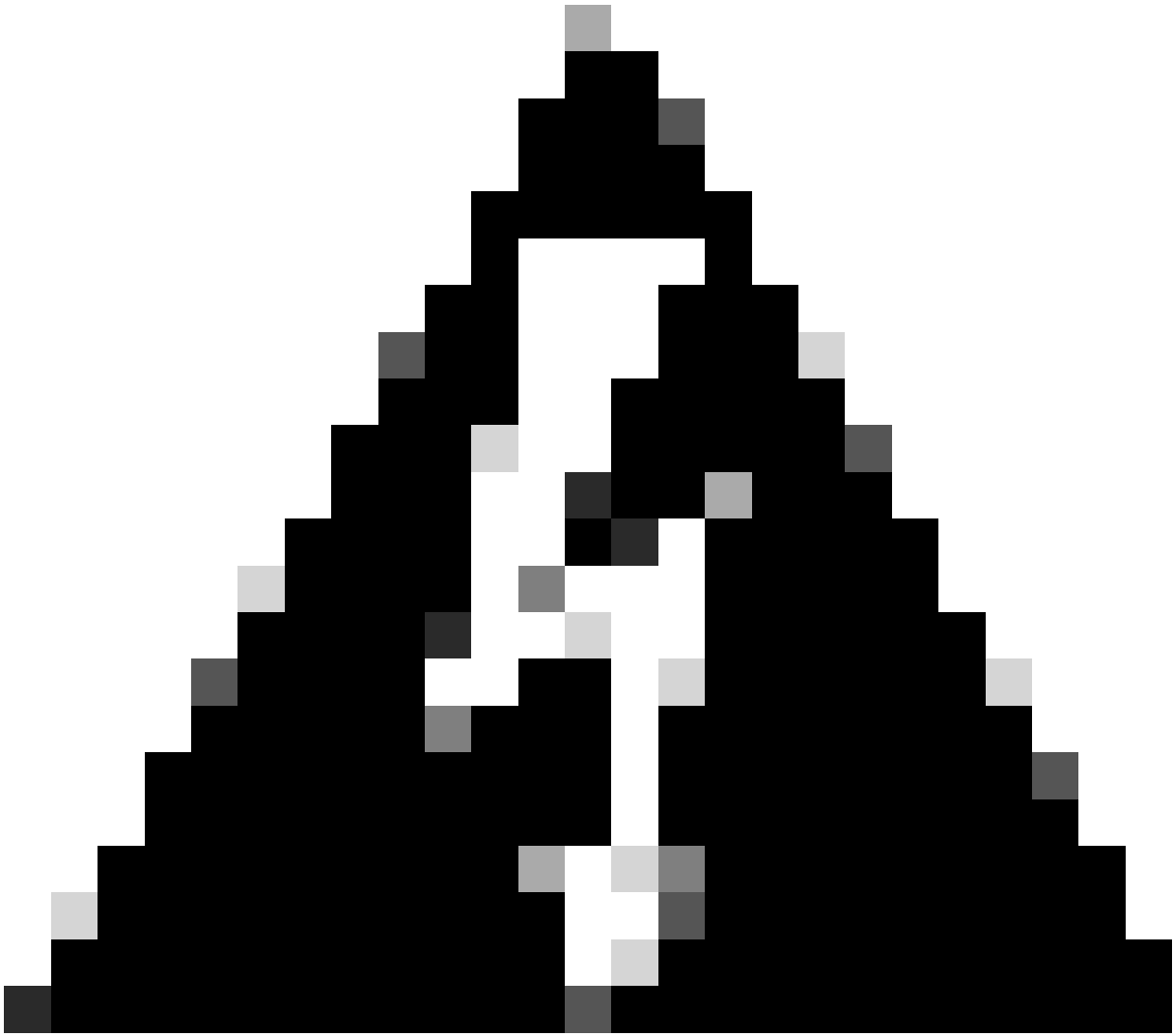
分配mDNS策略

WLC CLI

```

WLC#
WLC#conf t
WLC(config)#wireless profile policy ServiceUser-Policy
WLC(config-wireless-policy)#mdns-sd service-policy mDNSFlexSSIDPolicy
WLC(config-wireless-policy)#end
WLC#

```



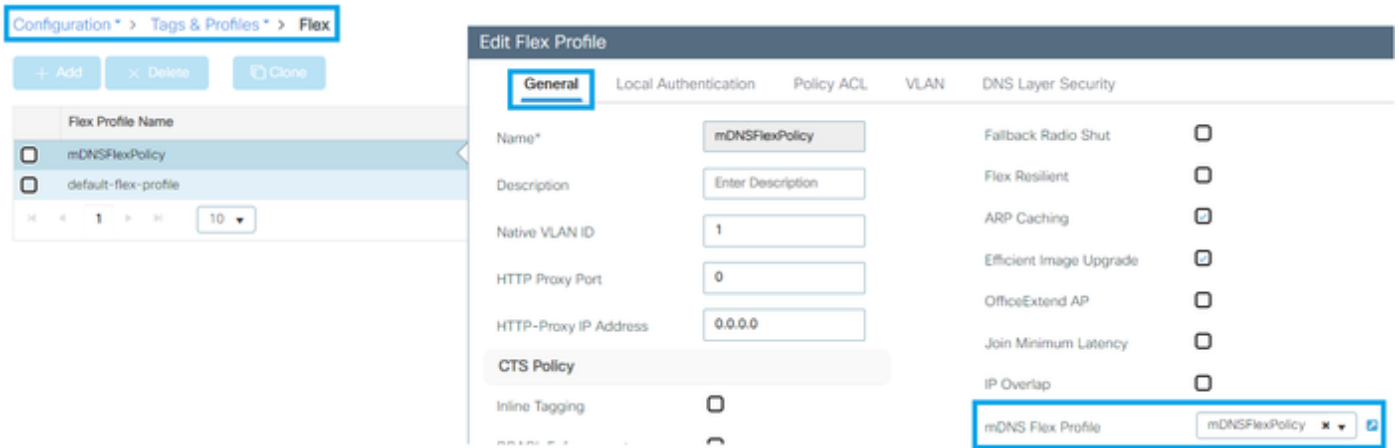
警告：WLAN策略中的配置更改导致连接的无线客户端从WLAN中丢弃。在生产期间，请小心处理WLAN策略中的任何配置。

注意：有关常规FlexConnect配置，请参阅[了解Catalyst 9800无线控制器上的FlexConnect](#)

在FlexConnect策略中配置mDNS Flex Profile

在应用Vlan、ACL等配置的FlexConnect策略中，需要选择mDNS Flex Profile才能将其应用于属于FlexConnect策略的AP。

WLC GUI



分配mDNS Flex配置文件

WLC CLI

```

WLC#
WLC#conf t
WLC(config)#wireless profile flex mDNSFlexPolicy
WLC(config-wireless-flex-profile)#mdns-sd profile mDNSFlexPolicy
WLC(config-wireless-flex-profile)#end
WLC#

```

验证

从WLC和AP，可使用这些命令检查配置。

WLC Show命令

可以使用以下命令检查常规FlexConnect mDNS配置的示例：

```
<#root>
```

```
WLC#
```

```
show run | sec mdns-sd
```

```

mdns-sd gateway
mdns-sd service-list FlexIN IN
  match airplay
  match spotify
  match airtunes
  match apple-tv
  match airserver
  match web-server
  match homesharing
mdns-sd service-list FlexOUT OUT
  match airplay
  match spotify

```

```
match airtunes
match apple-tv
match airserver
match web-server
match homesharing
mdns-sd service-policy mDNSFlexSSIDPolicy
service-list FlexIN IN
service-list FlexOUT OUT
mdns-sd flex-profile mDNSFlexPolicy
wired-vlan-range 11,21
wired-service-policy mDNSFlexSSIDPolicy
mdns-sd profile mDNSFlexPolicy
```

可以使用以下命令检查WLAN mDNS模式：

```
<#root>
```

```
WLC#
```

```
show wlan name ServiceUser | in mDNS
```

```
mDNS Gateway Status : Gateway
WLC#
```

```
show wlan name ServiceProvider | in mDNS
```

```
mDNS Gateway Status : Gateway
```

可以使用以下命令检查WLAN策略mDNS配置：

```
<#root>
```

```
WLC#
```

```
show wireless profile policy detailed ServiceUser-Policy | in mDNS
```

```
mDNS Service Policy name : mDNSFlexSSIDPolicy
WLC#
```

```
show wireless profile policy detailed ServiceProvider-Policy | in mDNS
```

```
mDNS Service Policy name : mDNSFlexSSIDPolicy
```

AP Show命令

可以使用以下命令从AP端检查与mDNS相关的配置：

```
<#root>
```

9130mDNSAP#

show mdns profile detail

```
FlexIN_IN _home-sharing._tcp.local ANY
FlexIN_IN _airplay._tcp.local ANY
FlexIN_IN _airserver._tcp.local ANY
FlexIN_IN _raop._tcp.local ANY
FlexIN_IN _spotify-connect._tcp.local ANY
FlexIN_IN _http._tcp.local ANY
FlexOUT_OUT _home-sharing._tcp.local ANY
FlexOUT_OUT _airplay._tcp.local ANY
FlexOUT_OUT _airserver._tcp.local ANY
FlexOUT_OUT _raop._tcp.local ANY
FlexOUT_OUT _spotify-connect._tcp.local ANY
FlexOUT_OUT _http._tcp.local ANY
```

<#root>

9130mDNSAP#

show mdns status

```
Global mDNS gateway:Enabled
vap_id ssid mdns_mode
0 ServiceUser Gateway
1 ServiceProvider Gateway
Active query interval:30
vap service_list_in service_list_out location
0 FlexIN_IN FlexOUT_OUT 0
1 FlexIN_IN FlexOUT_OUT 0
Wired vlan configuration: 11 21
mdns stats timer: 1
mdns cache timer: 1
AP Sync VLAN: 10
Wired service list IN: FlexIN_IN
Wired service list OUT: FlexOUT_OUT
```

<#root>

9130mDNSAP#

show mdns ap-table

```
AP_ETH_MAC Last_message_time Msg_seq Is_primary_ap
3C:57:31:55:E4:28 1721178339 133 YES
0C:D0:F8:98:1B:F0 1721178339 133 NO
```

故障排除

为了进行故障排除，本文档将说明mDNS在FlexConnect本地交换中经历的工作流程。请务必记住，由于部署模式是FlexConnect本地交换，WLC在如何管理mDNS方面不会起到任何作用。

AP本身将成为mDNS网关设备，AP从服务提供商获取服务并与服务用户共享服务，而AP、服务提供商和服务用户位于不同的VLAN中。

根据网络图部分：

- 服务提供商位于Vlan 11中
- 服务用户在Vlan 21中
- 接入点位于Vlan 10中

有线服务提供商

服务提供商一旦检测到网络存在连接，便会使用称为探测的机制发送mDNS查询，以确保是否有任何其他网络设备提供相同的mDNS服务。在探测之后，有线服务提供商使用通告机制，它会发送mDNS类型响应来通告其支持的服务。

接下来，从mDNS网关AP交换机端口捕获数据包，显示服务提供商宣布其支持的服务。数据包源自VLAN 11中服务提供商的MAC地址和IP地址，它具有mDNS的MAC地址和IP地址（包括UDP上的mDNS端口5353）的目的地址，它还包含服务提供商所支持的服务的答案。

下图中的答案部分显示我们感兴趣的服务，即airplay和spotify，随后AP会缓存这些服务并将其保存在数据库中。

No.	Time	Source	Destination	Protocol	Length	Info
322	00:06:04.535225	192.168.11.54	224.0.0.251	MDNS	848	Standard query response 0x0000 TXT, cach

```
> Frame 322: 848 bytes on wire (6784 bits), 848 bytes captured (6784 bits) on interface \Device\NPF {CC2095A1-094C-440E-A8E1-...}
> Ethernet II, Src: SamsungElect_45:8e:26 (e0:03:6b:45:8e:26), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
> Internet Protocol Version 4, Src: 192.168.11.54, Dst: 224.0.0.251
> User Datagram Protocol, Src Port: 5353, Dst Port: 5353
v Multicast Domain Name System (response)
  v Transaction ID: 0x0000
    > [Expert Info (Warning/Protocol): DNS response retransmission. Original response in frame 320]
    > Flags: 0x8400 Standard query response, No error
    Questions: 0
    Answer RRs: 10
    Authority RRs: 0
    Additional RRs: 3
  v Answers
    > ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._tcp.local: type TXT, class IN, cache flush
    > services.dns-sd.udp.local: type PTR, class IN, spotify-connect._tcp.local
    > _spotify-connect._tcp.local: type PTR, class IN, ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._tcp.local
    > ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._tcp.local: type SRV, class IN, cache flush, priority 0, we
    > Samsung CU7000 55 TV._airplay._tcp.local: type TXT, class IN, cache flush
    > services.dns-sd.udp.local: type PTR, class IN, airplay._tcp.local
    > _airplay._tcp.local: type PTR, class IN, Samsung CU7000 55 TV._airplay._tcp.local
    > Samsung CU7000 55 TV._airplay._tcp.local: type SRV, class IN, cache flush, priority 0, weight 0, port 7000, target lo
    > localhost.local: type A, class IN, cache flush, addr 192.168.11.54
    > localhost.local: type AAAA, class IN, cache flush, addr fe80::e203:6bff:fe45:8e26
```

mDNS服务提供商服务

从AP CLI中，还可以看到有线服务提供商通告，若要从AP本身查看任何mDNS信息，必须启用以下调试：

- AP#debug mdns events
- AP#debug mdns packets

<#root>

```
Jul 17 23:51:32 kernel: [*07/17/2024 23:51:32.0403] chatter: MDNSGW-EVENT:
```

```
flex mdns gw: Recieved wired mdns packet on vlan 11
```

```
Jul 17 23:51:32 kernel: [*07/17/2024 23:51:32.0403] chatter: MDNSGW-EVENT: push: adding ptr record to c
```

```
Jul 17 23:51:32 kernel: [*07/17/2024 23:51:32.0404] chatter: MDNSGW-EVENT: mdns_ptr_db:updated TXT reco
```

```
Jul 17 23:51:32 kernel: [*07/17/2024 23:51:32.0404] chatter: MDNSGW-EVENT: mdns_ptr_db:added/updated PTR
```

```
Jul 17 23:51:32 kernel: [*07/17/2024 23:51:32.0404] chatter: MDNSGW-EVENT:
```

```
push: added ptr record to cache: srv_name: _spotify-connect._tcp.local
```

```
Jul 17 23:51:32 kernel: [*07/17/2024 23:51:32.0404] chatter: MDNSGW-EVENT: push: adding ptr record to c
```

```
Jul 17 23:51:32 kernel: [*07/17/2024 23:51:32.0404] chatter: MDNSGW-EVENT: mdns_ptr_db:updated TXT reco
```

```
Jul 17 23:51:32 kernel: [*07/17/2024 23:51:32.0405] chatter: MDNSGW-EVENT: mdns_ptr_db:added/updated PTR
```

```
Jul 17 23:51:32 kernel: [*07/17/2024 23:51:32.0405] chatter: MDNSGW-EVENT:
```

```
push: added ptr record to cache: srv_name: _airplay._tcp.local
```

一旦AP获知服务，它会将相同的服务保存在数据库中。

可以使用以下命令检查AP数据库中保存的服务：

- AP#show mdns cache

就本文档而言，下一个输出显示相关信息以证明mDNS网关AP在其缓存中有服务，但是，输出更长。

接下来，突出显示服务、服务提供商的MAC地址以及从中获取该地址的VLAN。

<#root>

```
AP#show mdns cache
```

```
----- Service Provider Records-----
```

```
service_name service_provider
```

```
_airplay._tcp.local
```

```
Samsung CU7000 55 TV._airplay._tcp.1
```

```
_spotify-connect._tcp.local
```

```
ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._t
```

```
Total Services: 2
```

```
Total Service Providers: 2
```

```
----- PTR Records -----
```

```
service_name
```

```
client_mac
```



```

ap_mac ap_ether_mac wired is_rlan is_aaa_override
vlan
 wlan_id ttl flags client_type record_type target site_name ap_location ssid type
Samsung CU7000 55 TV._airplay._tcp.local
E0:03:6B:45:8E:26
 00:00:00:00:00:00 00:00:00:00:00:00 true false false
11
 16 3840 132 0 12 _airplay._tcp.local PTR
ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._tcp.local
E0:03:6B:45:8E:26
 00:00:00:00:00:00 00:00:00:00:00:00 true false false
11
 16 3840 132 0 12 _spotify-connect._tcp.local PTR

```

有线服务提供商宣布服务后，且AP已缓存服务并保存在其数据库中（如前面的步骤所示），服务用户（无线客户端）会查找将设备（笔记本电脑）的内容镜像到智能电视进行镜像显示。为了完成镜像显示，服务用户在本例中利用airplay服务。

由于服务用户通过无线连接，因此需要通过无线捕获数据包来查看来自服务用户的连接mDNS流。

从Over the Air捕获中可以看到，作为Vlan 21中无线客户端的服务用户如何发送具有mDNS的802.11目标MAC地址的mDNS查询，并且从IP地址部分使用mDNS的IP地址以及目标，端口是UDP 5353，并且在mDNS查询中请求播放。作为源，服务用户的MAC地址与其IP地址一起使用。

No.	Time	Source	Destination	Protocol	Length	Info
1954	01:59:02.016646	192.168.21.54	224.0.0.251	MDNS	264	Standard query

```

> 802.11 radio information
v IEEE 802.11 QoS Data, Flags: o.....TC
  Type/Subtype: QoS Data (0x0028)
  > Frame Control Field: 0x8881
    .000 0000 0011 0000 = Duration: 48 microseconds
    Receiver address: Cisco_b3:20:af (0c:75:bd:b3:20:af)
    Transmitter address: a6:c5:15:dc:dd:57 (a6:c5:15:dc:dd:57)
    Destination address: IPv4mcast_fb (01:00:5e:00:00:fb)
    Source address: a6:c5:15:dc:dd:57 (a6:c5:15:dc:dd:57)
    BSS Id: Cisco_b3:20:af (0c:75:bd:b3:20:af)
    STA address: a6:c5:15:dc:dd:57 (a6:c5:15:dc:dd:57)
    .... 0000 = Fragment number: 0
    0000 0000 1101 .... = Sequence number: 13
    Frame check sequence: 0x00000000 [unverified]
    [FCS Status: Unverified]
    [WLAN Flags: o.....TC]
  > Qos Control: 0x1316
  > HT Control (+HTC): 0x0000b20f
> Logical-Link Control
> Internet Protocol Version 4, Src: 192.168.21.54, Dst: 224.0.0.251
> User Datagram Protocol, Src Port: 5353, Dst Port: 5353
v Multicast Domain Name System (query)
  Transaction ID: 0x0000
  > Flags: 0x0000 Standard query
  Questions: 4
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  v Queries
    > _companion-link._tcp.local: type PTR, class IN, "QU" question
    > _rdlink._tcp.local: type PTR, class IN, "QU" question
    > _airplay._tcp.local: type PTR, class IN, "QU" question
    > _sleep-proxy._udp.local: type PTR, class IN, "QU" question

```

mDNS服务用户服务请求

从AP调试中可以看到AP如何接收无线mDNS数据包。调试显示请求的服务，这些服务与上一步捕获的数据包显示的服务相同。所使用的mDNS调试包括：

- AP#debug mdns events
- AP#debug mdns packets

<#root>

```
Jul 18 02:04:45 kernel: [*07/18/2024 02:04:45.1824] chatter: MDNSGW-EVENT:
```

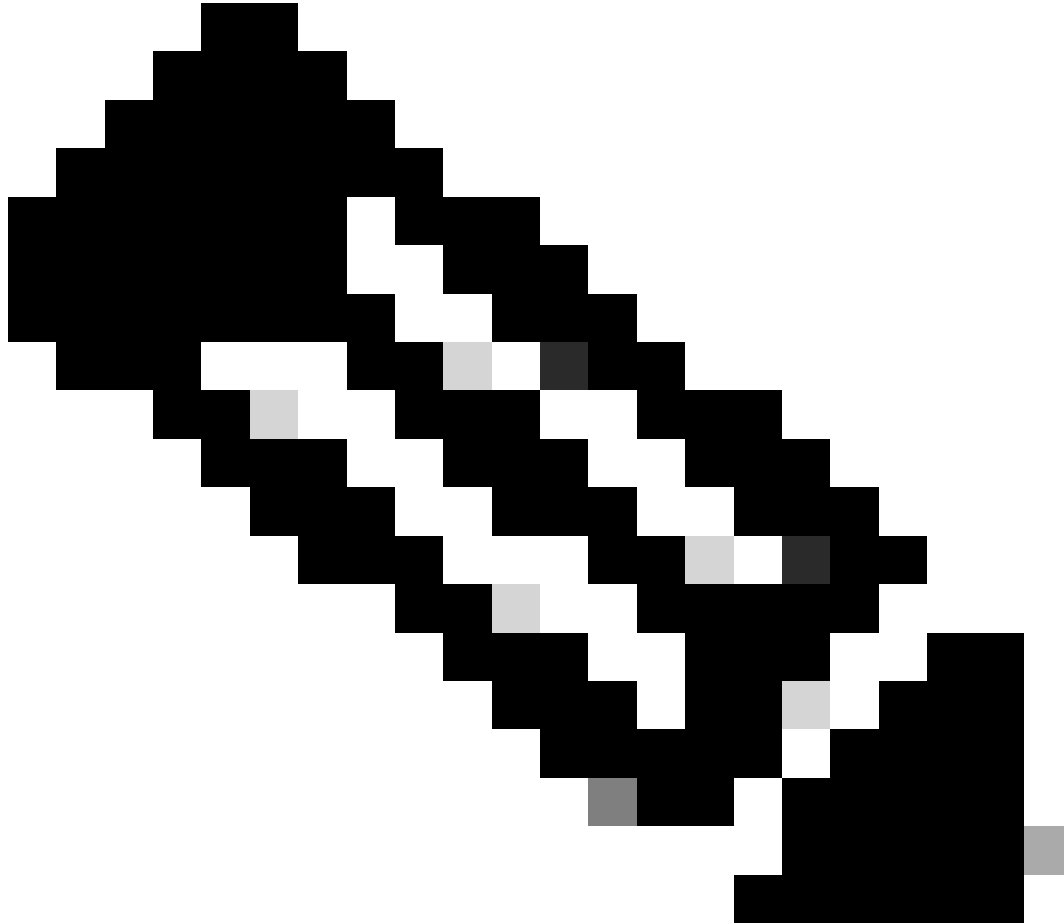
```
flex mdns gw: Recieved wireless mdns packet
```

```

Jul 18 02:04:45 kernel: [*07/18/2024 02:04:45.1824] chatter: MDNSGW-PAK: query: 0/3 '_companion-link._tcp.local'
Jul 18 02:04:45 kernel: [*07/18/2024 02:04:45.1824] chatter: MDNSGW-PAK: query: 1/3 '_rdlink._tcp.local'
Jul 18 02:04:45 kernel: [*07/18/2024 02:04:45.1824] chatter: MDNSGW-PAK: query: 2/3 '_sleep-proxy._udp.local'
Jul 18 02:04:45 kernel: [*07/18/2024 02:04:45.7442] chatter: MDNSGW-PAK: query: 0/1 '

```

_airplay._tcp.local



注意：要用处于嗅探器模式的AP来接管Air数据包捕获，请参阅本文档[在Catalyst 9800无线控制器上配置处于嗅探器模式的接入点](#)。要使用MacBook接管Air数据包捕获，请参阅本文档中的[收集MacBook上通过空气的数据包捕获](#)

AP收到来自服务用户的mDNS查询后，会构建mDNS响应并通过无线发送。响应源自接入点MAC Add和IP地址，目标是服务用户（无线客户端）MAC地址，但是，mDNS IP地址与作为应答的所需服务一起使用，这意味着此数据包将发送到服务用户，并且是mDNS数据包。

从数据包中可以看出，由于AP充当mDNS网关，因此AP如何在IP部分中使用自己的IP地址和mDNS端口UDP 5353将数据包发送至mDNS IP地址。

No.	Time	Source	Destination	Protocol	Length	Info
6077	01:59:15.050707	192.168.10.63	224.0.0.251	MDNS	720	Standard query response

```

> AiroPeek/OmniPeek encapsulated IEEE 802.11
> 802.11 radio information
v IEEE 802.11 QoS Data, Flags: .....F.C
  Type/Subtype: QoS Data (0x0028)
  > Frame Control Field: 0x8802
    .000 0000 0010 1100 = Duration: 44 microseconds
    Receiver address: a6:c5:15:dc:dd:57 (a6:c5:15:dc:dd:57)
    Transmitter address: Cisco_b3:20:af (0c:75:bd:b3:20:af)
    Destination address: a6:c5:15:dc:dd:57 (a6:c5:15:dc:dd:57)
    Source address: Cisco_b5:e9:d0 (0c:75:bd:b5:e9:d0)
    BSS Id: Cisco_b3:20:af (0c:75:bd:b3:20:af)
    STA address: a6:c5:15:dc:dd:57 (a6:c5:15:dc:dd:57)
    .... .... 0000 = Fragment number: 0
    0000 0000 0110 .... = Sequence number: 6
    Frame check sequence: 0x00000000 [unverified]
    [FCS Status: Unverified]
    [WLAN Flags: .....F.C]
  > Qos Control: 0x0006
> Logical-Link Control
> Internet Protocol Version 4, Src: 192.168.10.63, Dst: 224.0.0.251
> User Datagram Protocol, Src Port: 5353, Dst Port: 5353
v Multicast Domain Name System (response)
  Transaction ID: 0x0000
  > Flags: 0x8000 Standard query response, No error
  Questions: 0
  Answer RRs: 3
  Authority RRs: 0
  Additional RRs: 0
  v Answers
    > _meta_response._mdns._gateway._ap._local: type TXT, class IN
    > _airplay._tcp.local: type PTR, class IN, Samsung CU7000 55 TV._airplay._tcp.local
    > Samsung CU7000 55 TV._airplay._tcp.local: type TXT, class IN

```

来自AP的mDNS服务响应

从调试中可以看出，mDNS响应被发送到服务用户，而知道mDNS响应的方法就是检查服务用户的MAC地址和响应中接入点的MAC地址。它们一起显示，如下面显示的调试突出显示部分所示，从数据包捕获的上一步可以看出，服务用户的MAC地址是a6c515dcdd57，接入点的MAC地址是0c75bdb5e9d0。

<#root>

Jul 18 02:04:45 kernel: [*07/18/2024 02:04:45.7450] chatter: mdns response packet 599 |

a6c515dc dd570c75 bdb5e9d0

08004500 02490000 0000fa11 1ddec0a8 0a3fc0a8 153614e9 14e90235 6b330000 80000000 00030000 00000e5f 6d6

前面的步骤成功完成了FlexConnect本地交换的mDNS数据包流，其中服务提供商在Vlan 11中通过有线连接，AP在Vlan 10中连接，服务用户在Vlan 21中连接。

无线服务提供商

无线服务提供商的工作方式与有线服务提供商机制完全相同，它发送探测和服务通知，AP缓存服务并将它们保存到数据库中。本部分旨在说明当服务提供商通过无线连接时，执行mDNS网关的AP如何获取服务。

有线和无线服务提供商之间的区别在于，数据包自802.11发生以来在空中呈现的方式。在下一个数据包中，可以看到Vlan 11中的无线服务提供商如何通过端口UDP 5353发送带有源（其自身的MAC地址和IP地址）的mDNS数据包，且目的地是mDNS Mac地址和IP ADD，服务列为answers。

No.	Time	Source	Destination	Protocol	Length	Info
6567	02:48:39.311587	192.168.11.55	224.0.0.251	MDNS	969	Standard query response 0x0000 TXT, cache


```
IEEE 802.11 QoS Data, Flags: .....TC
  Type/Subtype: QoS Data (0x0028)
  > Frame Control Field: 0x8801
    .000 0000 0011 0000 = Duration: 48 microseconds
  Receiver address: Cisco_b3:20:ae (0c:75:bd:b3:20:ae)
  Transmitter address: SamsungElect_6e:eb:0c (68:fc:ca:6e:eb:0c)
  Destination address: IPv4mcast_fb (01:00:5e:00:00:fb)
  Source address: SamsungElect_6e:eb:0c (68:fc:ca:6e:eb:0c)
  BSS Id: Cisco_b3:20:ae (0c:75:bd:b3:20:ae)
  STA address: SamsungElect_6e:eb:0c (68:fc:ca:6e:eb:0c)
  .... .... 0000 = Fragment number: 0
  0010 1001 0001 .... = Sequence number: 657
  Frame check sequence: 0x00000000 [unverified]
  [FCS Status: Unverified]
  [WLAN Flags: .....TC]
  > Qos Control: 0x0000
  > Logical-Link Control
  > Internet Protocol Version 4, Src: 192.168.11.55, Dst: 224.0.0.251
  > User Datagram Protocol, Src Port: 5353, Dst Port: 5353
  > Multicast Domain Name System (response)
    > Transaction ID: 0x0000
    > Flags: 0x8400 Standard query response, No error
    Questions: 0
    Answer RRs: 10
    Authority RRs: 0
    Additional RRs: 3
    > Answers
      > ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._tcp.local: type TXT, class IN, cache flush
      > _services._dns-sd._udp.local: type PTR, class IN, _spotify-connect._tcp.local
      > _spotify-connect._tcp.local: type PTR, class IN, ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._tcp.local
      > ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._tcp.local: type SRV, class IN, cache flush, priority 0, wei
      > Samsung CU7000 55 TV._airplay._tcp.local: type TXT, class IN, cache flush
```

无线服务提供商mDNS服务

通过AP调试，可以看到AP如何获取无线mDNS数据包并将获取的服务添加到数据库。

<#root>

```
Jul 18 02:42:01 kernel: [*07/18/2024 02:42:01.7785] chatter: MDNSGW-EVENT:
```

```
flex mdns gw: Recieved wireless mdns packet
```

```
Jul 18 02:42:01 kernel: [*07/18/2024 02:42:01.7786] chatter: MDNSGW-EVENT:
```

```
push: added ptr record to cache: srv_name: _spotify-connect._tcp.local
```

```

Jul 18 02:42:01 kernel: [*07/18/2024 02:42:01.7786] chatter: MDNSGW-EVENT: push: adding ptr record to c
Jul 18 02:42:01 kernel: [*07/18/2024 02:42:01.7786] chatter: MDNSGW-EVENT: push: adding ptr record to c
Jul 18 02:42:01 kernel: [*07/18/2024 02:42:01.7787] chatter: MDNSGW-EVENT: mdns_ptr_db:updated TXT reco
Jul 18 02:42:01 kernel: [*07/18/2024 02:42:01.7787] chatter: MDNSGW-EVENT: mdns_ptr_db:added/updated PTR
Jul 18 02:42:01 kernel: [*07/18/2024 02:42:01.7787] chatter: MDNSGW-EVENT:

```

```

push: added ptr record to cache: srv_name: _airplay._tcp.local

```

一旦AP缓存了服务，就会构建数据库，并且与有线服务提供商服务相比显示一些差异，因为AP中的无线服务提供商数据库显示SSID名称、站点名称（站点标签）等详细信息，下面还会显示更多突出显示内容。

```

<#root>

```

```

AP#show mdns cache

```

```

----- Service Provider Records-----
service_name service_provider
  _airplay._tcp.local Samsung CU7000 55 TV._airplay._tcp.local
  _spotify-connect._tcp.local ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._tcp.local

```

```

Total Services: 2
Total Service Providers: 2

```

```

----- PTR Records -----
service_name client_mac ap_mac ap_ether_mac wired is_rlan is_aaa_override
vlan
wlan_id
ttl flags client_type record_type target
site_name
ap_location
ssid
type
Samsung CU7000 55 TV._airplay._tcp.local 68:FC:CA:6E:EB:0C 0C:75:BD:B3:20:A0 0C:75:BD:B5:E9:D0 false f
11
1
4320 132 0 12 _airplay._tcp.local m
DNSFlex-Site-TAG
RemoteLocation
ServiceProvider
PTR

```

ed9583d2b239afa30d7b0e7106c3710ddcfe5769._spotify-connect._tcp.local 68:FC:CA:6E:EB:0C 0C:75:BD:B3:20:A

11

1

4320 132 0 12 _spotify-connect._tcp.local

mDNSFlex-Site-TAG

RemoteLocation

ServiceProvider

PTR

mDNS用户服务查询和AP mDNS网关应答与有线服务提供商(Wired Service Provider)部分中已说明的完全相同，服务用户发送mDNS查询，AP mDNS充当网关，并向服务用户发送包含所需服务详细信息的响应。

主mDNS AP

每个站点标签只有一个主mDNS AP，它负责两项工作：

1. 只要所有AP mDNS数据库属于相同的站点标记，请保持更新这些数据库，这样每个AP的mDNS数据库都相同，并且不会丢失mDNS服务。
2. 通知WLC有关在远程位置获取的mDNS服务（这只是提供信息，仅用于管理，WLC不能使用这些服务）。

主AP通知从非主AP的角度更新，请记住所有AP都位于此站点的Vlan 10中：

<#root>

Jul 18 03:26:25 kernel: [*07/18/2024 03:26:25.4852] chatter:

MDNSGW-EVENT: flex mdns gw: Recieved wired mdns packet on vlan 10

Jul 18 03:26:25 kernel: [*07/18/2024 03:26:25.4853] chatter: MDNSGW-EVENT:

Received _heartbeat

record. data: digest=f7adbb063c274f6e4219f3a36abf7f787075b7e1

Jul 18 03:26:25 kernel: [*07/18/2024 03:26:25.4853] chatter: seq=355

Jul 18 03:26:25 kernel: [*07/18/2024 03:26:25.4853] chatter:

is_primary_ap=true

Jul 18 03:26:25 kernel: [*07/18/2024 03:26:25.4854] chatter: MDNSGW-EVENT: Calculated digest=f7adbb063c

Jul 18 03:26:25 kernel: [*07/18/2024 03:26:25.4854] chatter: MDNSGW-EVENT: Verified meta message

Jul 18 03:26:25 kernel: [*07/18/2024 03:26:25.4854] chatter: MDNSGW-EVENT: [0C:75:BD:B5:E9:D0]

Verified message from 3C:57:31:55:E4:28

Jul 18 03:26:25 kernel: [*07/18/2024 03:26:25.4854] chatter: MDNSGW-EVENT: New pkt from 3C:57:31:55:E4:

```
Jul 18 03:26:25 kernel: [*07/18/2024 03:26:25.4854] chatter: MDNSGW-EVENT: mdns_gw_ap_mgr :: MdnsGwApMgr
3C:57:31:55:E4:28
] Received _meta_heartbeat with message: seq=355,
is_primary=true

9130mDNSAP#
show mdns ap-table

AP_ETH_MAC

Last_message_time Msg_seq
Is_primary_ap

3C:57:31:55:E4:28

1721273666 363
YES

9130mDNSAP#
```

主mDNS AP 向其他AP通知主AP在站点TAG和主用AP所属网络中获知的服务，一旦mDNS信息数据包到达同一站点标记中的其他AP，在获知新服务时，AP中的mDNS缓存数据库将会更新：

<#root>

```
Jul 18 03:41:26 kernel: [*07/18/2024 03:41:26.1021] chatter:
MDNSGW-EVENT: forward_packet: sending packet on vlan 10

Jul 18 03:41:26 kernel: [*07/18/2024 03:41:26.1023] chatter:
send meta packet

177 | 01005e00 00fb3c57 3155e428 08004500 00a30000 0000fa11 1469c0a8 0a3de000 00fb14e9 14e9008f 450e00
```

对WLC的主要mDNS AP数据库更新：

<#root>

```
Jul 18 03:35:26 kernel: [*07/18/2024 03:35:26.3127] chatter: MDNSGW-EVENT:
mdns_gw_visibility

:: MdnsGwVisibility: MDNS Stats Timer triggered
```



```

Jul 18 03:35:26 kernel: [*07/18/2024 03:35:26.3128] chatter: MDNSGW-PAK: mdns_gw_visibility :: MdnsGwVi
Jul 18 03:35:26 kernel: [*07/18/2024 03:35:26.3130] chatter: MDNSGW-EVENT: mdns_gw_visibility :: MdnsGwVi
Jul 18 03:35:26 kernel: [*07/18/2024 03:35:26.3131] chatter: MDNSGW-EVENT: mdns_gw_visibility ::

```

MdnsGwVisibility: sending mdns cache IAPP payload. Total payloads sent - 2

主AP向WLC通知的服务提供的信息包含了解到的服务，如果AP（在本示例中是有线服务提供商）通过有线或无线方式了解这些服务，则提供获取这些服务的站点TAG和Vlan以及服务提供商名称。对于无线服务提供商，WLAN ID反映其连接的WLAN。

Monitoring > Services > mDNS

ClearAll

Record Name	TTL(sec)	Client MAC	Client Type	Connection Type	AP MAC	Site Tag	WLAN/GLAN/RLAN ID	VLAN ID	Record Data
_airplay._tcp.local	4500	e003.6b45.8e26	WIRED	Wired	0000.0000.0000	mDNSFile... Site-TAG (Flex)	11	11	Samsung CU7000 55'
_spotify-connect._tcp.local	4500	e003.6b45.8e26	WIRED	Wired	0000.0000.0000	mDNSFile... Site-TAG (Flex)	11	11	ed9583d2b239afa30d connect._tcp.local

1 - 2 of 2 clients

从WLC GUI监控mDNS服务

每个mDNS服务列表不允许的服务

mDNS服务列表和策略允许控制网络中允许的mDNS服务，这里举例说明如何过滤服务列表IN和OUT中不允许的mDNS服务。

要查看正在通告或查询但不允许的服务，请在AP中启用此调试：

- AP#debug mdns errors

这些mDNS服务

- _airplay-bds._tcp.local
- _wake._tcp.local

不允许，因为未在“选择mDNS服务”中配置的“服务列表”中配置和选择它们。

<#root>

```
Jul 18 03:46:41 kernel: [*07/18/2024 03:46:41.6986] chatter:
```

```
MDNSGW-ERROR: Handle query:
```

```
service_string:_airplay-bds._tcp.local
```

```
not allowed by policy
```

```
. Skipping it.
```

```
Jul 18 03:46:53 kernel: [*07/18/2024 03:46:53.7270] chatter:
```

```
MDNSGW-ERROR: Handle query:
```

```
service_string:6A:FC:CA:6E:EB:0C@0.0.0.0._wake._tcp.local
```

not allowed by policy

. Skipping it.

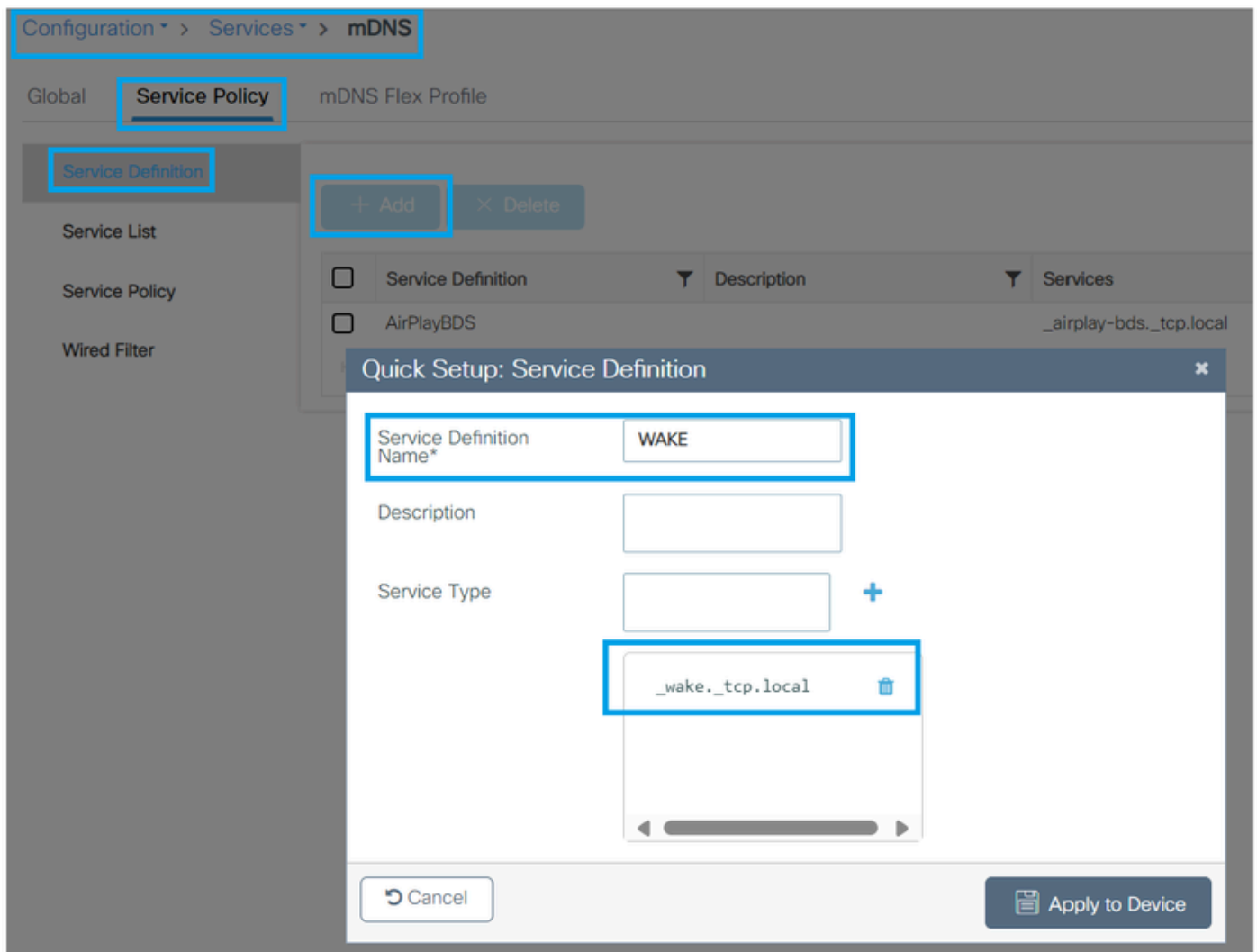
服务定义中的自定义mDNS服务配置

如果需要特殊服务列表，需要将相同服务添加到WLC中的mDNS配置的服务定义部分。

在WLC中添加服务并在服务列表IN和OUT中选择服务后，这些服务将通过mDNS服务策略推送到FlexConnect AP。

为此，我们需要了解所需的确切服务，并从“服务定义”部分为服务和服务字符串添加自定义名称。

在本示例中，我在每个mDNS服务列表不允许的服务部分中添加了由mDNS网关AP过滤的两个服务。



mDNS自定义服务定义

FlexConnect mDNS桥接模式

本文档不涵盖mDNS桥接模式，因为事实上，在FlexConnect本地交换中，此mDNS模式从AP的角度被视为常规数据流量。在FlexConnect本地交换中为mDNS启用桥接模式时，AP仅转发从有线或无线接收的mDNS数据包。这些数据包仅在相同的VLAN中转发，这意味着服务提供商和服务用户必须位于同一个VLAN中才能运行mDNS。mDNS桥接在Vlan中无法运行。

Flexconnect mDNS丢弃模式

如果某些WLAN不需要mDNS，但其他WLAN确实需要mDNS，则可以针对每个WLAN配置mDNS模式丢弃。启用mDNS丢弃后，mDNS不会通过连接到WLAN的设备。

The screenshot shows the 'Edit WLAN' configuration page. The 'Advanced' tab is selected, and the 'mDNS Mode' is set to 'Drop'. The 'Selected WLANs' list on the left shows 'ServiceUser' selected. The 'mDNS Mode' dropdown is highlighted with a blue box.

General	Security	Advanced	Add To Policy Tags
Coverage Hole Detection	<input checked="" type="checkbox"/>	Universal Admin	<input type="checkbox"/>
Aironet IE	<input type="checkbox"/>	OKC	<input checked="" type="checkbox"/>
Advertise AP Name	<input type="checkbox"/>	Load Balance	<input type="checkbox"/>
P2P Blocking Action	Disabled	Band Select	<input type="checkbox"/>
Multicast Buffer	DISABLED	IP Source Guard	<input type="checkbox"/>
Media Stream Multicast-direct	<input type="checkbox"/>	WMM Policy	Allowed
11ac MU-MIMO	<input checked="" type="checkbox"/>	mDNS Mode	Drop

mDNS丢弃模式

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。