



# mDNS Technical Guide for Cisco Catalyst 9800 Series Wireless Controllers

## Cisco IOS XE Amsterdam 17.3.1

Mike Adler, TME

## Table of Contents

|                                                                                          |    |
|------------------------------------------------------------------------------------------|----|
| Objectives .....                                                                         | 2  |
| Audience .....                                                                           | 3  |
| Cisco IOS Software Documentation .....                                                   | 3  |
| Platform Support .....                                                                   | 3  |
| Supported Software releases .....                                                        | 3  |
| mDNS Overview .....                                                                      | 3  |
| C9800 mDNS Services in release IOS-XE 16.11-17.3 .....                                   | 5  |
| mDNS Gateway in Flex Connect Mode .....                                                  | 7  |
| Modes of mDNS Gateway FC AP Operation .....                                              | 8  |
| Discovering mDNS Services on the Controller and Flex APs .....                           | 9  |
| mDNS on the Global Level .....                                                           | 9  |
| mDNS Gateway on the WLAN Level .....                                                     | 10 |
| Caching mDNS Services on the Controller and Flex AP .....                                | 11 |
| mDNS Gateway Service Policy .....                                                        | 12 |
| mDNS gateway with Guest Anchor support .....                                             | 14 |
| Foreign Controller in Bridging Mode and Anchor Controller in GW enabled mode .....       | 14 |
| Foreign Controller in GW Mode Enabled and Anchor Controller in GW or Bridging mode ..... | 15 |
| Mobility Support for the mDNS Gateway .....                                              | 18 |
| Intra Controller Roaming .....                                                           | 18 |
| Inter Controller L3 Roaming .....                                                        | 18 |
| Common Local and Flex Mode mDNS Configuration Steps on C9800 .....                       | 19 |
| Configuring Policy Profile with mDNS settings .....                                      | 27 |
| Mapping Policy Tag with mDNS Policy Profile on the WLAN .....                            | 28 |
| Local or Native mDNS Policy Profile .....                                                | 29 |
| Mapping Site and Policy Tags to AP .....                                                 | 31 |
| Configuring Service Policy on the “VLAN Interface” .....                                 | 32 |
| mDNS-AP support in IOS-XE 17.1.1 and above .....                                         | 34 |
| Enable/Disable mDNS-AP .....                                                             | 35 |
| Flex Connect AP mDNS-Gateway Configuration Steps .....                                   | 36 |
| Global mDNS Monitoring on the C9800 .....                                                | 39 |

## Objectives

This Technical Guide provides information on the theory of operation and configuration of the mDNS solution on the Cisco

Catalyst 9800 IOS-XE controller rel 17.3.1 in support of multicast applications such as mDNS protocol such as Apple Bonjour. The Bonjour protocol enables Apple devices to query and announce for specific services such as AirPlay, which allows audio and video to be shared between devices dynamically.

## Audience

This Technical Guide is intended primarily for users who configure and maintain C9800 Wireless Controllers, but are not necessarily familiar with tasks, the relationship between tasks, or the commands necessary to perform particular tasks. In addition, this document is intended for users with some familiarity with Wireless Networks.

## Cisco IOS Software Documentation

In addition to the information provided in this guide, you might need to refer to the Cisco IOS-XE documentation set. The Cisco IOS software documentation is divided into several modules. Each module consists of a configuration guide and a corresponding command reference. Chapters in a configuration guide describe protocols, configuration tasks, and Cisco IOS software functionality and contain comprehensive configuration examples. Chapters in a command reference provide complete command syntax information. The configuration guide can be used in conjunction with its corresponding command reference. This deployment guide also should be used in conjunction with the mDNS modules in the Configuration and Command reference publications.

[https://www.cisco.com/c/en/us/td/docs/wireless/controller/9800/17-2/config-guide/b\\_wl\\_17\\_2\\_cg.html](https://www.cisco.com/c/en/us/td/docs/wireless/controller/9800/17-2/config-guide/b_wl_17_2_cg.html)

<https://www.cisco.com/c/dam/en/us/td/docs/wireless/controller/9800/17-1/deployment-guide/c9800-mdns-deployment-guide-rel-17-1.pdf>

## Platform Support

Catalyst wireless platforms 9800-L-F, 9800-40, 9800-80, 9800-CL

Catalyst COS Access Points: C9115, C9117, C9120 and C9130.

Wave-2 COS Access Points: AP18xx, AP2802, AP3802, AP4800, 1540 and 1560

Wave-1 IOS Access Points do not support mDNS Flex Gateway: AP1700, AP2700, AP3700 and AP1570

## Supported Software releases

IOS-XE -17.3.1 CCO posted software

<https://software.cisco.com/download/home/286322524>

## mDNS Overview

Bonjour is an Apple service discovery protocol, which locates devices such as printers, other mDNS advertising computers and the services that those devices offer on the network using multicast Domain Name System (mDNS) service records. The Bonjour protocol operates on service announcements and service queries, which allow devices to ask and advertise specific applications such as:

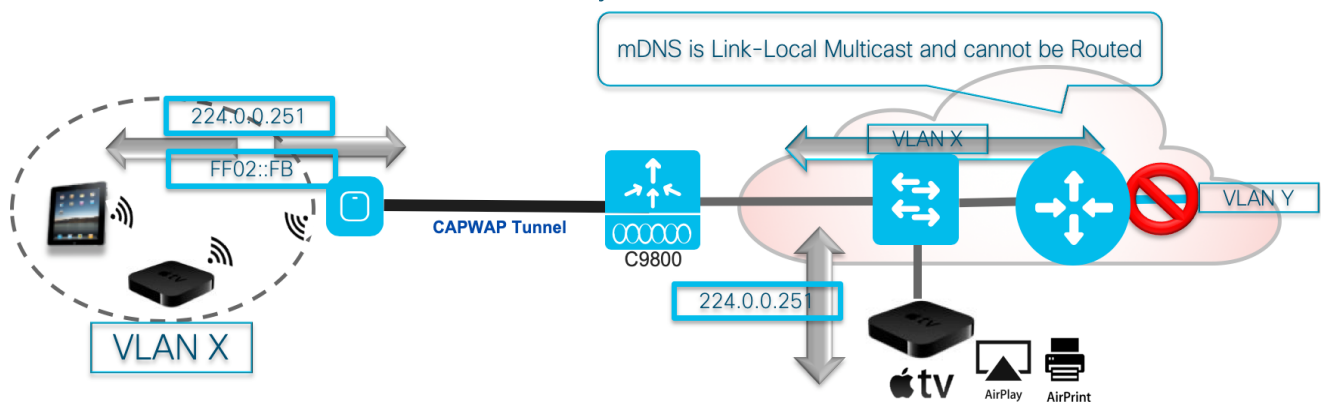
- Printing Services
- File Sharing Services
- Remote Desktop Services
- iTunes File Sharing
- iTunes Wireless iDevice Syncing
- AirPlay offering the following streaming services:
  - Music broadcasting
  - Video broadcasting
  - Full screen mirroring

Each query or advertisement is sent to the Bonjour multicast address for delivery to all clients on the subnet. Apple's Bonjour protocol relies on mDNS operating at UDP port 5353 and sent to the following reserved group addresses:

- IPv4 Group Address – 224.0.0.251
- IPv6 Group Address – FF02::FB

The addresses used by the Bonjour protocol are link-local multicast addresses, and thus are only forwarded to the local L2 domain. Routers cannot use multicast routing to redirect the traffic because the time to live (TTL) is set to one, and link-local multicast is meant to stay local by design.

#### mDNS services with Wired and Wireless Bonjour Devices



## C9800 mDNS Services in release IOS-XE 16.11-17.3

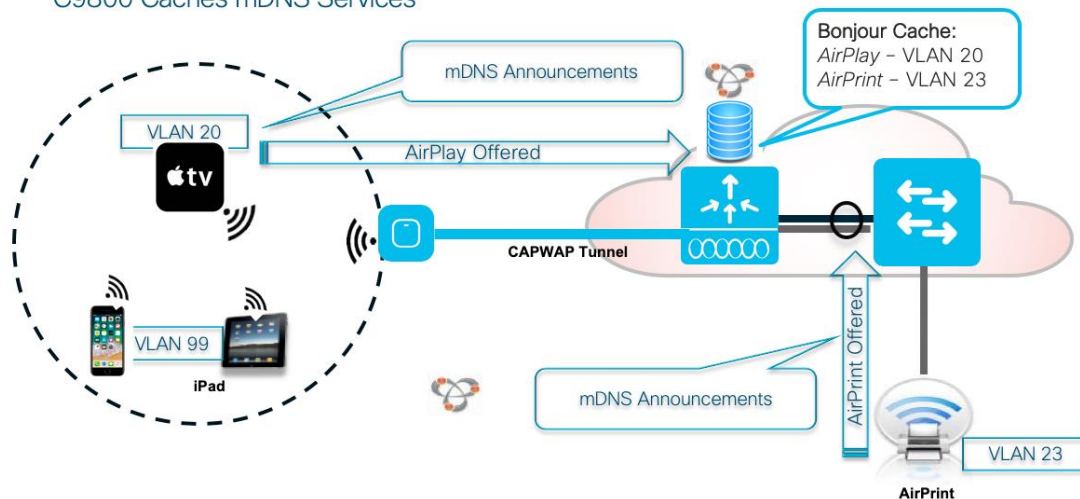
Catalyst 9800 is the next generation Wireless controller developed by Cisco. As we know penetration of Wireless devices is increasing day by day and it is heavily present in places like schools, universities and enterprises where people may not want to be aware of the of the networking details and configuration but still be able to discover devices and go on with their work as usual. mDNS is one of the protocols which is used for discovering the Bonjour services.

mDNS gateway snoops and caches Bonjour services across VLANs and periodically refreshes the same. Controller acts as a proxy for all mDNS services published by wireless and wired devices. Introduced in release 16.11, In 17.1 the mDNS gateway service gets more features and additional capabilities to, for example, filter cached wired / wireless service instances based on the credentials of the querying client and its location.

The following mDNS features included in IOS-XE rel 16.11:

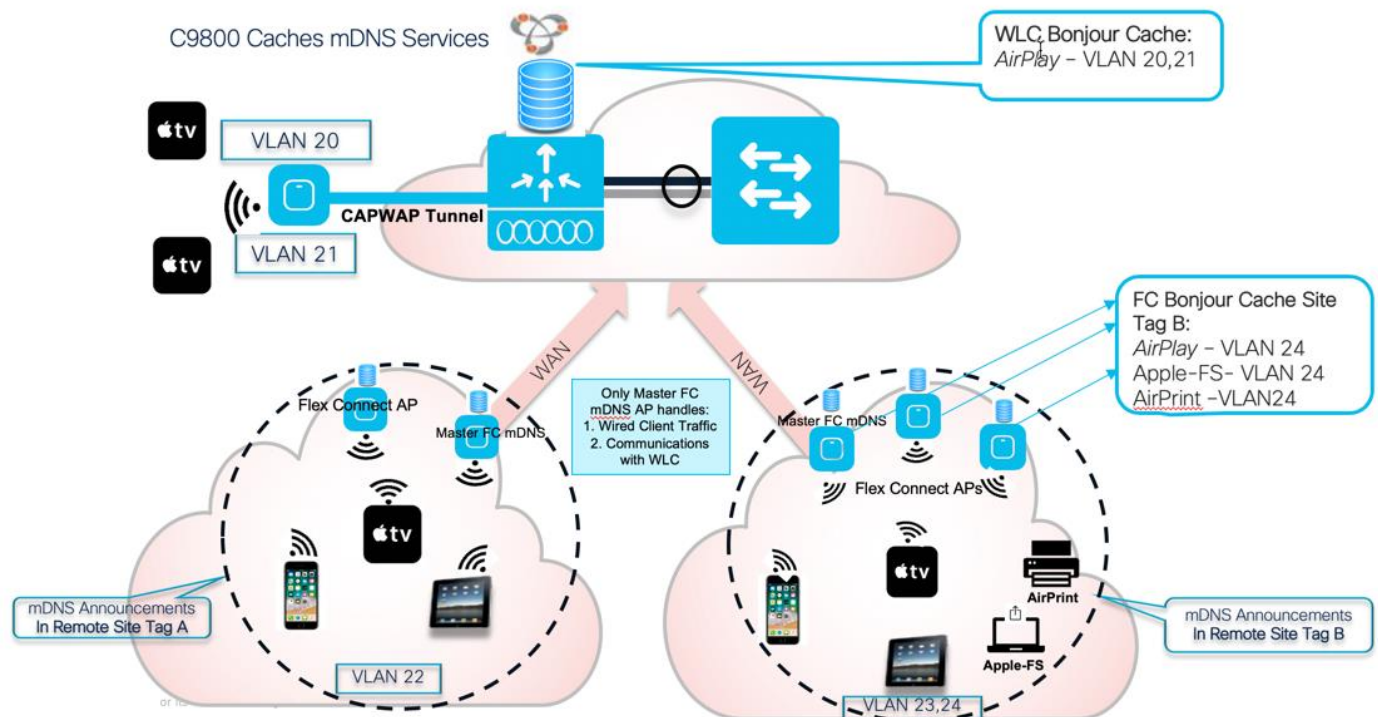
- Controller mDNS gateway
- Controller mDNS snooping
- mDNS support per WLAN
- mDNS Wireless Services Support
- mDNS Wired Services Support
- mDNS Service Policy profiles
- mDNS Policy Profiles
- Location Specific Services (LSS) RRM based for wireless service
- Location Services Site Tag based
- mDNS gateway with Guest Anchor support
- mDNS HA SSO (later release)
- mDNS RLAN mode is supported
- mDNS gateway supports both IPv4 and IPv6 records and transports
- mDNS L2 and L3 Intra and Inter Controller Mobility is supported
- mDNS will be supported by Yang/Netconf Model
- Bonjour debugging

## C9800 Caches mDNS Services



## Features added in IOS-XE release 17.1.1

- mDNS Service Policy support on Native/Local profile
- mDNS Service Policy on VLAN SVI interface.
- Extending Location Based Filtering (WLAN/SSID, AP Name, AP Location, Regular expression matching on AP Name/Location)
- mDNS AP Support
- mDNS Support for Wired Guest Access
- Separation between Guest and non-Guest wireless and wired service instances discoveries and sharing
- mDNS support on RLAN Profile
- Features added in IOS-XE release 17.3.1
- mDNS Gateway support on the Flex Connect APs
- Maximum of 1024 clients or services supported per FC site.
- Maximum of 64 services will be included on the database and each service includes up to 16 service providers.
- A “master-service-list” from the mDNS Service Policy created the C9800 WLC can be used to **define the services** supported on the Flex Cache Database.



## mDNS Gateway in Flex Connect Mode

In IOS-XE release 17.3.1 mDNS Gateway database and function implemented on each FC APs such that APs can distribute mDNS service information across IP subnets to client without the need of the controller other than the initial phase of configuration or monitoring.

FC master mDNS APs then will export the mDNS data to the controller for central point of management. Flex mDNS support will be implemented on the COS Wave-2 and Wi-Fi 6 APs. This means:

- Wave-1 IOS APs can't run the mDNS GW functionality on them and will support **Bridging** mode only.
- Clients behind the IOS APs can only see the Service Providers (SP) in their VLAN.
- When Clients behind IOS AP advertise Service, COS APs in the same site tag will identify it as Wired service and cache it.
- Clients behind the COS APs can see the services behind the IOS APs as wired services.

AP will snoop any mDNS packet coming in from Wireless and Wired SPs and take appropriate actions. AP acting as mDNS gateway consumes the mDNS response and builds service-string database. All other FC APs will build their service and service-string database independently of each other but will synch the databases with each other.

For Wired service-provider mDNS response, only one auto selected **Master-AP** will build service and service-string database and respond to Wired clients queries. One selected Master-AP is responsible to periodically sync mDNS data to the controller for monitoring purposes. If the Master-AP goes down, then the image master flow will select another master in the FC group. When all APs are connected to the WLC, there is only one Master-AP because when the WLC designates a new master AP, it always sets old master AP to non-master AP.

When the master AP disconnects and goes into standalone mode, the WLC will select another AP as new master AP, so there may be more than one master APs in the flex group, at that case, both new master AP and old master AP work and both of them can respond to the query from wired clients since the duplicated responses are harmless. Again, only Master-AP will serve wired service-provider, VLANs on which wired service-provider are connected and has to be configured on the AP Trunk port.



The mDNS GW would use the L2 multicast designated for the mDNS packets to communicate among the APs in the given flex group to synch up cached databases and control messages. If an AP goes down, all the service-providers connected to that AP will be invalidated by other APs in the Flex Connect Group.

The AP would forward the mDNS service advertisements in the client VLAN itself and all the APs would be able to see it independent of their management VLAN. The client VLANs are expected to be available on all the APs in the flex group to support the roaming within the flex group. The multicast tunnel will include 3 types of messaging to ensure all APs have the same copy of the full mDNS service-provider Database:

- **Service provider addition:** When an AP discovers a new service or an update from an existing service provider and its services, AP will multicast the new information to all APs in the group through the multicast tunnel.
- **Service provider deletion:** When a wireless service provider goes away or the services had timeout, AP will multicast the new information to all APs in the group through the multicast tunnel. In addition, when a WLAN is disabled or a radio is disabled, the wireless service provider clients on the WLAN or radio will go away. In the case, AP knows that certain resource record data is invalid (for example, wireless client service provider has gone), the AP can send an unsolicited Multicast DNS response packet.
- **AP periodic heart beat:** AP will periodically send out heart beat message to indicate their presence in the group. If heart beat from an AP is missing for certain period, APs will delete all entries for service providers attaching to the missing AP. If an AP receives heartbeat message from an AP for very first time, it considers this as new AP joining the network, in this case, AP proactively sends a cache update message to the new AP so that new AP can be in sync with all the other APs. Heartbeat interval by default is 1min.
- **Active query:** timer enables pro-active learning of services. Active query timer is configured on controller. Default interval is 30 mins. AP sends out multicast query to all the service providers it learnt to either learn new services or validate existing services.
- **Visibility Element:** Master mDNS Flex AP is responsible to export mDNS data (cache/stats) to the controller periodically. Default timer interval to update stats and cache is 1 min. This interval is configurable from controller. Update messages are sent using IAPP message format

Each AP knows all the service (service is represented by service-string) present at other AP. When wireless Client send mDNS query, AP will respond with all the service which are present in the network.

*NOTE: Services could be across L2 domain. If Client delete happens, all services associated with that client will be deleted.*

When Wired client sends mDNS query, only Master AP will respond with list of all service (This is because AP cannot apply the policy on Wired clients). Master AP will only respond with the wireless client that in the same VLAN as the wired client.

*NOTE: AP does not provide mDNS service information from outside subnet or wired devices to a wired client.*

## Modes of mDNS Gateway FC AP Operation

Just like the mDNS GW on the C9800 controller, the mDNS gateway on AP can function in 3 different modes. Mode configurations will be pushed from controller to AP (mode configuration is per WLAN):

- **mDNS gateway (mDNS snoop) mode:**
  - AP will snoop the mDNS packet.
  - AP provides gateway functionality.
- **mDNS drop mode:**
  - AP will neither snoop nor learn the mDNS packet.
  - This is like dropping mDNS packet, so that client cannot discover service-providers.
  - Wired client can still discover wired service-provider, provided they are in same VLAN.
- **mDNS bridge mode:**



- This is the default mode of mDNS gateway.
- In this mode, AP will forward the mDNS packet as if AP doesn't care about packet type.
- For upstream multicast mDNS packet (from wireless side), AP will forward the packet as multicast on wired with tagging a VLAN mapped by the WLAN, and wireless side.
- For downstream multicast mDNS packet (from wired side), if the VLAN is mapped by the WLAN working in bridge mode, AP will forward the packet as multicast on wireless side for that particular WLAN. If there are more than one WLAN mapped to the same VLAN, the mDNS packet will be forwarded to WLANs which are in bridge mode, and meanwhile, the mDNS gateway will play the role if there are WLANs in gateway mode.

## Discovering mDNS Services on the Controller and Flex APs

On the Controllers, with AP in Local Centrally switched mode If the mDNS **gateway is enabled**, all ingress mDNS packets received from the wired network on a L3 interface (SVI or physical) would be intercepted by the Controller software and processed. If ingress mDNS packets are received from wireless network (through CAPWAP interfaces), then they are automatically processed. C9800 will snoop all mDNS Query packets from clients but will not forward the same on the air or on the wired network.

*Note: mDNS discoveries are not shared between mDNS Local and mDNS Flex.*

**mDNS Local mode** -services discovered and cached on the controller configured with mDNS gateway.

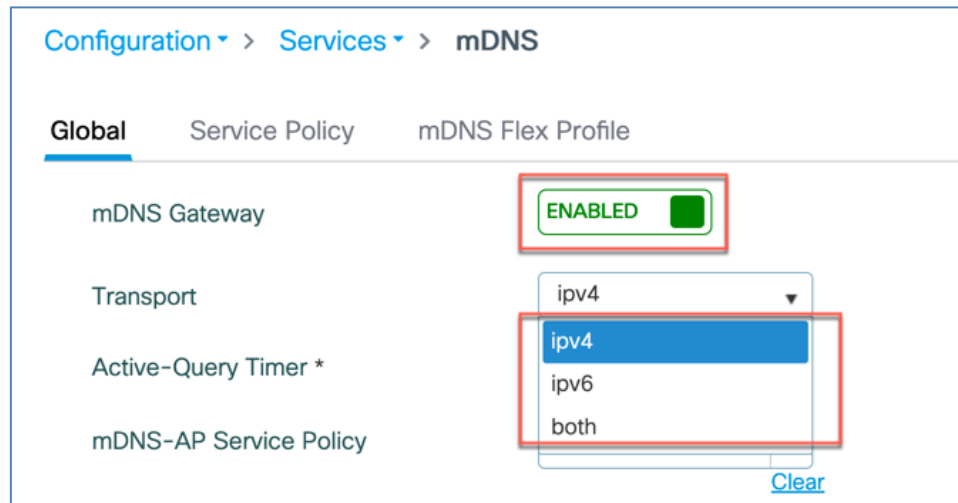
**mDNS Flex Mode** - services discovered and cached on the Flex APs synched among all Flex APs in the site.

In Flex mode when mDNS gateway is enabled on the APs, the AP will snoop any mDNS packet coming in from Wireless and Wired SPs and take appropriate actions. APs in the Flex group will cache the mDNS responses and builds service-string database. FC APs will build their service and service-string database independently of each other and then synch-up.

On the C9800 and Flex Mode AP, mDNS Gateway configuration is done at the global configuration level and/or at the WLAN level

## mDNS on the Global Level

- mDNS gateway on C9800 or FC-AP is enabled/disabled at the global level
- Bridging is the default behaviour at the Controller and FC-AP global level and at the WLAN level
- mDNS gateway if enabled, will receive wired client's mDNS packet as well
- if mDNS gateway is not enabled, the packet will not be processed by the Controller or FC-AP and will be either locally bridged, if bridging is configured, or dropped based on the configuration
- As shown below the mDNS Gateway supports IPv4, IPv6 or both Transports



In **Local Bridging mode** (when mDNS GW is not enabled), packets with mDNS multicast IP and multicast mac will be sent on multicast CAPWAP tunnel. A multicast CAPWAP tunnel is a special CAPWAP tunnel used for reducing the number of copies of multicast packet that are required to be generated for each AP CAPWAP tunnel. Sending packets on the multicast CAPWAP tunnel requires the outer IP header to be destined to the multicast CAPWAP tunnel's address, which all APs are subscribed to.

In Local mDNS bridging mode mDNS packet will be L2 bridged on to the VLAN. Packets with multicast mac destination will be flooded in the VLAN to all wired and wireless interfaces. Known mac destination unicast packet will be sent to a particular interface, and unknown mac destination unicast packet will be flooded in the VLAN to all wired interfaces.

In **Flex Bridging mode**, AP will forward the mDNS packet as if AP doesn't care about packet type. For upstream multicast mDNS packet (from wireless side), AP will forward the packet as multicast on wired with tagging a VLAN mapped by the WLAN, and wireless side. For downstream multicast mDNS packet (from wired side), if the VLAN is mapped by the WLAN working in bridge mode, AP will forward the packet as multicast on wireless side for that particular WLAN. If there are more than one WLAN mapped to the same VLAN, the mDNS packet will be forwarded to WLAN which are in bridge mode, and meanwhile, the mDNS gateway will play the role if there are WLAN in gateway mode.

## mDNS Gateway on the WLAN Level

- WLAN can be configured in Bridge, Gateway or Drop mode. By default it would be in bridge mode
- mDNS gateway can be disabled, Drop Mode, on specific WLAN if user does not want to learn or query for any mDNS services

**Edit WLAN**

⚠ Changing WLAN parameters while it is enabled will result in loss of connectivity for clients connected to it.

General   Security   **Advanced**   Add To Policy Tags

|                               |                                     |                      |                          |
|-------------------------------|-------------------------------------|----------------------|--------------------------|
| Coverage Hole Detection       | <input checked="" type="checkbox"/> | Universal Admin      | <input type="checkbox"/> |
| Aironet IE                    | <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            | Gateway                  |
| Max Client Connections        |                                     | Off Channel Scanning | Gateway                  |
|                               |                                     |                      | Drop                     |

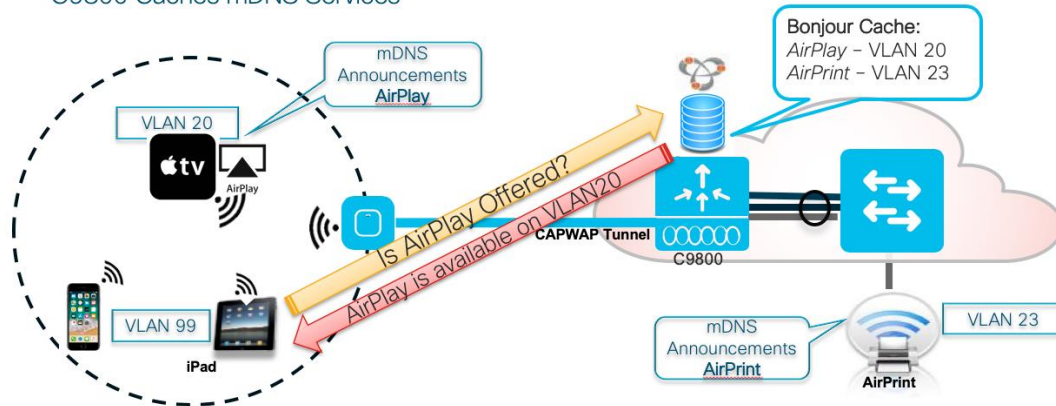
## Caching mDNS Services on the Controller

mDNS Gateway when enabled on the controller has a proxy function that is primarily used to reduce the multicast on the “AIR” interface; with many new devices adopting mDNS as a service discovery mechanism it becomes imperative to support it so that the AIR interface could be effectively used. In Flex mode the caching services will be performed on the FC AP itself.

On Wired side we have link-local multicast which makes the service discovery learning on the same VLAN seamless, mDNS gateway is used to learn services from other VLANs. In the wireless case since multicast has to be blocked, mDNS gateway has to play that additional role of advertising for local VLAN also.

- Wireless and Wired Clients that require mDNS services will receive those services in Unicast from the controller
- When a query comes from a **wired** client, it will be responded with matching service instances from wireless for that VLAN
- When a query comes from a **wireless** client, it will be responded with matching service instances learned from both wired and wireless VLANs.
- The response to the queries is unicast for both wired and wireless client

## C9800 Caches mDNS Services



Services once learnt need to be refreshed, mDNS gateway uses a mechanism called Active Query (AQ) to do this, it sends multicast query and waits for responses. For the wireless this needs to be avoided as much as possible.

- A **multicast** AQ that queries all clients periodically every 15-120 minutes, default is 30min.
- A **unicast** per service AQ is sent when there is approximately 15-20 min of the service TTL is left

## mDNS Gateway Service Policy

mDNS **service policy** is a construct that can contains one or a combination of following and will be used for service filtering while learning services or responding to queries

- Service type
  - Service type can be used in limiting / filtering the service types that are learnt and the queries that are responded
- WLAN or VLAN the advertisement was learnt
  - Gateway Shall respond with queries from same WLAN plus services learnt from other WLAN's.
  - Gateway could also have a policy to only respond for a query for the service instance learnt from a specific WLAN
- Location based policy could be based on the following
  - LSS (Location Specific Services) Based Filtering

Only service instances from the client querying neighboring APs will be given back to the querying client and service instances from the rest of the APs will be filtered. Neighboring AP list will be based upon Cisco RRM data base. When mDNS service advertisement is received and mDNS cache entry is added/updated, Radio/WTP MAC address of the AP associated with the service provider/client will also be added/updated in cache. When mDNS query is received, it's associated AP Radio/WTP MAC address is used to query the RRM data base to get the AP neighbor list. Subsequently response formulation to client query

would filter the services from mDNS cache by matching the AP neighbor list retrieved from RRM DB. LSS filtering applies only to wireless service instances in mDNS cache. Wired services can't be subjected to LSS based filtering and will always be responded unless configured otherwise.

- Site Tag (aka AP-Group) Based Filtering.

Site Tag is a construct introduced in C9800 IOS-XE and is a way to group AP(s). Service instances learned from service provider in that site-tag will be responded if query is received from clients associated to any AP in that same site-tag. This option of site-tag based filtering offers more control to administrator. Site-tag associated with the service provider AP will also be added/updated in mDNS cache entry. Wired service instances can't be subjected to site-tag based filtering and will always be responded unless configured otherwise.

- AP Name Based Filtering

This option offers to filter the service instances to the granularity of AP name. i.e Only service instances that are matching the AP name string (regular expression) as that of query client are returned and rest everything else is filtered. For instance group of AP's can be named with a conventions like "Bld1-Eng-4flr", in such cases service that are matching (regular expression) the AP name should be provided for eg : if user mentions location as "Bld1\*" then matching service instance learnt from the AP having name starting with "Bld1" needs to be provided. Needless to say, in this case as well, wired service instances can't be subjected to AP location-based filtering and will always be responded unless configured otherwise.

- mDNS AP group

In some deployments AP group or site-tag can form a very large set of AP's and clients, in such cases it would be preferred to go with mDNS-AP model where user would create a mDNS-AP group and map as set of APs to it

- SSID

For some SSID's like guest WLAN, service learnt from the same SSID needs to be given

- AAA override

In mDNS Local mode with GW configured on the controller the policy configured under wireless policy profile can be overridden by AAA configs that are downloaded as part of client authorization. Only the mDNS service policy name will be part of the cisco av-pair (mDNS-ServicePolicy-name) that downloaded. User is required to configure a local service policy with that name. The AAA override is not supported in Flex Mode with mDNS GW configured on the AP itself.

**Note:** Before IOS-XE release 17.1 location-based policy was based on LSS model (RRM based) Site Tag, in the release 17.1 and above it is based on LSS, Site Tags and multiple Location Based filters documented below.

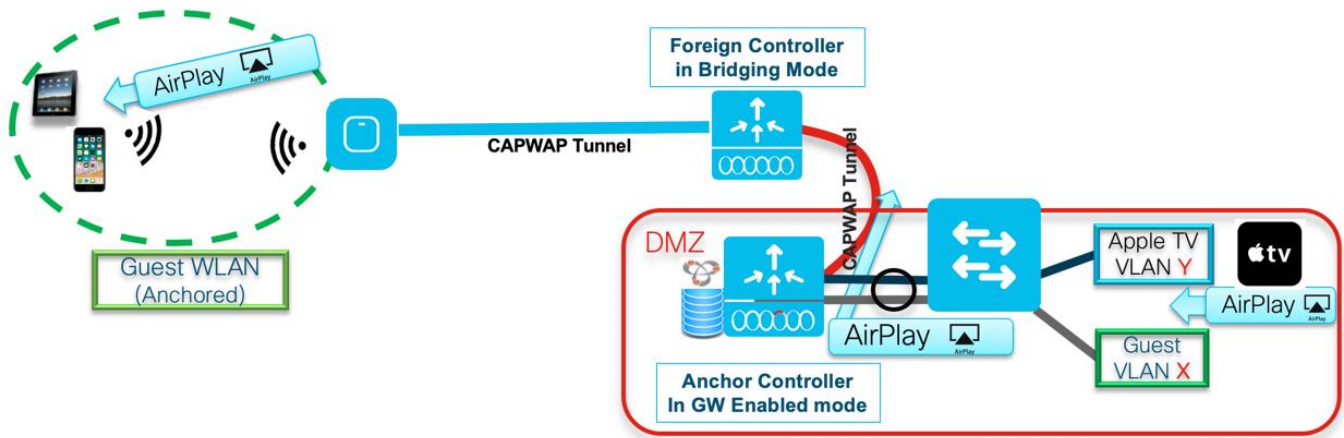
The screenshot displays the Cisco configuration interface for mDNS. On the left, the breadcrumb navigation shows 'Configuration > Services > mDNS'. Below this, the 'Service Policy' tab is selected and highlighted with a red box. The 'Service List' section shows a table with columns 'Service Policy Name' and 'Location'. The first entry, 'tme-bonjour-SP1', is selected and highlighted with a red box. The 'Edit Service Policy' panel on the right shows the configuration for 'tme-bonjour-SP1'. The 'Service Policy Name\*' field is set to 'tme-bonjour-SP1'. The 'Service List Input' and 'Service List Output' fields are both set to 'tme-bonjour-sl-in' and 'tme-bonjour-sl-out' respectively. The 'Location' dropdown menu is open, showing a list of options: 'ssid', 'None', 'ap-location', 'ap-name', 'lss', 'regex ap-location', 'regex ap-name', 'site-tag', and 'ssid'. The 'ssid' option is selected and highlighted with a red box.

## mDNS gateway with Guest Anchor support

mDNS gateway functionality will be supported in guest anchor deployment where clients on guest WLAN with guest anchor enabled will not be responded with any services or cache from export foreign WLC even if export foreign has services present in mDNS cache.

## Foreign Controller in Bridging Mode and Anchor Controller in GW enabled mode

- The guest WLAN will be able to query mDNS service announcements to the Anchor controller. Anchor Controller will handle all the mDNS traffic
- mDNS traffic (query & advertisements) received on the Foreign Controller will get bridged over the CAPWAP mobility tunnel towards Anchor Controller
- Clients on guest WLAN with guest anchor enabled will be responded with any services or cache from Anchor if the services are present in the mDNS cache



## Foreign Controller in GW Mode Enabled and Anchor Controller in GW or Bridging mode

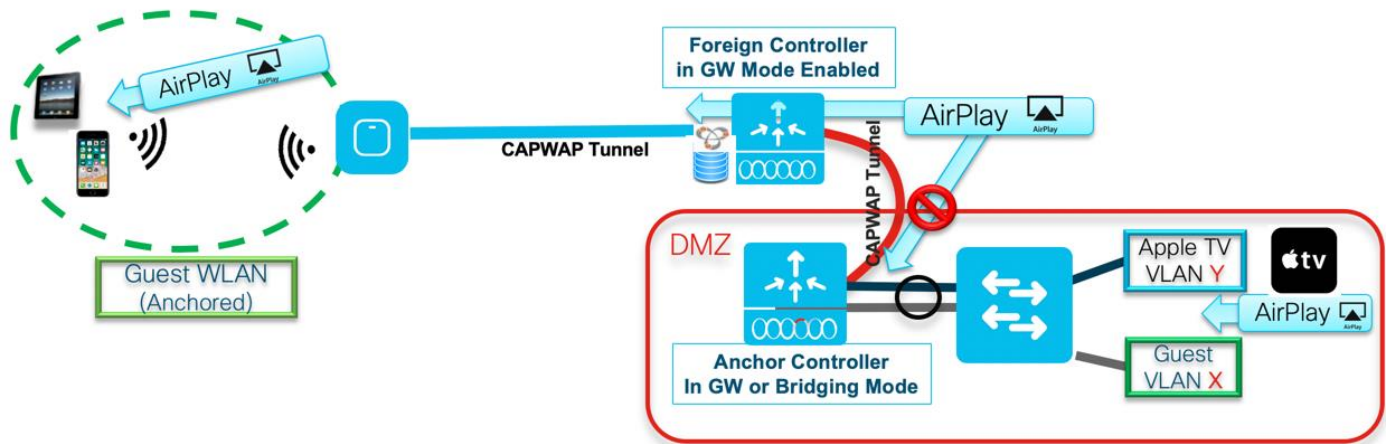
Begin with release 16.12.1, irrespective of the mDNS mode configured on the Anchor Controller, mDNS traffic coming from wireless clients gets terminated on the mDNS Gateway on the Foreign Controller itself. mDNS traffic from the guest wireless clients is subjected to the local mDNS policies as configured on the local Foreign controller. In this mode, mDNS traffic from the guest wireless clients is not forwarded to the anchor controller. In this case mDNS Gateway on the Foreign Controller will not be able to see mDNS services from any other controller (i.e. Guest Anchor Controller or any other Foreign Controller).

This behavior is in line with what is currently support in AireOS mDNS Guest solution

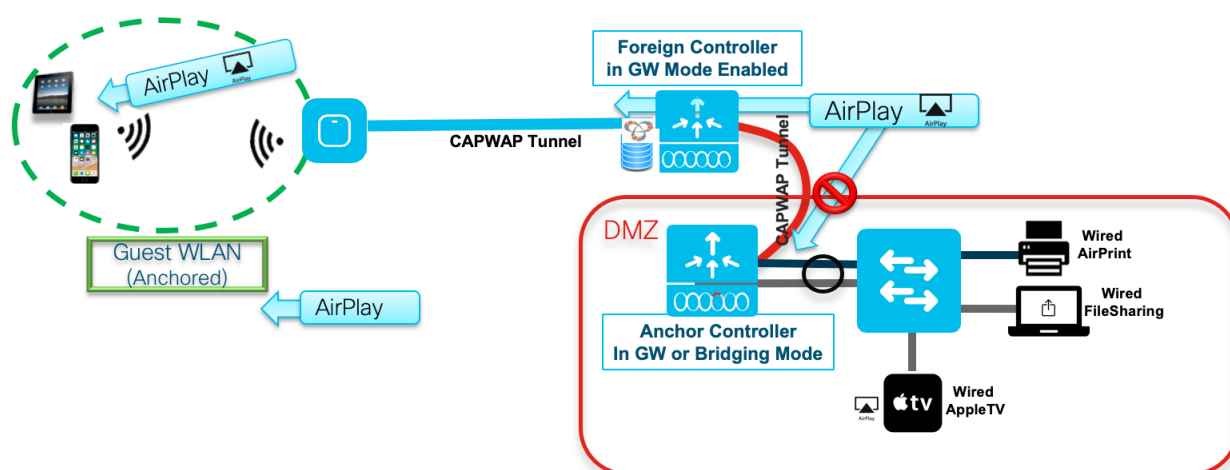
- The guest WLAN will be able to query mDNS service announcements to the Foreign controller.
- mDNS traffic, query & advertisement, will get terminated on the Foreign Controller itself
- mDNS services/advertisements will be learned on the Foreign Controller and mDNS queries received from Wireless Guest Clients will get responded from the local mDNS cache of the Foreign
- mDNS traffic is not tunneled to Anchor Controller.

**Note:** LSS will not work in case of mDNS Gateway is operational on Guest Anchor Controller.





Begin with release 17.1, Wired Guest support was added in C9800 controllers. The new mDNS enhancement added support for wired guests as illustrated below. mDNS support for wired guest is now on par with the wireless clients.

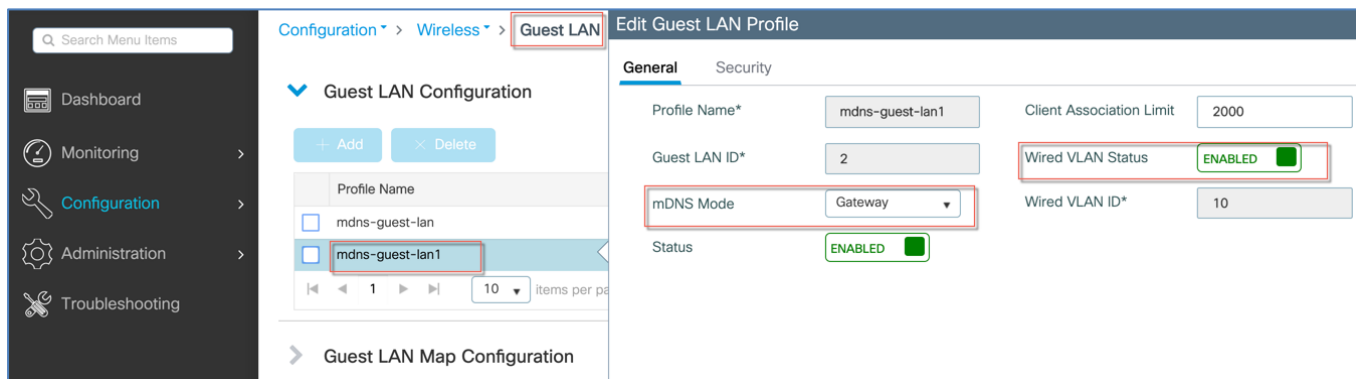


As part of this new enhancement in rel 17.1 new guest-lan profile specific CLI/WebUI is added to allow the administrator to configure mDNS mode on the guest-lan profile. This will be same CLI/WebUI configuration as what is present currently in WLAN profile mode for setting the mDNS mode. The default mDNS mode on the guest lan (without any explicit configuration) will be mDNS Bridging mode.

In a mixed (guest & non-guest) deployments, prior to release 17.1 mDNS gateway solution did not restrict the mDNS service discovery within the guest WLAN(s) & LAN(s). Basically, guest user was able to query and see the mDNS services from the non-guest WLAN(s). This behavior is not desirable and is enhanced. In IOS\_XE 17.1, this behavior is changed in such a way that queries from the guest (wired/wireless) users will be responded back from the mDNS service instances which are learned on guest WLAN(s) & LAN(s). mDNS service instances from non-guest WLAN(s) will not be shared. There will not be any change in the way wired mDNS service instances are shared with the querying clients. i.e guest users will be able to get all the wired mDNS service instances.

**Note:** Location based mDNS service instance filtering does not work in case of mDNS Gateway enabled on Guest Anchor. This is existing behavior what we have in prior releases and there is no change in this behavior as part of the release 17.1.1.

Guest-LAN configuration example is shown below.



Same Guest-LAN configurations are available from the CLI mode, see configuration examples below.

| FOREIGN                                                                                                                                                                                                                                                                                                                                                                                                                                                       | ANCHOR                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre># wireless profile policy wireless profile policy wired_mdns_guest_pol central switching mobility anchor 1.1.1.1 priority 3 mdns-sd service-policy default-mdns-policy  no shutdown  # creating Wired Guest profile guest-lan profile-name mdns_wired_guest 1 wired- vlan 10 no security web-auth no shutdown  # mapping Wireless Profile to wired guest wireless guest-lan map example-map guest-lan mdns_wired_guest policy wired_mdns_guest_pol</pre> | <pre># wireless profile policy wireless profile policy wired_mdns_guest_pol central switching mobility anchor mdns-sd service-policy default-mdns-policy vlan 70 no shutdown  # no wired-vlan used on anchor guest-lan profile-name mdns_wired_guest 1 client association limit 2000 no security web-auth no shutdown  # no map required, unless required by AVC</pre> |

The same new mDNS configuration options are now available under the RLAN as well. A new remote-lan profile specific CLI/WebUI is added to allow the administrator to configure mDNS mode on the remote-lan profile. This will be same CLI/WebUI as what is present currently in WLAN profile mode for setting the mDNS mode. The default mDNS mode on the remote lan (without any explicit configuration) will be mDNS Bridging mode. Currently even though we support learning of mDNS services from clients connected to remote-lan, we always learn the mDNS services if the mDNS gateway is configured globally.

Below are illustrations of the RLAN profile and RLAN Policy creation for mDNS services.

Configuration > Wireless > Remote LAN

RLAN Profile RLAN Policy

+ Add - Delete Enable RLAN Pr

Number of RLAN Profiles selected : 0

Status

1 10 items per page

Edit RLAN Profile

General Security

Profile Name\* mdns-guest

RLAN ID\* 1

Status ENABLED

Client Association Limit 0

mDNS Mode Gateway

Configuration > Wireless > Edit RLAN Policy

RLAN Profile RLAN Policy

+ Add - Delete

Name rlan-mdns-policy

1 10

General Access Policies Advanced

Violation Mode replace

Session Timeout (sec) 1800

mDNS Service Policy tme-bonjour-SP1 Clear

User Personal Network

Status

Drop Unicast

AAA Policy Params

AAA Override

AAA Policy Name default-aaa-policy

Split Tunnel Parameters

Status

ACL Name Not Configured

Override Status

Gateway Address xxx.xxx.xxx.xxx

Netmask Address xxx.xxx.xxx.xxx

DHCP

IPv4 DHCP Required

DHCP Server IP Address xxx.xxx.xxx.xxx

## Mobility Support for the mDNS Gateway

### Intra Controller Roaming

Client can roam across AP(s) which are bound to same C9800 controller. In this case, everything remains same other than the AP association of the client. mDNS gateway within Controller instance will register for and get the CLIENT RUN event. While processing CLIENT RUN event, mDNS gateway will check if there is a change in associated AP, If there is a change in the associated AP, mDNS gateway will update its mDNS local cache database table by updating all the cache entries from this specific client with the new AP Radio/WTP MAC and it's corresponding site-tag.

### Inter Controller L3 Roaming

In this scenario, the new eWLC controller where the client has roamed will not have the same VLAN/subnet available and the new controller will become the foreign controller whereas the old controller will remain as the anchor controller. Keeping in mind the location aware mDNS service(s), mDNS query and/or advertisement packets on the foreign controller will be punted and processed in foreign controller instead of forwarding the same to anchor controller. mDNS service-instance cache entries for the roamed client will be cleared from the anchor controller.

## Common Local and Flex Mode mDNS Configuration Steps on C9800

mDNS Wired Services with mDNS gateway on the controller don't require multicast services to be enabled, for the wireless services to be advertised the Multicast has to be enabled on VLAN.

With deployment of mDNS gateway the mDNS Services won't flood subnet with mDNS announcements or queries

mDNS services with Global mDNS Gateway and WLAN mDNS are enabled.

- mDNS Gateway enabled will reduce the multicast traffic on Air
- In the Bridging mode (mDNS Gateway disabled) the multicast will be flooded on the VLANs
- mDNS gateway is used to learn services from other VLAN's
- mDNS-AP is used to learn wired mDNS services that are not seen by the controller (Local mode support only).
- mDNS gateway has to play an additional role of advertising for local VLAN as well
- Services once learnt need to be refreshed, mDNS gateway uses a mechanism called active query timer to do this
- A multicast AQ ( Active Query) that queries all clients periodically, default is 30min.
- A unicast per service AQ is sent when there is approximately 15-20 min of the service TTL is left
- From the Global configuration screen you can also elect an IP transport as IPv4 or IPv6 or support for both at the same time

The screenshot shows the 'Configuration > Services > mDNS' page. The 'Global' tab is selected. The configuration items and their values are:

| Configuration Item     | Value   |
|------------------------|---------|
| mDNS Gateway           | ENABLED |
| Transport              | ipv4    |
| Active-Query Timer *   | 30      |
| mDNS-AP Service Policy | Default |

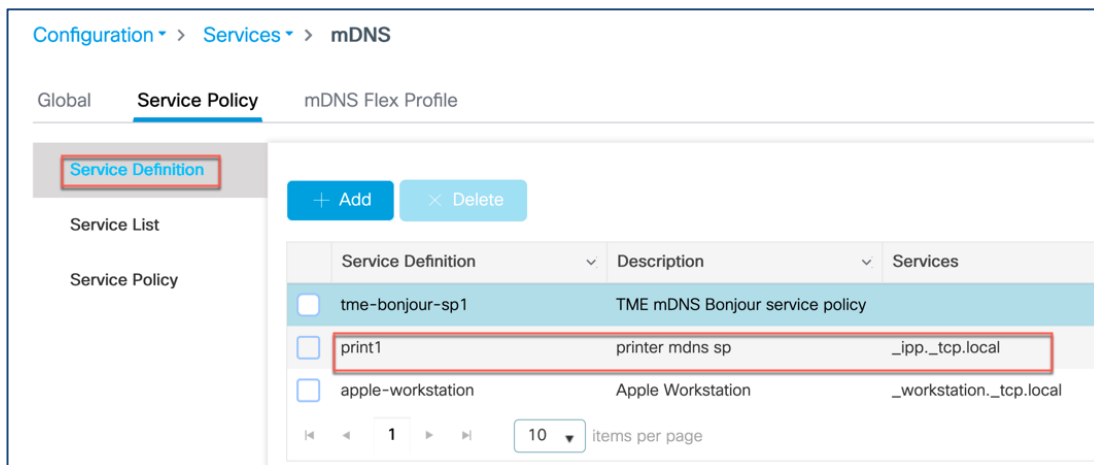
The 'Transport' dropdown menu is open, showing options: 'ipv4' (selected), 'ipv6', and 'both'. A 'Clear' button is located at the bottom right of the dropdown menu.

When mDNS gateway is enabled, mDNS packets will be consumed by the mDNS gateway and clients or device will be deprived of learning this service. In-order to share the service with the device and to provide ease of configuration to the administrator, few standard Service Types will be shared by **default** on the wireless network. The list comprising of these standard service types is termed as Default Service Policy and would have the following service types as listed in the table below.

| Default name                 | mDNS service-type          |
|------------------------------|----------------------------|
| Airplay                      | _airplay._tcp.local        |
| AirTunes                     | _raop._tcp.local           |
| HomeSharing                  | _home-sharing._tcp.local   |
| Printer-LPD                  | _printer._tcp.local        |
| Printer-IPP                  | _ipp._tcp.local            |
| Printer-IPPS                 | _ipps._tcp.local           |
| Printer-SOCKET               | _pdl-datastream._tcp.local |
| Googlecast                   | _googlecast._tcp.local     |
| iTuneWirelessDeviceSharing_2 | _apple-mobdev2._tcp.local  |

mDNS custom services definitions can be optionally added on the Global mDNS Level. There are also pre-loaded definitions, the preloaded service-definitions cannot be modified or deleted.

User can configure a user-friendly name for mDNS services-type as shown below:



Service Definitions and the Services Lists are created on the Global mDNS Interface.

Services Types used can be one that exist in the Default Service Type or “Custom created Services” elected from the Master Service Type list. Once the custom service types are defined you can create a custom Service List from them as illustrated in the examples below.

| Service definition | Service Name                                       |
|--------------------|----------------------------------------------------|
| <b>fax</b>         | _fax-ipp._tcp.local                                |
| <b>roku</b>        | _rsp._tcp.local                                    |
| <b>airplay</b>     | _airplay._tcp.local                                |
| <b>scanner</b>     | _scanner._tcp.local                                |
| <b>spotify</b>     | _spotify-connect._tcp.local                        |
| <b>airtunes</b>    | _raop._tcp.local                                   |
| <b>airserver</b>   | _airplay._tcp.local , _airserver._tcp.local        |
| <b>apple-rdp</b>   | _afpovertcp._tcp.local , _net-assistant._tcp.local |
| <b>web-server</b>  | _http._tcp.local                                   |
| <b>homesharing</b> | _home-sharing._tcp.local                           |
| <b>printer-ipp</b> | _ipp._tcp.local                                    |
| <b>printer-lpd</b> | _printer._tcp.local                                |
| <b>workstation</b> | _workstation._tcp.local                            |

|                                       |                                                            |
|---------------------------------------|------------------------------------------------------------|
| <b>printer-ipp</b>                    | _ipp._tcp.local                                            |
| <b>apple-homekit</b>                  | _hap._tcp.local , _homekit._ipp.local                      |
| <b>apple-keynote</b>                  | _keynotecontrol._tcp.local , _keynotepair._tcp.local       |
| <b>amazon-fire-tv</b>                 | _amzn-wplay._tcp.local                                     |
| <b>apple-airprint</b>                 | _ipp._tcp.local , _universal._sub._ipp._tcp.local          |
| <b>printer-socket</b>                 | _pdl-datastream._tcp.local                                 |
| <b>apple-continuity</b>               | _companion-link._tcp.local                                 |
| <b>apple-file-share</b>               | _afpovertcp._tcp.local                                     |
| <b>apple-timecapsule</b>              | _adisk._tcp.local , _afpovertcp._tcp.local                 |
| <b>google-chromecast</b>              | _googlecast._tcp.local                                     |
| <b>apple-itunes-music</b>             | _daap._tcp.local                                           |
| <b>apple-itunes-photo</b>             | _dpap._tcp.local                                           |
| <b>apple-remote-login</b>             | _sftp-ssh._tcp.local , _ssh._tcp.local                     |
| <b>apple-screen-share</b>             | _rfb._tcp.local                                            |
| <b>apple-remote-events</b>            | _eppc._tcp.local                                           |
| <b>phillips-hue-lights</b>            | _hap._tcp.local                                            |
| <b>apple-itunes-library</b>           | _atc._tcp.local                                            |
| <b>multifunction-printer</b>          | _fax-ipp._tcp.local , _ipp._tcp.local, _scanner._tcp.local |
| <b>apple-timecapsule-mgmt</b>         | _airport._tcp.local                                        |
| <b>apple-windows-fileshare</b>        | _smb._tcp.local                                            |
| <b>itunes-wireless-devicesharing2</b> | _apple-mobdev2._tcp.local                                  |

As shown in the example below a Custom Service definition was created for an Apple Workstation and the Service Type was elected from the Master List.

Once the service types and definitions have been created a Service List can be created based on the Master Services and Custom defined services as shown below in the example

The Service list is matched on IN (request, ingress filter) and/or OUT (respond, egress filter) to the queries. The list shown comprising of the service types is termed as Master Service List and is not a part of the Default Service List.

**Note:** In rel 17.1 and below Default Service List is not available in WebUI configuration display it can only be seen and configured from the CLI



```
C9800-MA21#sh mdns-sd default-service-list ←
-----
mDNS Default Service List
-----

Service Definition: airplay
Service Names: _airplay._tcp.local

Service Definition: airtunes
Service Names: _raop._tcp.local

Service Definition: homesharing
Service Names: _home-sharing._tcp.local

Service Definition: printer-ipp
Service Names: _ipp._tcp.local

Service Definition: printer-lpd
Service Names: _printer._tcp.local

Service Definition: printer-ippes
Service Names: _ippes._tcp.local

Service Definition: printer-socket
Service Names: _pdl-datastream._tcp.local

Service Definition: google-chromecast
Service Names: _googlecast._tcp.local

Service Definition: itune-wireless-devicesharing2
Service Names: _apple-mobdev2._tcp.local
```

**Note:**

- Location would be disabled on mDNS default service-policy
- This default service-policy would contain the same set of services for both IN and OUT direction
- Contents of mDNS default service-policy cannot be changed by user. However, user can create separate mDNS service-policies and associate them under wireless-policy-profile

In the Service policy we associate the Service Lists IN/OUT, Site Tag and LSS

**Note:** By default, “default-mDNS-service-policy” gets created in the system and “mDNS-AP Service Policy” will use “default-mDNS-service-list” configuration for filtering mDNS service announcement and queries from the mDNS-AP located somewhere in the different subnet and tunnelling services across a CAPWAP tunnel back to the controller.

Configuration > Services > mDNS

Global Service Policy mDNS Flex Profile

mDNS Gateway **ENABLED**

Transport **ipv4**

Active-Query Timer \* **30**

mDNS-AP Service Policy **default-mdns-service**

[Clear](#)

**When Location Specific Services or LSS is selected:** Only service instances from the querying client AP's neighbouring AP's will be given back to the querying client and service instances from the rest of the AP(s) will be filtered. Neighbouring AP list will be based upon Cisco RRM DB.

**When Site Tag is selected:** Site Tag is a construct introduced in C9800 IOS-XE and is a way to group AP(s). Service instances learned from Service Provider in that site-tag will be responded if query is received from clients associated to any AP in that site-tag.

Prior to release IOS-XE 17.1 mDNS Gateway implementation supported mDNS service filtering are based on LSS (RRM based) and Site-tag. In Release 17.1 location based mDNS service filtering will be extended with the following additional options.

**Based on WLAN/SSID Name selected:** Prior to release 17.1 we responded back with the services irrespective of the WLAN/SSID service belongs to. There are few scenarios/cases, where administrator would want the ability to filter the mDNS services in such a way that only service(s) which are in the same WLAN as that of the query client should be responded whereas not from all the WLAN(s)/SSID(s). In release 17.1 mDNS location-based filtering will be enhanced to filter services from the same SSID/WLAN if configured.

Quick Setup: Service Policy

Service Policy Name\* **tme-bonjour-SP1**

Service List Input **tme-bonjour-sl-in**

Service List Output **tme-bonjour-sl-out**

Location **ssid**

None  
ap-location  
ap-name  
lss  
regex ap-location  
regex ap-name  
site-tag  
**ssid**

[Cancel](#) [Apply to Device](#)

**Based on AP Name selected:** There are use cases where the requirement is to only respond with mDNS services which are learned from the same AP as that of the querying client. In release 17.1 mDNS location-based filtering is enhanced to filter services from the same AP-name if configured.

**Based on AP Location selected:** A group of APs usually share the same AP location. In other words, a particular AP Location string is usually shared by a group of APs. There are use cases, where the requirement is to only respond with mDNS service(s) which are learned from the AP(s) which share the same AP Location string as that of the querying client AP Location string.

Basically, from the user perspective, only the services which share the same AP name as that of the querying client will be responded. In release 17.1 and above mDNS location-based filtering is enhanced to filter services from the AP Location if configured.

**Based on Regular Expression matching on AP location (supported only in Local not Flex Mode):** There are use cases where the requirement is to filter and match mDNS services based upon “regular expression” matching on AP Location. For Example, administrator may want to respond to mDNS query with mDNS services whose AP Location start from string “Bldg-18” and end with anything after that. In release 17.1 mDNS location-based filtering will be enhanced to filter services based upon AP Location based regular expression, if configured as shown in the screen shot below.

**Based on Regular Expression matching on AP name:** There are use cases where the requirement is to filter and match mDNS services based upon regular expression matching on AP Name. For Example, administrator may want to respond to mDNS query with mDNS services whose AP names start from string “C9120-Floor-2” and end with anything after that. In release 17.1 and above mDNS location-based filtering will be enhanced to filter services based upon AP Name based regular expression, if configured as shown in the example below.

**Note:** Wired Service Instances can’t be subjected to Site-tag, LSS or Location-based filtering and will always be responded unless configured otherwise.

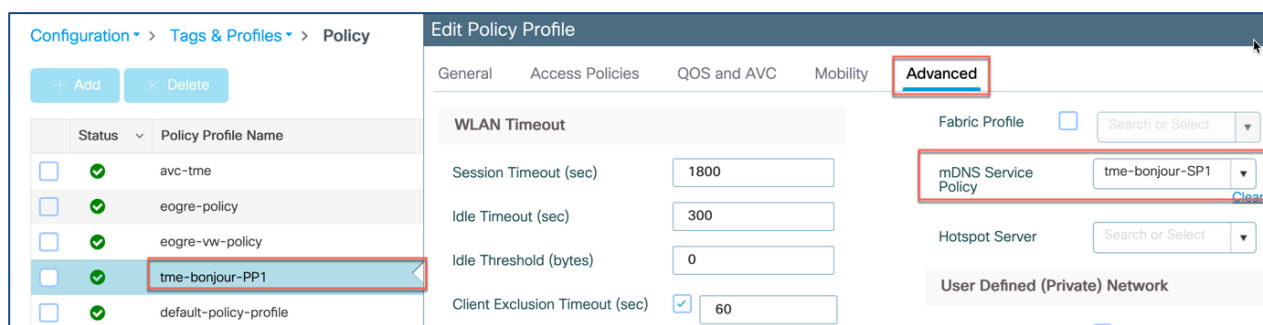
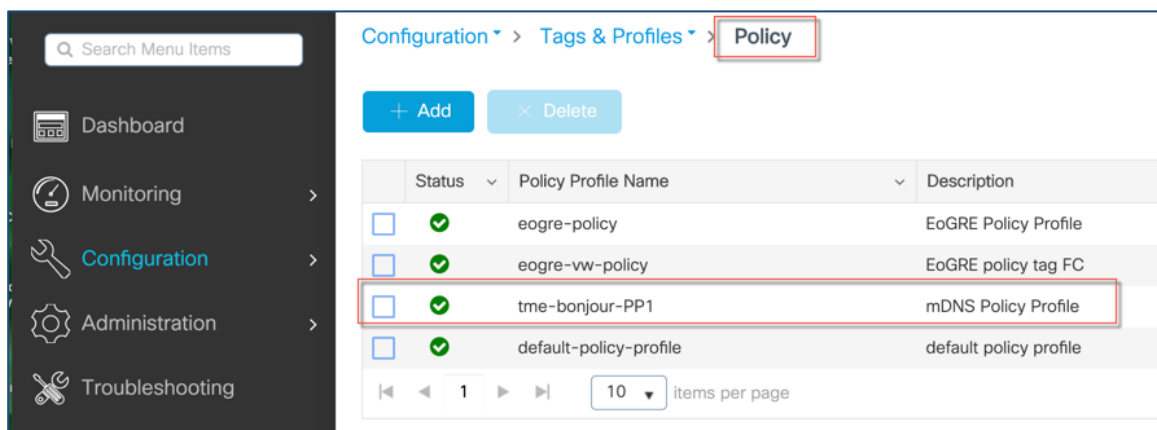
In release 17.1 and above a new exec mode show CLI will be added to provide/display service instance count for each service type. Following is the example of CLI command that is added in release 17.1 to show service-type aka PTR available.

```
!
show mdns-sd service statistics
PTR-Name      Count
_ipp._tcp.local    10
_airplay._tcp.local  1
!
```

## Configuring Policy Profile with mDNS settings

Configure Policy Profile as shown in the example below with the specific **mDNS Service Policy** configured in the previous steps.

Configuring under Configure > Tags & Profiles > Policy > Advanced mode will associate “mDNS service policy” with Policy Profile under configuration as shown in the example below.



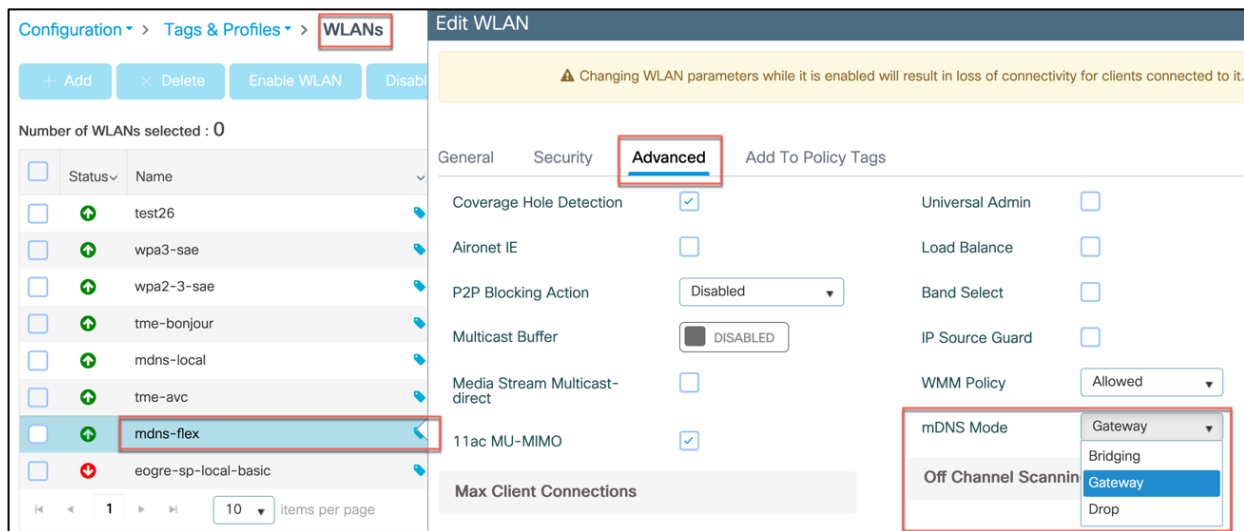
WLAN mDNS Mode is configured under WLAN tab. Different WLANs can have different Policies and Profiles, which can be also AAA overridden for more granularity.

This command is configured under WLAN and gives the flexibility to enable or disable mDNS gateway or completely disable mDNS packet processing on the WLAN:

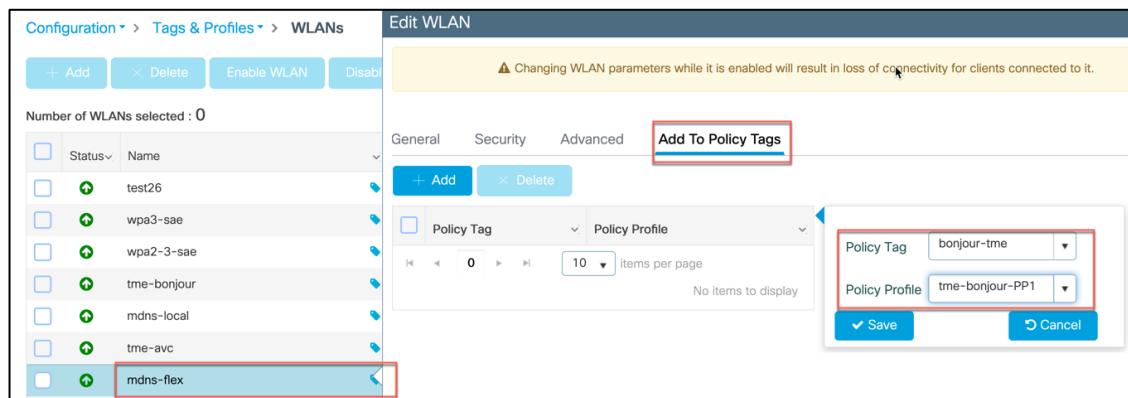
|                           |                                                                                                                                                                                                            |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>mdns-sd drop</b>       | <b>Disables mDNS function on that WLAN meaning , that all mDNS packets, announcements and queries, will be dropped on that WLAN</b>                                                                        |
| <b>mdns-sd gateway</b>    | This would be effective only when gateway is enabled at the global level otherwise WLAN will be effectively in drop mode                                                                                   |
| <b>no mdns-sd gateway</b> | No of the above commands would put back the WLAN in default bridging mode. In Bridging mode mDNS packets are flooded on both wireless and wired interfaces of the specific VLAN that WLAN is connected to. |

## Mapping Policy Tag with mDNS Policy Profile on the WLAN

Enable mDNS Gateway on the WLAN as shown below:

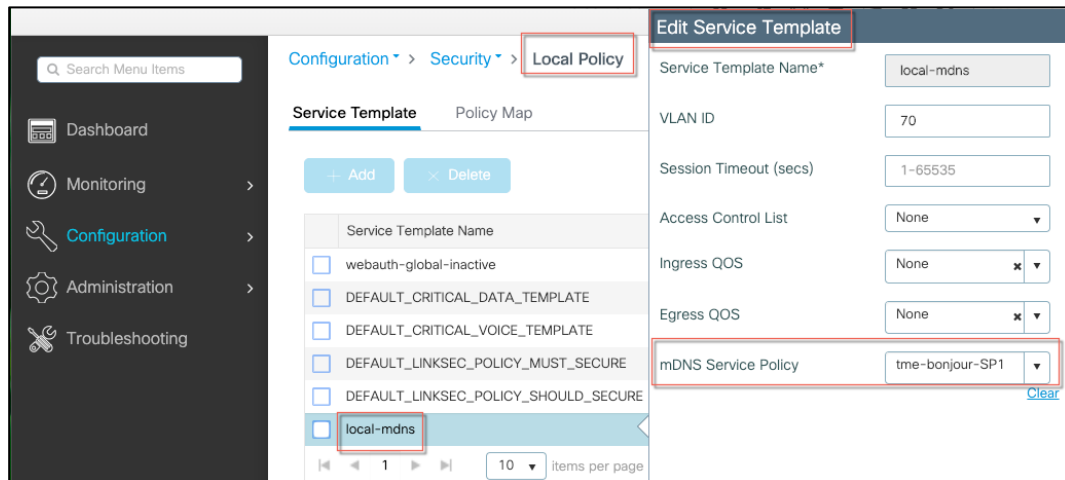


The following command under Configuration > WLAN , will Map the Policy Profile to the WLAN with mDNS GW mode enabled in the previous step. Previously configured Policy Profile and Policy Tag will be added to the selected WLAN.



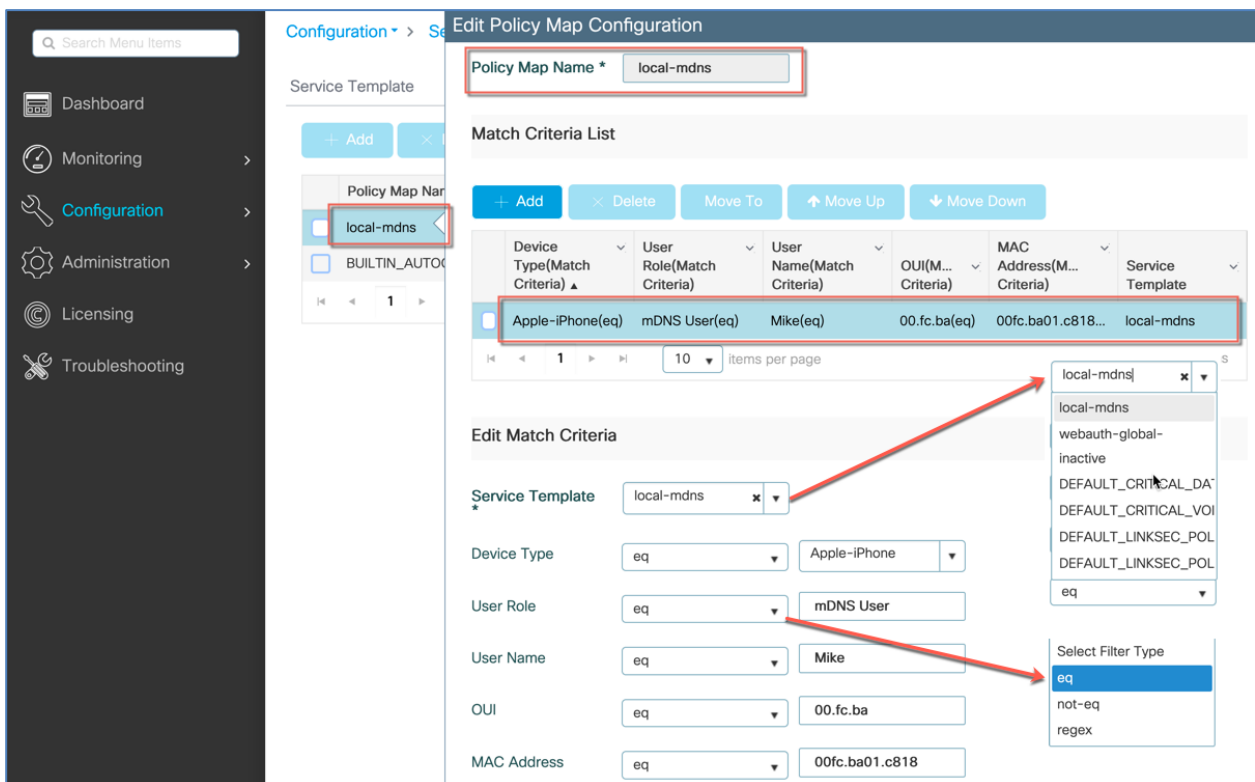
## Local or Native mDNS Policy Profile

When Administrator configures local Authentication and Authorization and does not expect to get any mDNS policy from AAA server, Administrator can configure a Local or Native profile to select a mDNS policy based on user / role / device type. Currently parameters like VLAN, VNID, OpenDNS etc. are part of this. Service-template will be enhanced to accept mDNS service policy to the list of parameters that the administrator can configure. When this Local / Native profile mapped to the Wireless profile policy, mDNS service policy will be applied on the mDNS packets that are processed on that WLAN. In the example below when Service policy "tme-bonjour-SP1" applied and there is no policy returned from AAA server the local policy "local-mdns" will take effect.



Under Policy Map the administrator can configure Service template with parameters such as

- Device Type
- User Role
- Username
- OUI
- MAC address
- Other custom criteria



**Edit Policy Map Configuration**

Policy Map Name \*

Match Criteria List

|                          | Device Type(Match Criteria) | User Role(Match Criteria) | User Name(Match Criteria) | OUI(Match Criteria) | MAC Address(Match Criteria) | Service Template |
|--------------------------|-----------------------------|---------------------------|---------------------------|---------------------|-----------------------------|------------------|
| <input type="checkbox"/> | Apple-iPhone(eq)            | mDNS User(eq)             | Mike(eq)                  | 00.fc.ba(eq)        | 00fc.ba01.c818(eq)          | local-mdns       |

items per page
 1 - 1 of 1 items

## Mapping Site and Policy Tags to AP

And the last Local or Flex configuration step, is the command under Configuration Tag>AP>Static, will map the Policy Tag to the selected APs.

**Configuration > Tags & Profiles > Edit Tags**

Policy Site RF **AP**

Tag Source **Static** Filter

Number of AP Tag mappings selected : 1

|                                     |                |
|-------------------------------------|----------------|
| <input checked="" type="checkbox"/> | AP MAC Address |
| <input checked="" type="checkbox"/> | 00fc.ba01.c818 |

AP MAC Address\*   
 Policy Tag Name   
 Site Tag Name   
 RF Tag Name

After the Site-Tag and Policy-Tags are applied to the specific AP, in the mDNS Monitoring will show under the Cached Service Type Detail



The screenshot displays the Cisco IOS-XE GUI for mDNS configuration. The breadcrumb navigation at the top shows 'Monitoring > Services > mDNS'. The left sidebar contains menu items: Dashboard, Monitoring (selected), Configuration, Administration, Licensing, and Troubleshooting. The main content area is titled 'mDNS Details' and includes a 'ClearAll' button. Below this is a table of mDNS Cache Details. The record '\_airplay.\_tcp.local' is selected and highlighted. The details for this record are as follows:

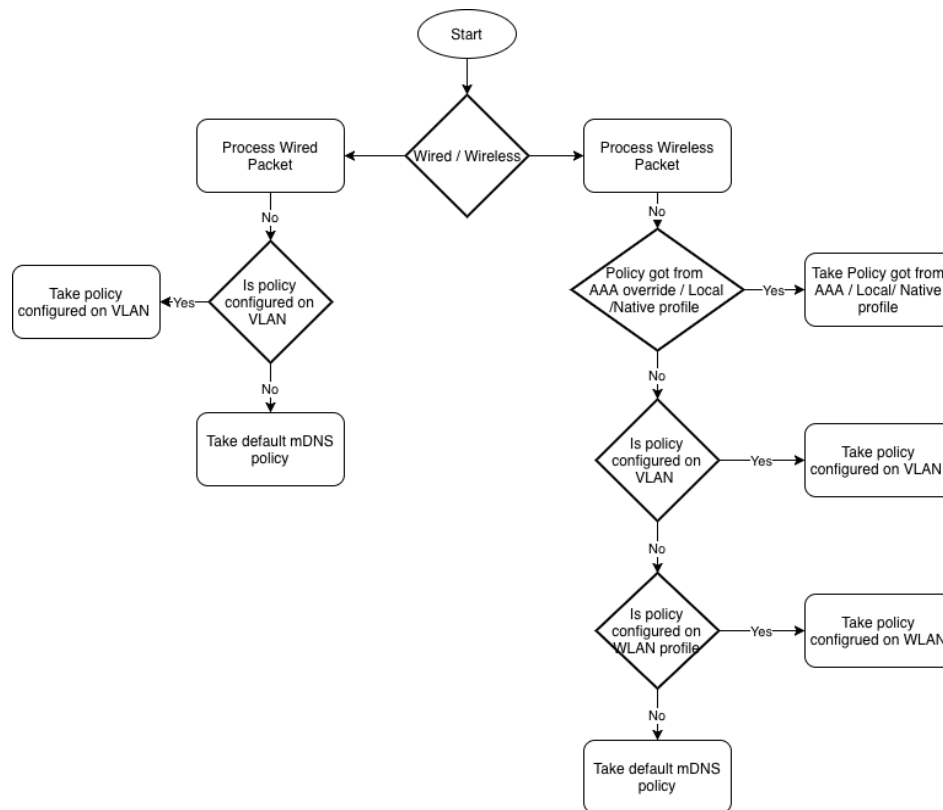
| mDNS Cache Details  |                                           |
|---------------------|-------------------------------------------|
| Record Name         | _airplay._tcp.local                       |
| Type                | DB_TYPE_PTR                               |
| TTL(sec)            | 4500                                      |
| Connection Type     | Wired                                     |
| Client Type         | WIRED                                     |
| VLAN ID             | 70                                        |
| Client MAC          | 7056.81db.cda0                            |
| AP MAC              | 0000.0000.0000                            |
| AP Ethernet MAC     | 0000.0000.0000                            |
| Service Policy      | tme-bonjour-SP1                           |
| Record Data         | Office Apple TV_2 (2)._airplay._tcp.local |
| Remaining Time(sec) | 4289                                      |

## Configuring Service Policy on the “VLAN Interface”

There are some use cases when Wireless network can have a single WLAN/SSID and at the same time can be mapped to Multiple VLAN's with the same SSID. In such cases administrator may desire to setup different conditions to mDNS packets coming on different VLAN (SVI) interface. By providing Administrator the option to configure mDNS service policy on a VLAN SVI interface, Administrator can configure different settings to the mDNS packets on per VLAN interface basis and not on per WLAN basis.

This enhancement also has an impact on handling Wired mDNS packets, Prior to IOS-XE release 17.1 fixed default service policy was applied, now we allow the mDNS service policy to be configured on the VLAN interface instead of using default service policy.

As part of this enhancement, mDNS service policy will be used as per the following hierarchy at the time of processing mDNS packet.

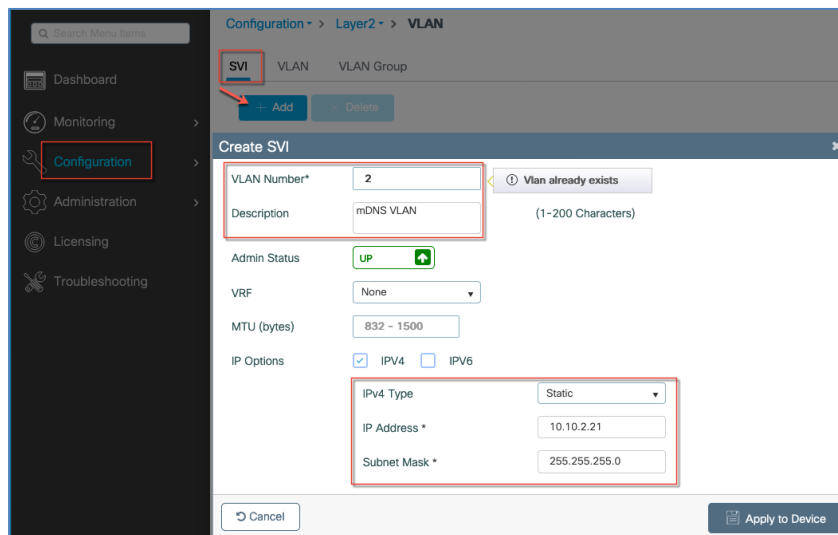


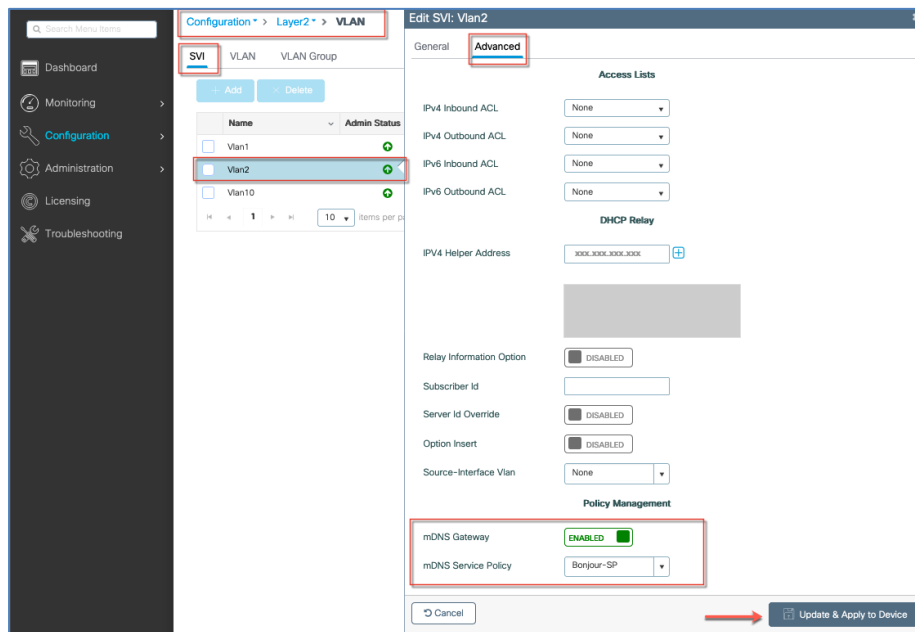
With release 17.1, “mdns-sd gateway” mode and “service-policy <policy-name>” CLI under this submode will be made visible on the VLAN SVI interface.

```

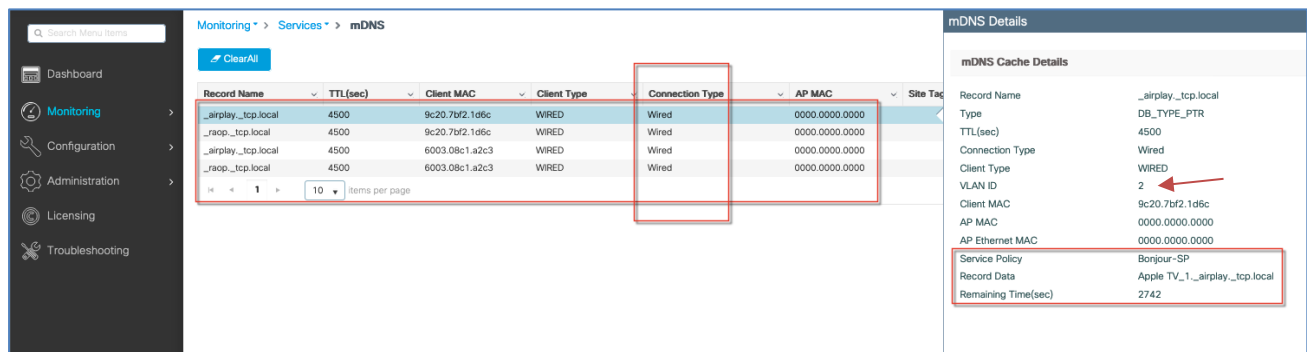
!
interface Vlan70
 ip address 10.70.0.211 255.255.255.0
 mdns-sd gateway
  service-policy tme-bonjour-SP1
!
  
```

If for example there is Apple TVs wire connected to the switch on VLAN2. In order for the WLC discover mDNS Services on the wired VLAN we have to configure SVI VLAN2 and enable mDNS GW and previously configured mDNS Policy.





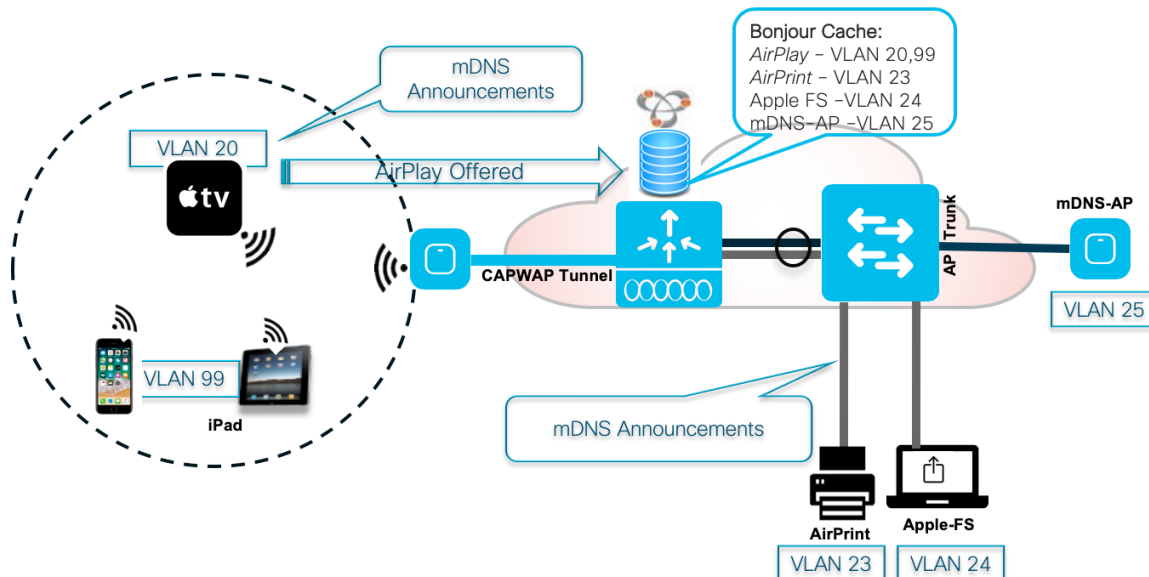
The Service from the Wired SP Apple TV on VLAN 2 will show as shown in the example below.



## mDNS-AP support in IOS-XE 17.1.1 and above

There are use cases where the VLAN in which mDNS Service provider is present may not be available on the C9800 controller or controller may not have visibility into that VLAN, therefore those services cannot be learnt on the mDNS Gateway on the controller. However, with an enhancement on the AP it can be made to listen to mDNS multicast packets and sent it to the wireless controller via a CAPWAP tunnel. Once these mDNS packets reach mDNS gateway, the controller will cache those services and share it when any device queries for them. This solution goes by the name mDNS-AP. It has two parts. AP side changes and wireless controller side changes.

The AP side support is already available as this feature was supported in AireOS WLC. Begin with IOS-XE release 17.1 the C9800 controllers are enhanced to support this feature. This feature is only supported on local mode and monitor mode AP(s).



In the above diagram, the File server and Printer are in wired network and AP is also on the same switch to which those devices are connected. The config on the switch will be changed so that AP can receive those mDNS packets. This is done by adding the AP trunk to allow the VLAN(s) on which the wired service provider exists. Once AP gets these mDNS packet since its configured as mDNS-AP it will send it to the Wireless Controller via CAPWAP and controller will learn those services from VLANs 23 and 24. mDNS-AP will be able to support up to 10 additional VLANs.

**Note:** Policy given by AAA override will have the highest priority followed by local / native profile followed by the policy configured under VLAN followed by Policy under wireless profile AAA Override > Local/Native > VLAN > Wireless profile.

## Enable/Disable mDNS-AP

On Enabling / Disabling mDNS-AP on a specific AP, Control-plane will send a AP-Update message to the Data-plane to indicate that this AP is capable of sending mDNS packets from a wired client on a VLAN that not visible to the controller.

```
ap name <AP name> mdns-ap {enable | disable} [vlan <vlan-id>]
```

### Adding Vlan to mDNS-AP already enabled

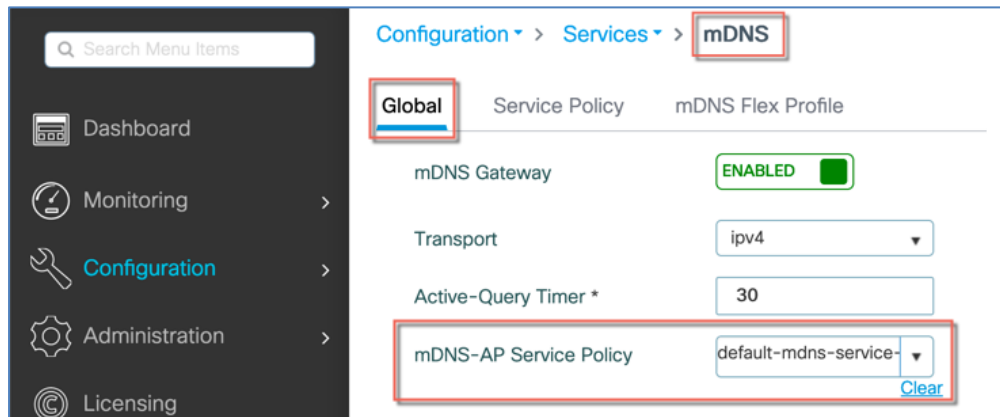
```
ap name <AP name> mdns-ap vlan {add | delete} <vlanid>
```

### Additional commands added are:

```
AP# show mdns-sd cache detail
AP#show mdns-sd cache wired
AP#show mdns-sd mdns-ap summary
```

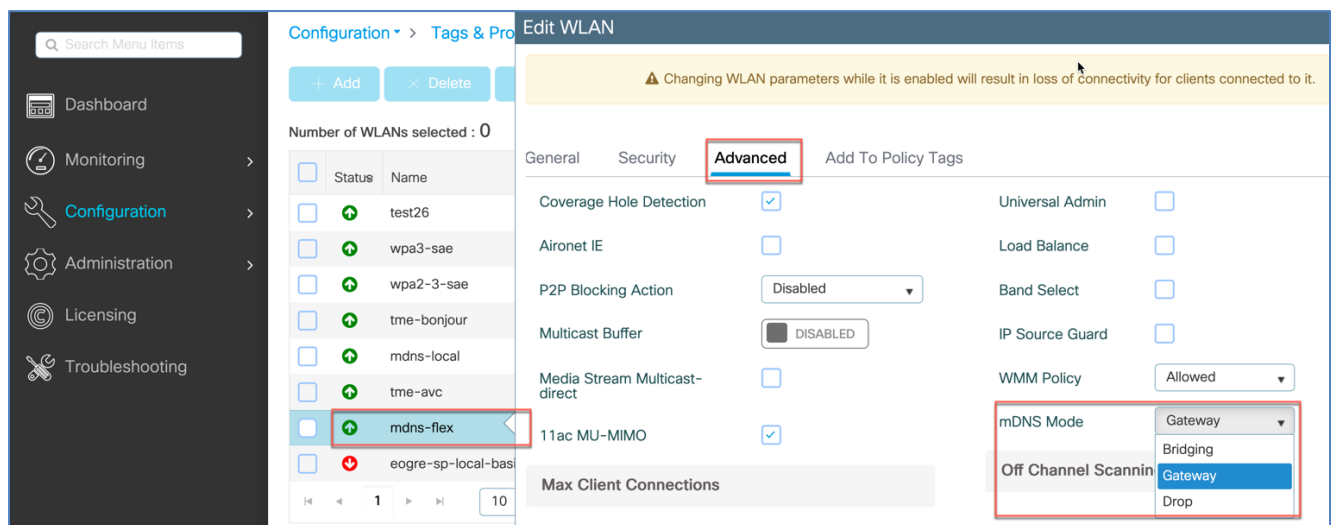
**Note:** mDNS-AP commands are available from CLI mode only.

To enable mDNS policy on the mDNS-AP, configure mDNS global setting and chose the mDNS-AP Service policy as shown in the example below.



## Flex Connect AP mDNS-Gateway Configuration Steps

For the FC AP mDNS-Gateway mode to function it has to be enabled Globally, the same as in the Local mode illustrated above. mDNS gateway has to be enable on the FC WLAN just like it was done for the Local mode in the steps above,



In the release 17.3 new Flex Mode mDNS configuration steps available to configure the mDNS Flex Profile under the mDNS configuration tab.

Configuration > Services > mDNS

Global Service Policy **mDNS Flex Profile**

+ Add - Delete

| Profile Name         | Service Ca |
|----------------------|------------|
| Bonjour-flex-profile | 1          |

1 10 items per page

**Edit mDNS Flex Profile**

Profile Name\* **Bonjour-flex-profile**

Service Cache Update Timer **1**

Statistics Update Timer **1**

VLAN(s) **23-24**

After the mDNS Flex profile is configured, configure the other parameters in that section and the Supported Local VLANs.

Note:

- Default value for update-timer for Service-cache & Statistics is 1 minute. Maximum 16 VLAN's can be configured in the wired VLAN range.
- Wireless client VLAN(s) and wired VLAN(s) must not be extended or visible on the WLC.
- All FC APs in a flex site must be on same VLAN.

Next step in configuring the Flex mDNS mode would be creating a Flex Profile under the Tags&Profiles Flex tab and mapping the earlier created "mDNS Flex Profile" to the Flex Profile as shown in the screen shot below the TME-Flex-Profile is tagged with the earlier created mDNS Flex Profile named "Bonjour Flex Profile"

Configuration > Tags & Profiles

+ Add - Delete

**Edit Flex Profile**

General Local Authentication Policy ACL VLAN Umbrella

Flex Profile Name

default-flex-profile

eogre-vw-flexprofile

**TME-Flex Profile**

1 1

Name\* **TME-Flex Profile**

Description **Flex Profile**

Native VLAN ID **1**

HTTP Proxy Port **0**

HTTP-Proxy IP Address **0.0.0.0**

CTS Policy

Inline Tagging ☐

SGACL Enforcement ☐

CTS Profile Name **default-sxp-profile**

Fallback Radio Shut ☐

Flex Resilient ☐

ARP Caching ☒

Efficient Image Upgrade ☒

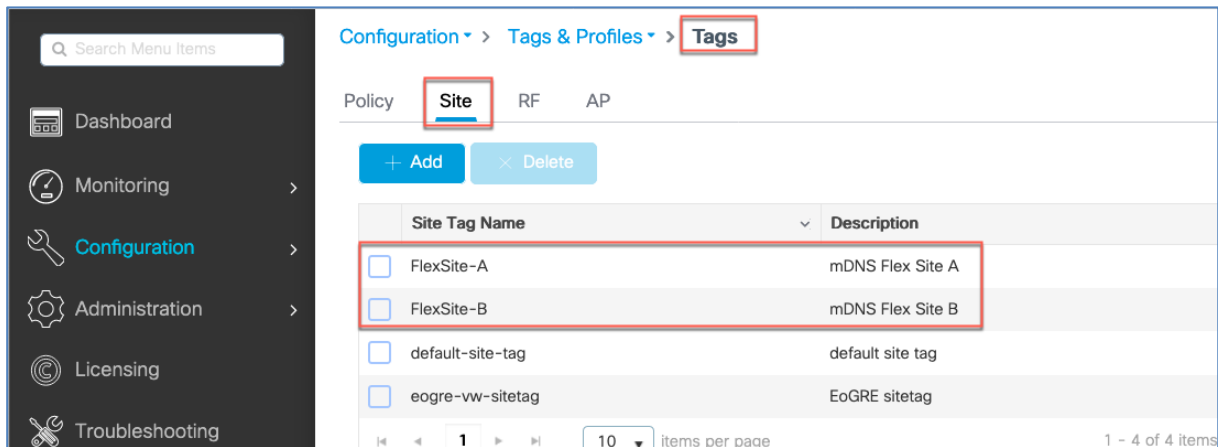
Office Extend AP ☐

Join Minimum Latency ☐

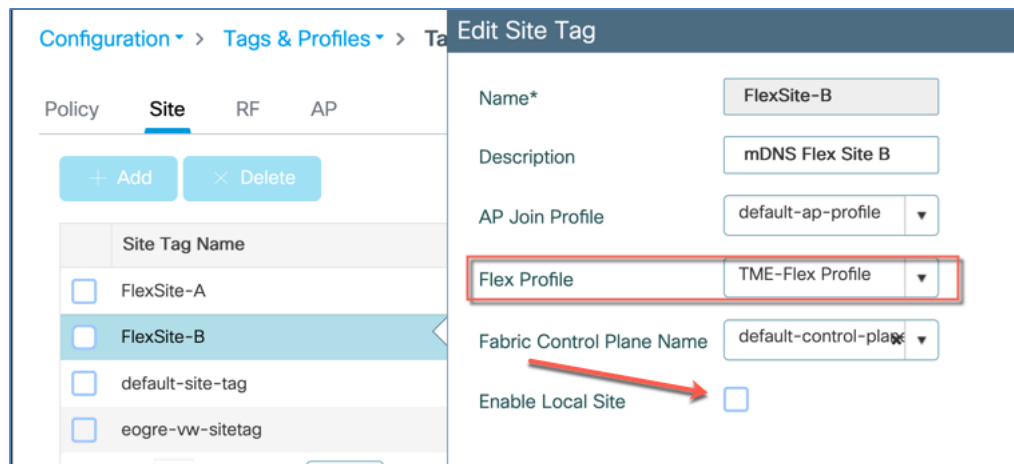
mDNS Flex Profile **Bonjour-Flex-Profile**

Next few steps will have to do with Flex Connect Mode configuration and applying proper policies and Tags.

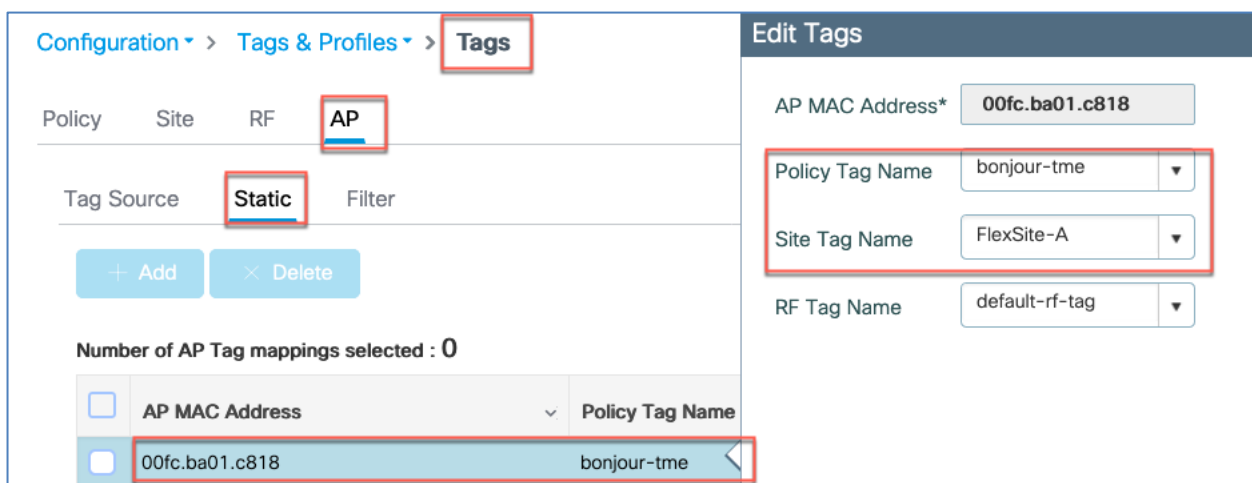
To configure Flex Sites go to the Configuration>Tags&Profiles>Tags and configure mDNS Flex Site A and B as shown in the screen shot below.



Make sure the earlier configured “TME-Flex Profile” is selected as a “Flex Profile” and the “Enable Local Site” box is unchecked as shown in the screen shot below.



In the next step under the Configuration>Tags&Profiles>Tags choose the “AP>Static” tab choose an AP and map to that AP(s) with Policy Tag Name or WLAN “bonjour-tme” and the Site Tag Name “FlexSite-A” created in the steps above and as shown in the example screen shot below.



By applying the above-mentioned Site Tag with “Enable Local Site” unchecked the AP(s) will be converted to the Flex mode and all Flex policies and profiles will be applied to them. Verify that all selected Tags and Policies are applied correctly by going to Configuration>Wireless>Access Points and selecting AP(s) to which the Tags were applied.

Configuration > Wireless > Access Points

▼ All Access Points

Number of AP(s): 1

| AP Name          | AP Model   | Slots | Admin Status | IP Address  | Base Radio MAC | AP Mode | Operation Status | Configuration Status | Policy Tag  | Site Tag   | RF Tag         | Tag Source | Location         | Country |
|------------------|------------|-------|--------------|-------------|----------------|---------|------------------|----------------------|-------------|------------|----------------|------------|------------------|---------|
| AP00FC.BA01.C818 | C9117AXI-B | 2     | ✓            | 10.70.0.131 | 00fc.ba01.d460 | Flex    | Registered       | Healthy              | bonjour-tme | FlexSite-A | default-rf-tag | Static     | default location | US      |

To see what mDNS services being advertised on the FlexSite-A configured in the previous steps go to Monitoring>Services>mDNS and all wireless or wired services being advertised on that Flex site.

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     | 7056.81db.cda0 | WIRED       | Wired           | 0000.0000.0000 | FlexSite-A (Flex) | 1                 | 1       | Office Apple TV_2 (2)._airplay._tcp.local       |
| _raop._tcp.local    | 4500     | 7056.81db.cda0 | WIRED       | Wired           | 0000.0000.0000 | FlexSite-A (Flex) | 1                 | 1       | 705681DBCDA0@Office Apple TV_2._raop._tcp.local |

## Global mDNS Monitoring on the C9800

When the GW is disabled (bridging mode) on the Global and WLAN levels – no services being cached on the controller and the Service list is empty even if the Clients with Service Advertisements are connected and available as illustrated below. In this mode Controller is not doing any mDNS snooping.

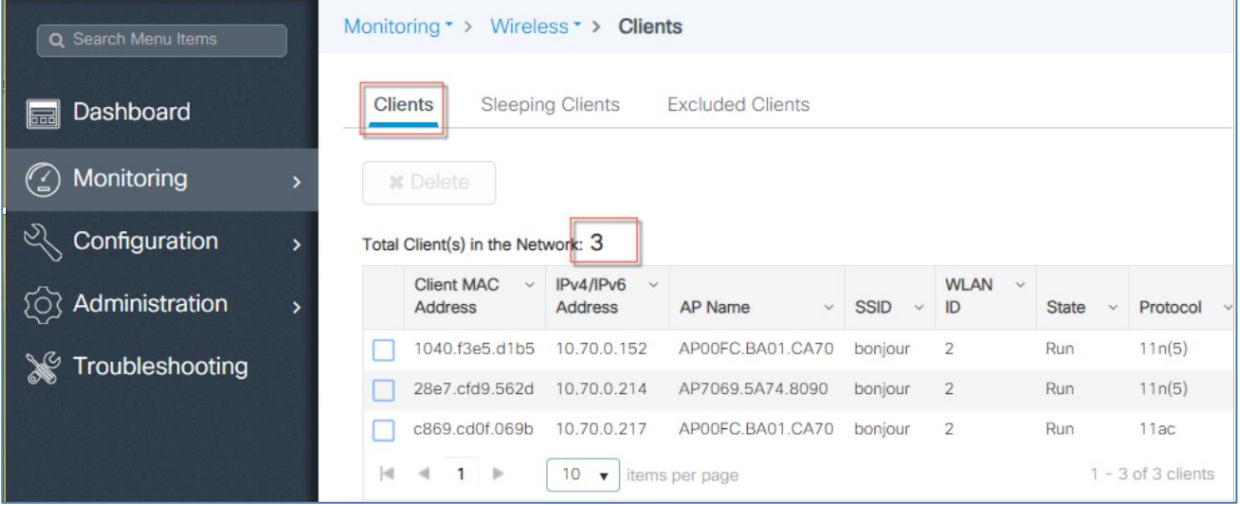
Monitoring > Services > mDNS

ClearAll

| Record Name         | TTL(sec) | Client MAC | Client Type | Connection Type | AP MAC | WLAN/GLAN/RLAN ID | VLAN ID | Record Data |
|---------------------|----------|------------|-------------|-----------------|--------|-------------------|---------|-------------|
| No items to display |          |            |             |                 |        |                   |         |             |

10 items per page





Monitoring > Wireless > Clients

**Clients** Sleeping Clients Excluded Clients

✖ Delete

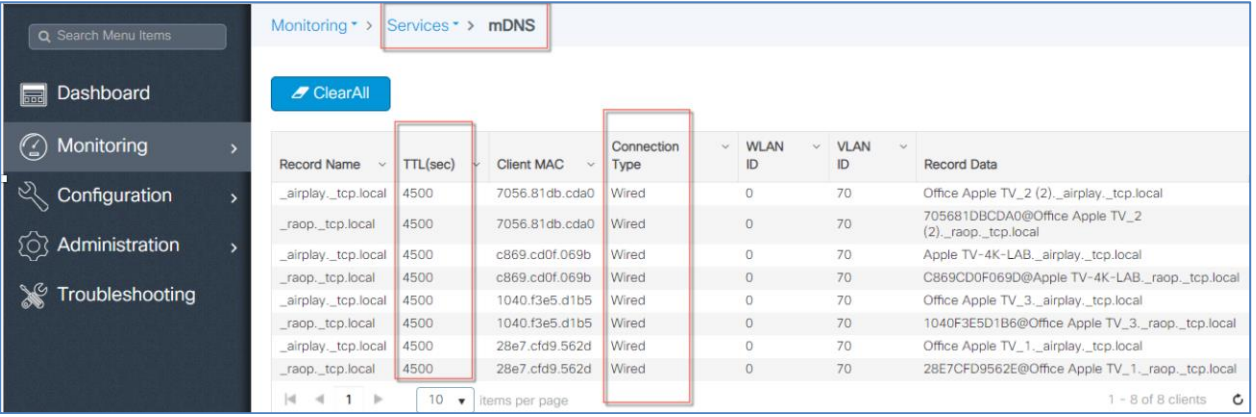
Total Client(s) in the Network: **3**

|                          | Client MAC Address | IPV4/IPV6 Address | AP Name          | SSID    | WLAN ID | State | Protocol |
|--------------------------|--------------------|-------------------|------------------|---------|---------|-------|----------|
| <input type="checkbox"/> | 1040.f3e5.d1b5     | 10.70.0.152       | AP00FC.BA01.CA70 | bonjour | 2       | Run   | 11n(5)   |
| <input type="checkbox"/> | 28e7.cfd9.562d     | 10.70.0.214       | AP7069.5A74.8090 | bonjour | 2       | Run   | 11n(5)   |
| <input type="checkbox"/> | c869.cd0f.069b     | 10.70.0.217       | AP00FC.BA01.CA70 | bonjour | 2       | Run   | 11ac     |

10 items per page 1 - 3 of 3 clients

When the GW is enabled on the Global and WLAN GW is disabled – then only wired services being cached on the controller since WLAN is in bridging mode.

Once Services are learned they need to be refreshed as per Active-Query Timer setting – default 30min (15-120). Default TTL will be 75 min and only refreshed when TTL timer has only ~15-20 min left of the service timer.



Monitoring > Services > mDNS

ClearAll

| Record Name        | TTL(sec) | Client MAC     | Connection Type | WLAN ID | VLAN ID | Record Data                                       |
|--------------------|----------|----------------|-----------------|---------|---------|---------------------------------------------------|
| _airplay_tcp.local | 4500     | 7056.81db.cda0 | Wired           | 0       | 70      | Office Apple TV_2 (2)_airplay_tcp.local           |
| _raop_tcp.local    | 4500     | 7056.81db.cda0 | Wired           | 0       | 70      | 705681DBCDA0@Office Apple TV_2 (2)_raop_tcp.local |
| _airplay_tcp.local | 4500     | c869.cd0f.069b | Wired           | 0       | 70      | Apple TV-4K-LAB_airplay_tcp.local                 |
| _raop_tcp.local    | 4500     | c869.cd0f.069b | Wired           | 0       | 70      | C869CD0F069D@Apple TV-4K-LAB_raop_tcp.local       |
| _airplay_tcp.local | 4500     | 1040.f3e5.d1b5 | Wired           | 0       | 70      | Office Apple TV_3_airplay_tcp.local               |
| _raop_tcp.local    | 4500     | 1040.f3e5.d1b5 | Wired           | 0       | 70      | 1040F3E5D1B6@Office Apple TV_3_raop_tcp.local     |
| _airplay_tcp.local | 4500     | 28e7.cfd9.562d | Wired           | 0       | 70      | Office Apple TV_1_airplay_tcp.local               |
| _raop_tcp.local    | 4500     | 28e7.cfd9.562d | Wired           | 0       | 70      | 28E7CFD9562E@Office Apple TV_1_raop_tcp.local     |

10 items per page 1 - 8 of 8 clients

As before only three clients are still connected.

Monitoring > Wireless > Clients

Clients Sleeping Clients Excluded Clients

✖ Delete

Total Client(s) in the Network: 3

|                          | Client MAC Address | IPv4/IPv6 Address | AP Name          | SSID    | WLAN ID | State | Protocol | User Name |
|--------------------------|--------------------|-------------------|------------------|---------|---------|-------|----------|-----------|
| <input type="checkbox"/> | 1040.f3e5.d1b5     | 10.70.0.152       | AP00FC.BA01.CA70 | bonjour | 2       | Run   | 11n(5)   |           |
| <input type="checkbox"/> | 28e7.cfd9.562d     | 10.70.0.214       | AP7069.5A74.8090 | bonjour | 2       | Run   | 11n(5)   |           |
| <input type="checkbox"/> | c869.cd0f.069b     | 10.70.0.217       | AP00FC.BA01.CA70 | bonjour | 2       | Run   | 11ac     |           |

10 items per page 1 - 3 of 3 clients

When the **GW is enabled on the Global and WLAN level**– then both wired and wireless services being cached on the controller. In setup as shown below there are 3 Wireless and 1Wired Apple TV connected.

## Legal Information

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR

INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies are considered un-Controlled copies and the original on-line version should be referred to for latest version.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

#### **Cisco Trademark**

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

#### **Cisco Copyright**

© 2020 Cisco Systems, Inc. All rights reserved.