mDNS Deployment Guide for Cisco Catalyst 9800 Series Wireless Controllers, Cisco IOS XE Amsterdam 17.1

First Published: March 12, 2020

Table of Contents

Objectives

This document provides information on the theory of operation and configuration for the Cisco Catalyst 9800 IOS-XE based solution 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 Deployment 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.

Platform Support

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

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

11ac Wave 1 and Wave 2 Access Points: AP1700, AP2700, AP3700, AP18xx, AP2802, AP3802, AP4800, 1540, 1560, 1570

Supported releases

IOS-XE -17.1 and higher

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:

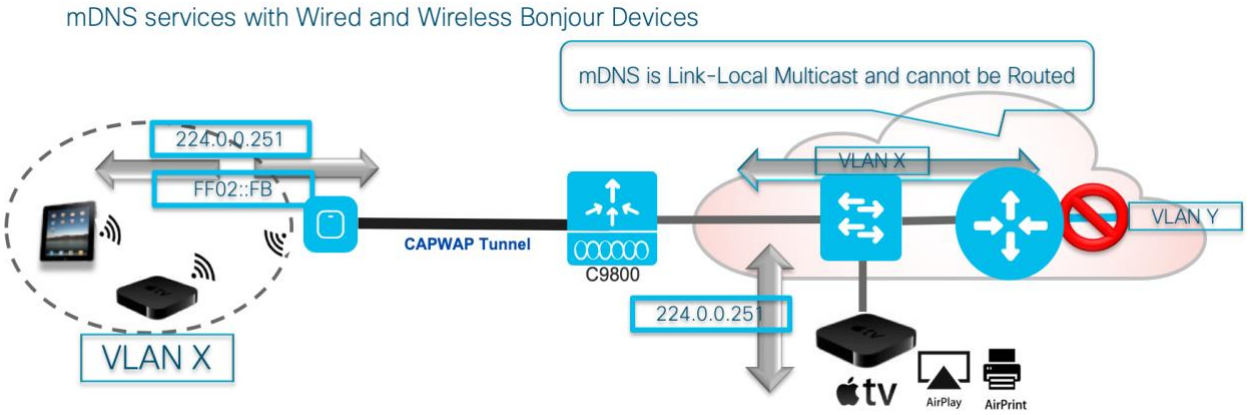
- Printing Services
- File Sharing Services
- Remote Desktop Services
- iTunes File Sharing
- iTunes Wireless iDevice Syncing
- AirPlay offering the following streaming services:

- 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.



C9800 mDNS Services in release IOS-XE 17.1

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.

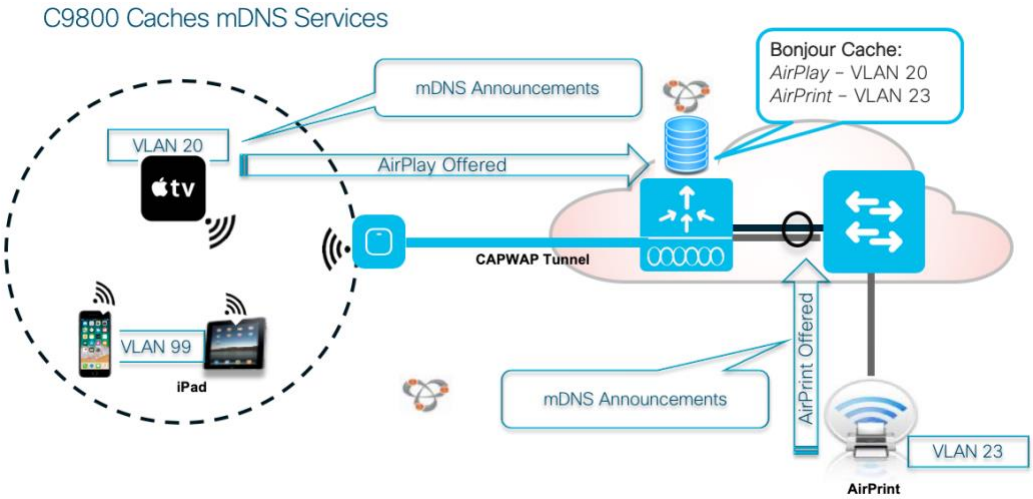
The following mDNS features included in IOS-XE rel 16.1:

- 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

Features added in IOS-XE release17.1

- 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



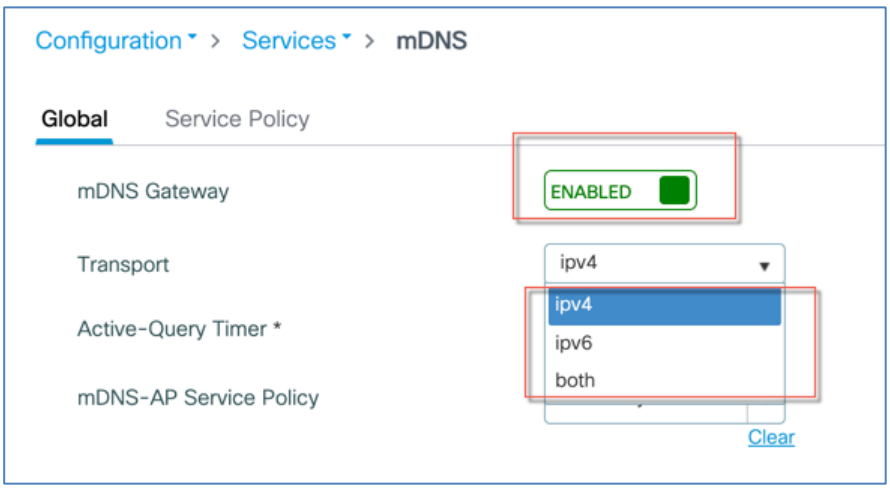
Discovering mDNS Services

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

On the C9800, mDNS Gateway configuration can be done at the global configuration level and/or at the WLAN level

mDNS on the Global Level

- mDNS gateway on C9800 is enabled/disabled at the global level
- Bridging is the default behavior at the Controller 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 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

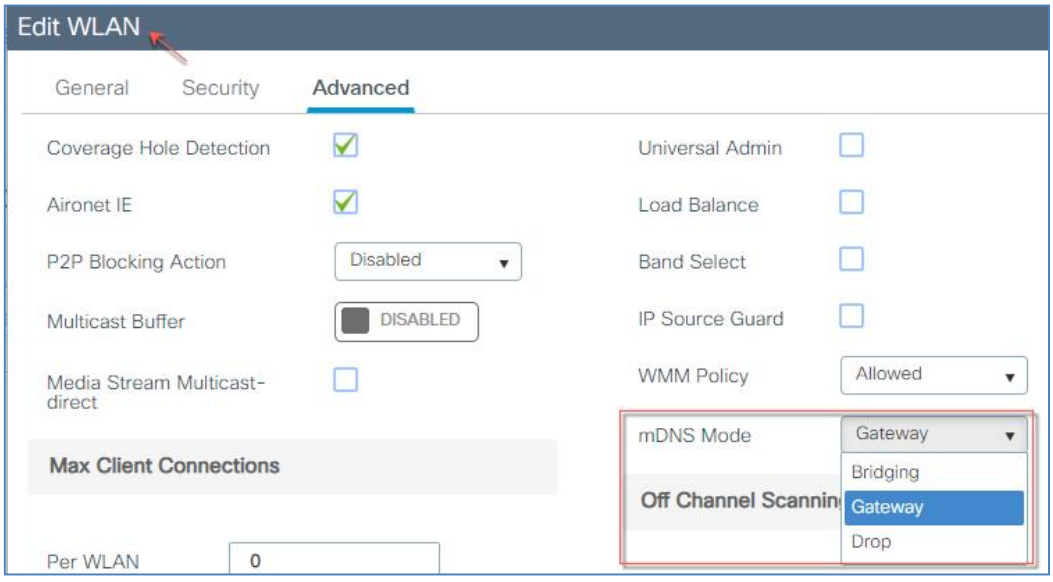


In Bridging mode (when mDNS GW is not enabled), packets with mDNS multicast IP and multicast mac will be sent on multicast CAPWAP tunnel. A multicast CAPWAP tunnel is a special CAPWAP tunnel used for reducing the number of copies of multicast packet that are required to be generated for each AP CAPWAP 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 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.

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

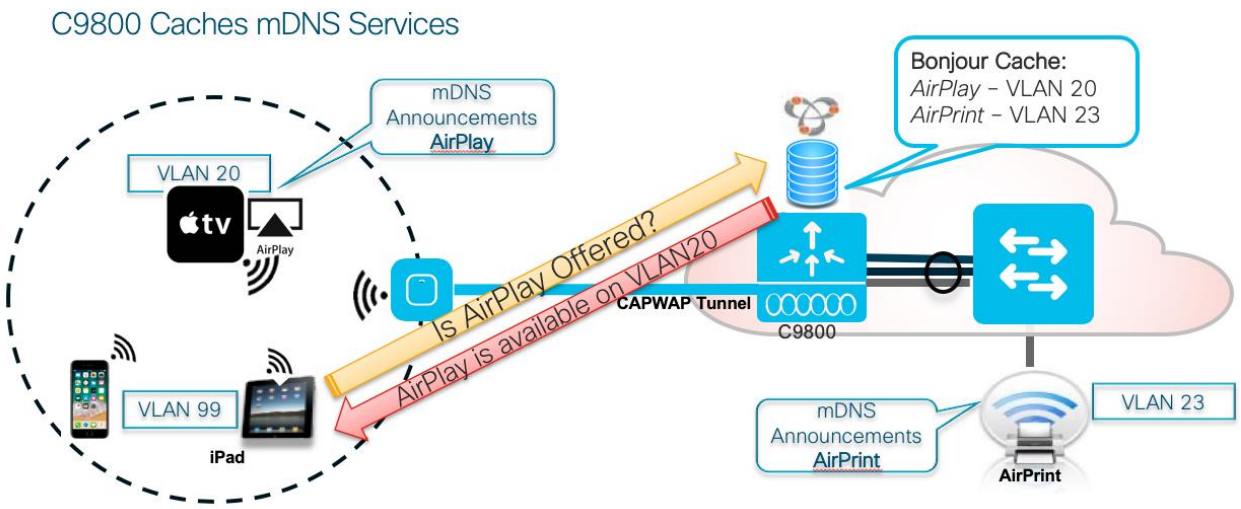


Caching mDNS Services

mDNS Gateway 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.

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



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

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.

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

The policy configured under wireless policy profile can be overridden by AAA configs that are downloaded as part of client authorisation. Only the mDNS service policy name will be part of the cisco av-pair (mDNS-ServicePolicy-name) that downloaded. User is require to configure a local service policy with that name.

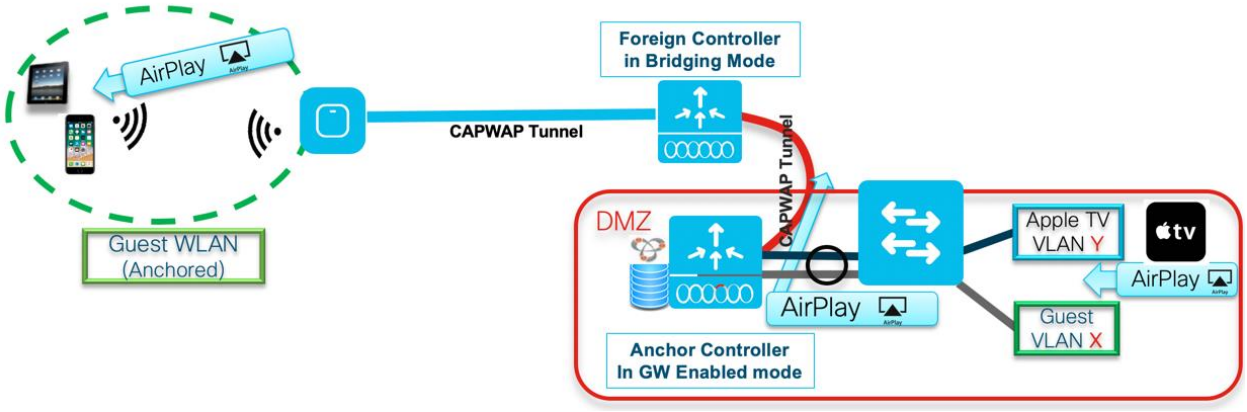
Note: Before IOS-XE release 17.1 location-based policy was be based on LSS model (RRM based) Site Tag, in the release 17.1 it is based on LSS, Site Tags and multiple Location Based filters documented below.

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

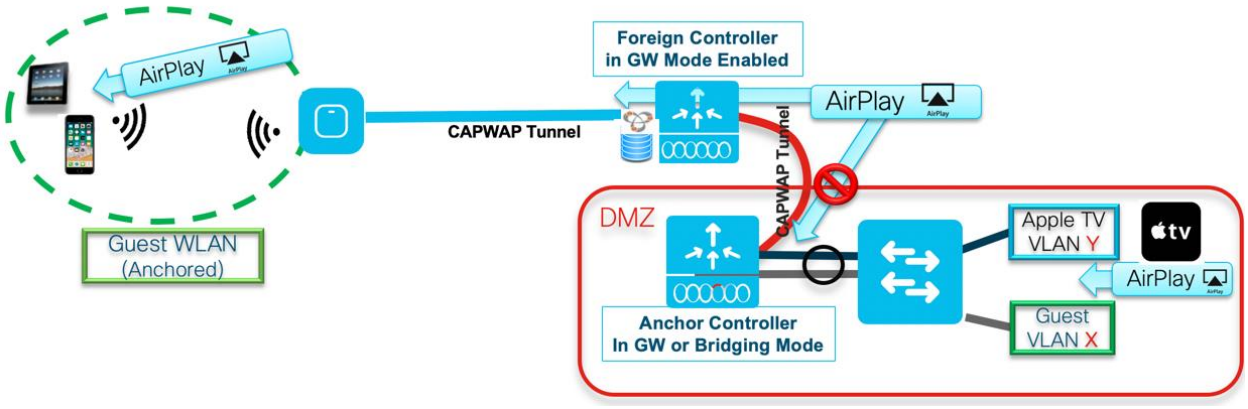
Begin with release 16.12.1, irrespective of the mDNS mode configured on the Anchor Controller, mDNS traffic coming from wireless clients gets

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).

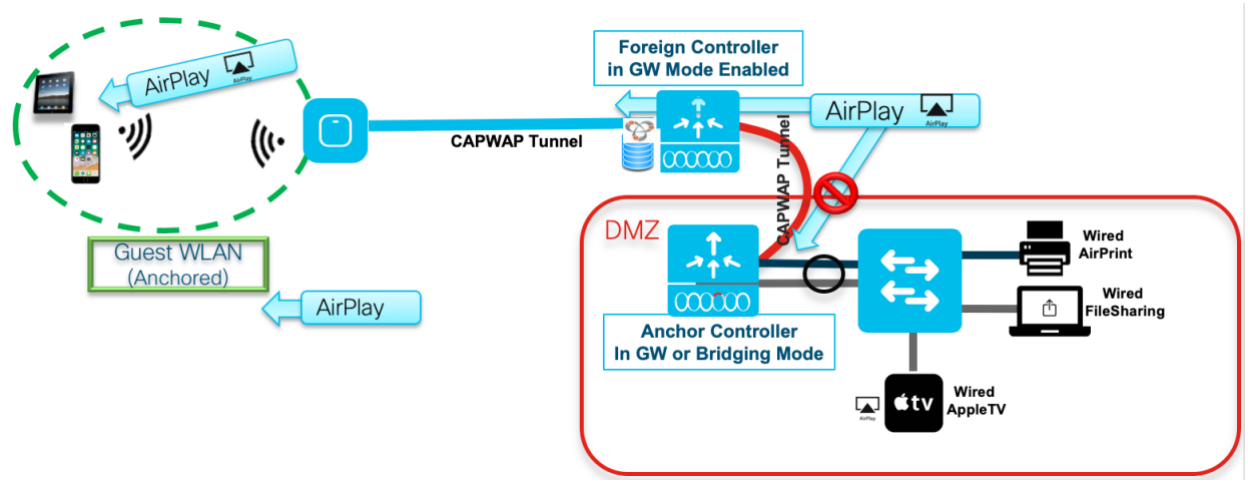
This behavior is in line with what is currently support in AireOS mDNS Guest solution

- 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.

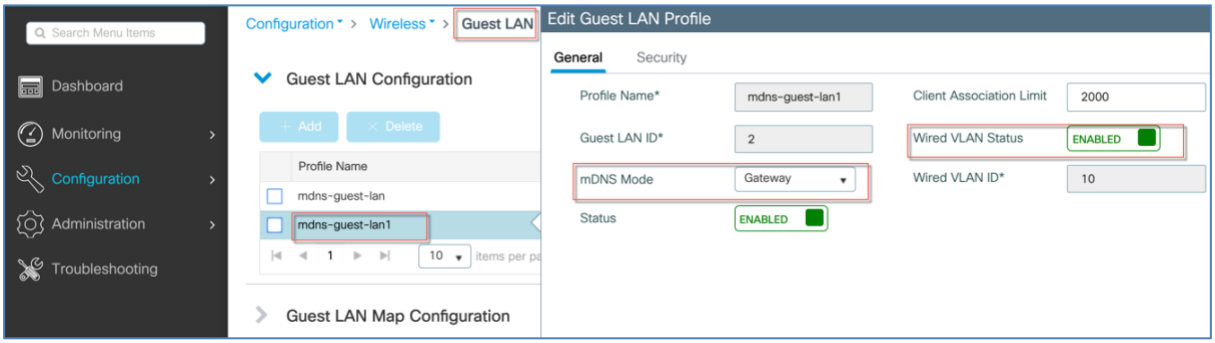


As part of this new enhancement in rel 17.1 new guest-lan profile specific CLI/WebUI is added to allow the administrator to configure mDNS mode on the guest-lan profile. This will be same CLI/WebUI configuration as what is present currently in WLAN profile mode for setting the mDNS mode. The default 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 this release 17.1.

Guest-LAN configuration example is shown below.



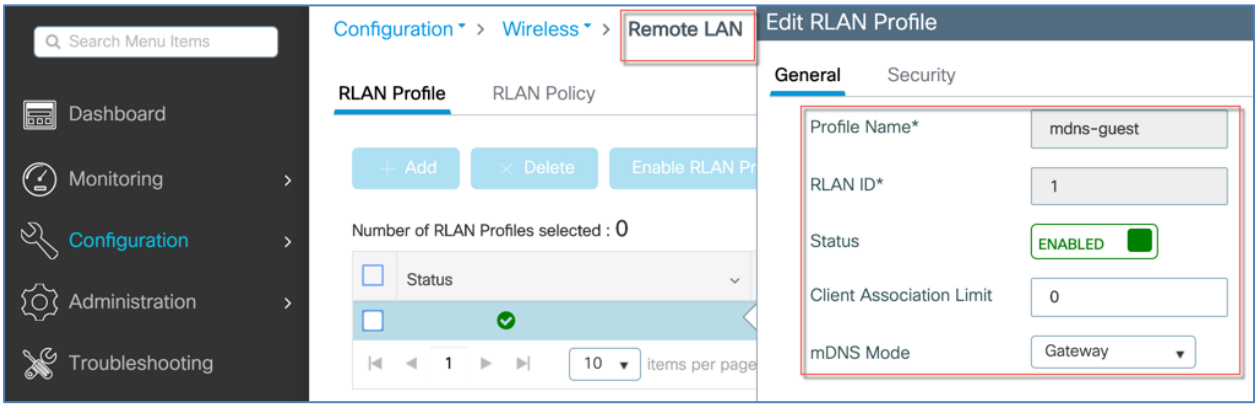
Same Guest-LAN configurations are available from the CLI mode, see configuration examples below.

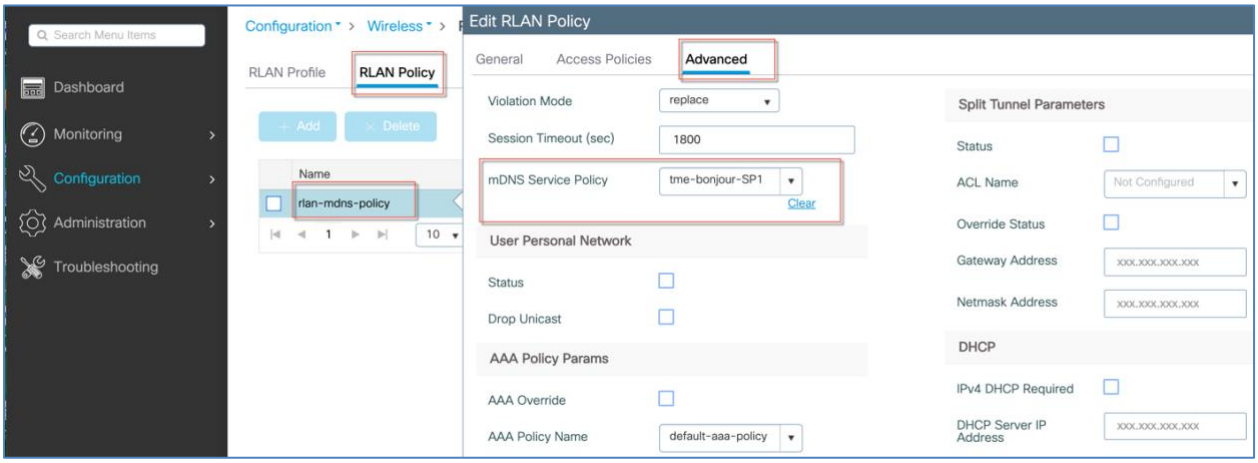
FOREIGN	ANCHOR

`# 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`
	`# 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`

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.





mDNS Configuration Steps

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.

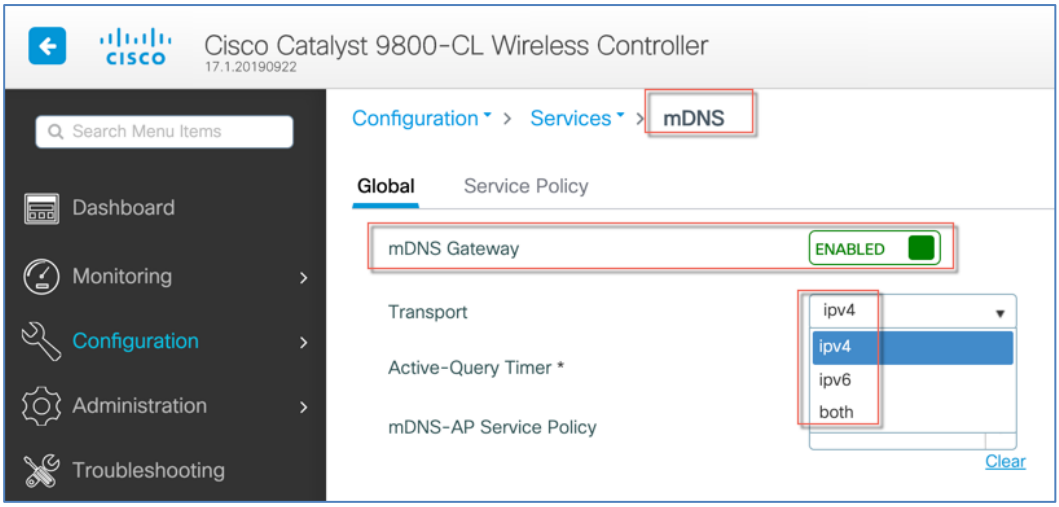
With deployment of mDNS gateway the mDNS Services won't flood subnet with mDNS announcements or queries

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.
- 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

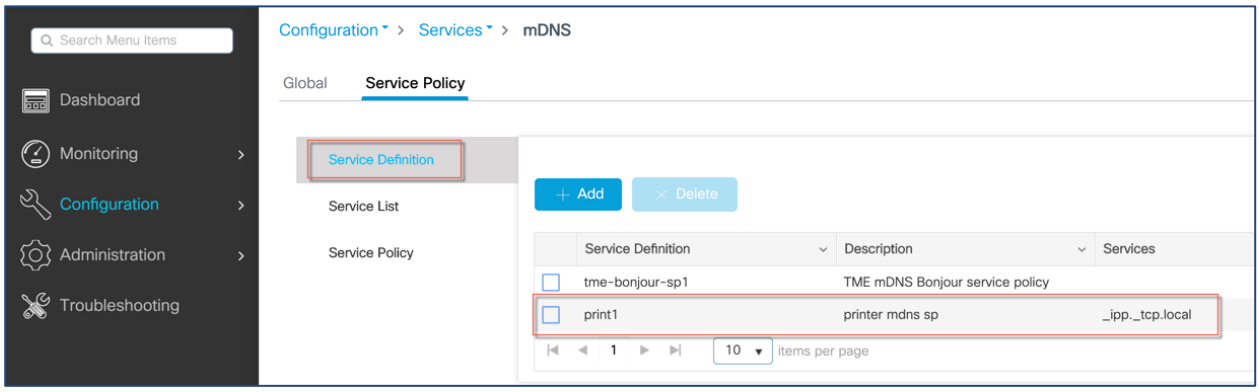


When mDNS gateway is enabled, mDNS packets will be consumed by the mDNS gateway and clients or device will be deprived of learning this service. In-order to share the service with the device and to provide ease of configuration to the administrator, few standard Service Types will be shared by **default** 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 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:



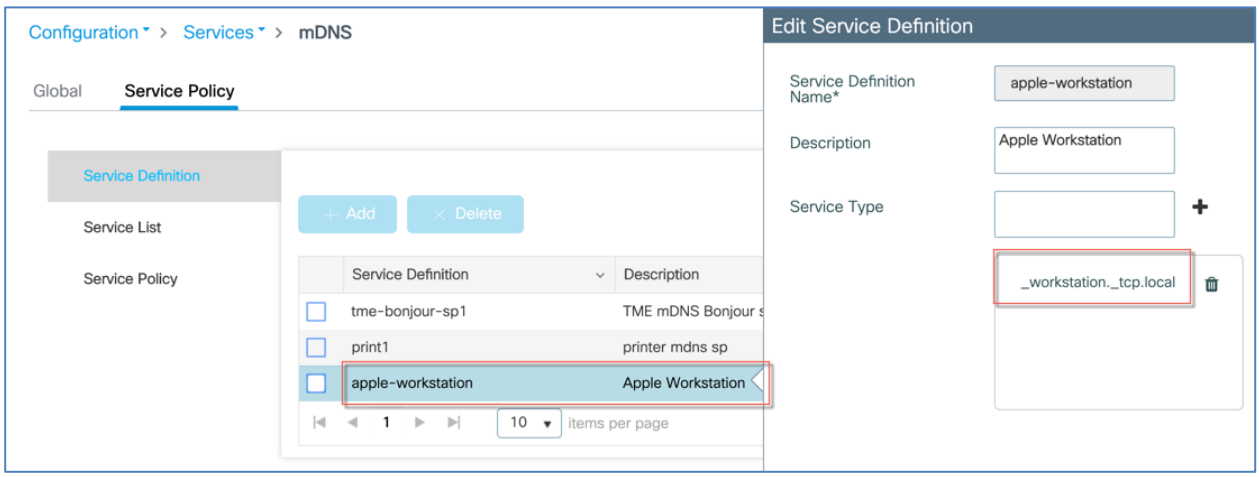
Service Definitions and the Services Lists are created on the Global mDNS Interface.

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.

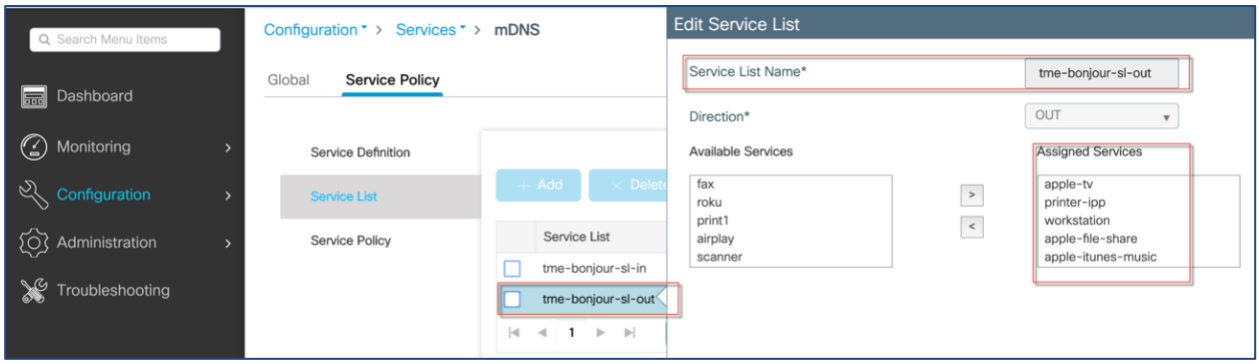
Service definition	Service Name
fax	_fax-ipp._tcp.local
roku	_rsp._tcp.local
airplay	_airplay._tcp.local
scanner	_scanner._tcp.local
spotify	_spotify-connect._tcp.local
airtunes	_raop._tcp.local
airserver	_airplay._tcp.local , _airserver._tcp.local
apple-rdp	_afpovertcp._tcp.local , _net-assistant._tcp.local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

apple-keynote	_keynotecontrol._tcp.local , _keynotepair._tcp.local
amazon-fire-tv	_amzn-wplay._tcp.local
apple-airprint	_ipp._tcp.local , _universal._sub._ipp._tcp.local
printer-socket	_pdl-datastream._tcp.local
apple-continuity	_companion-link._tcp.local
apple-file-share	_afpovertcp._tcp.local
apple-timecapsule	_adisk._tcp.local , _afpovertcp._tcp.local
google-chromecast	_googlecast._tcp.local
apple-itunes-music	_daap._tcp.local
apple-itunes-photo	_dpap._tcp.local
apple-remote-login	_sftp-ssh._tcp.local , _ssh._tcp.local
apple-screen-share	_rfb._tcp.local
apple-remote-events	_eppc._tcp.local
phillips-hue-lights	_hap._tcp.local
apple-itunes-library	_atc._tcp.local
multifunction-printer	_fax-ipp._tcp.local ,_ipp._tcp.local,_scanner._tcp.local
apple-timecapsule-mgmt	_airport._tcp.local
apple-windows-fileshare	_smb._tcp.local
itune-wireless-devicesharing2	_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 17.1and below Default Service List is not available in WebUI configuration display it can only be seen and configured from the CLI

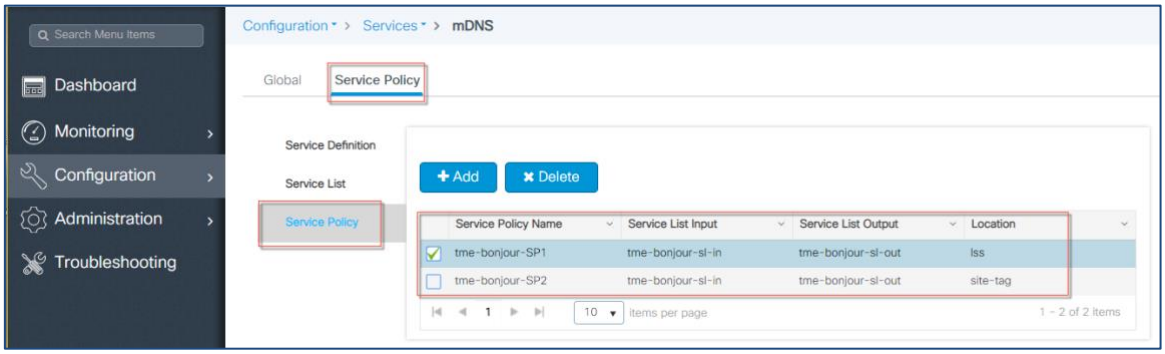
```
C9800-MA21#sh mdns-sd default-service-list
-----------------------------------------
        mDNS Default Service List
-----------------------------------------

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-ipps
Service Names: _ipps._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 it will use “default-mDNS-service-list” configuration for filtering mDNS service announcement and queries.

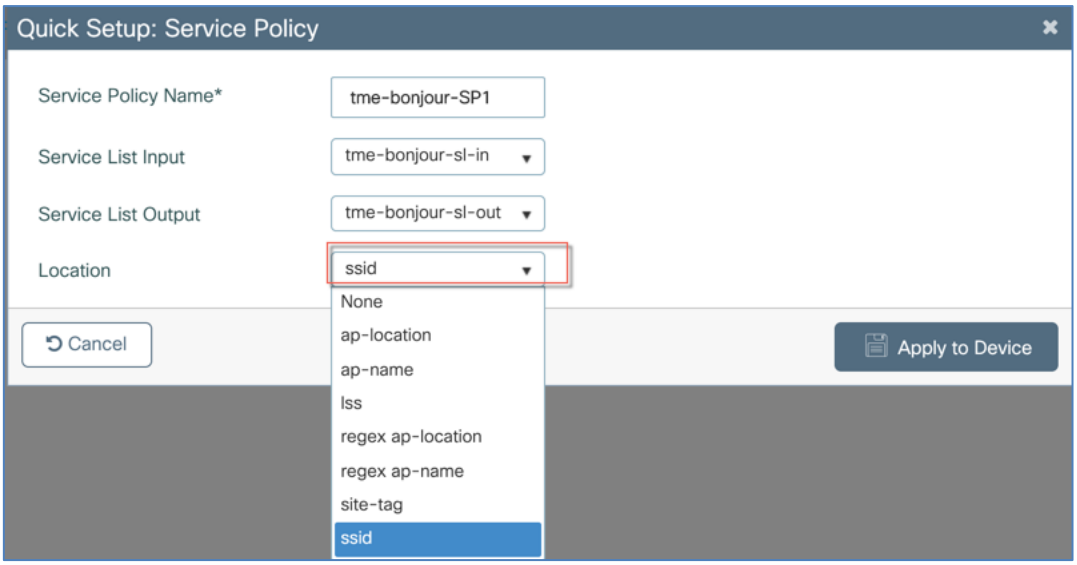


When Location Specific Services or LSS is selected: Only service instances from the querying client AP's neighbouring AP's will be given back to the querying client and service instances from the rest of the AP(s) will be filtered. Neighbouring AP list will be based upon Cisco RRM DB.

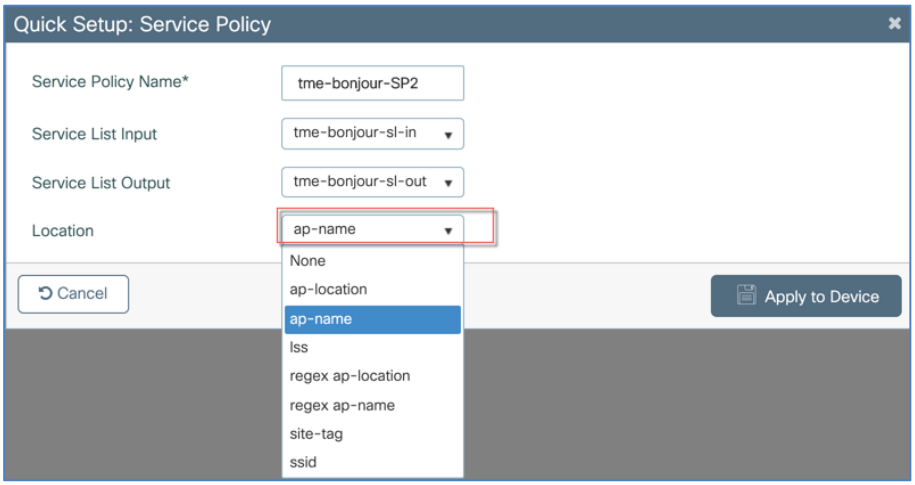
When Site Tag is selected: Site Tag is a construct introduced in C9800 IOS-XE and is a way to group AP(s). Service instances learned from Service Provider in that site-tag will be responded if query is received from clients associated to any AP in that site-tag.

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.



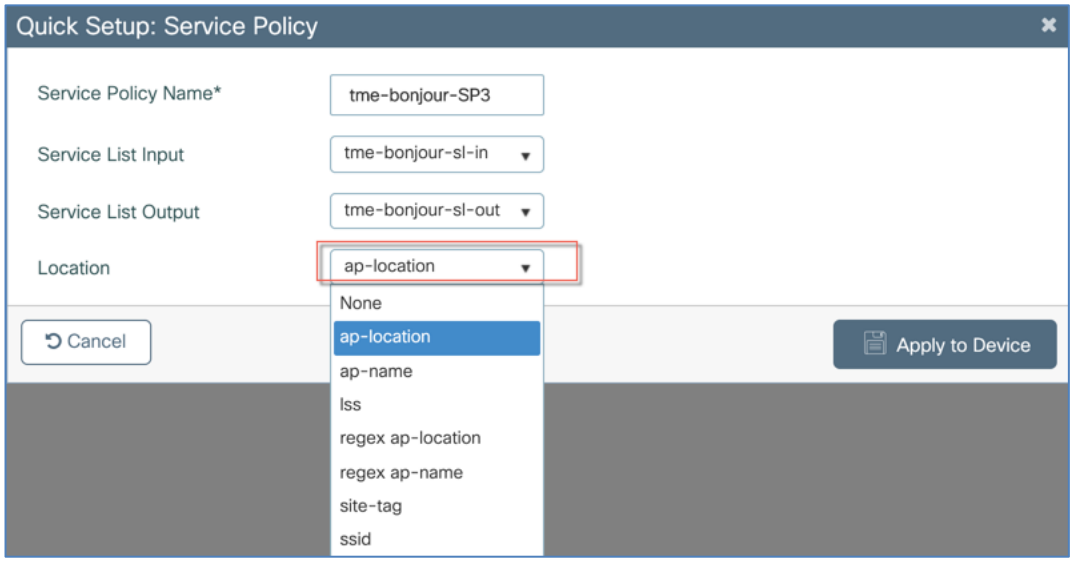
Based on AP Name selected: There are use cases where the requirement is to only respond with mDNS services which are learned from the same AP as that of the querying client. In release 17.1 mDNS location-based filtering is enhanced to filter services from the same AP-name if configured.



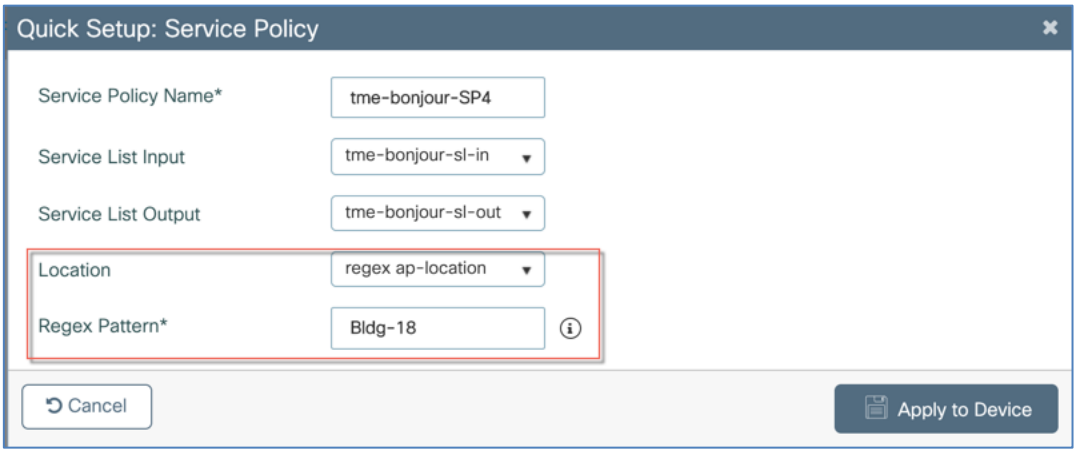
Based on AP Location selected: A group of APs usually share the same AP location. In other words, a particular AP Location string is usually shared by a group of APs. There are use cases, where the requirement is to only respond with mDNS service(s) which are learned from the

AP(s) which share the same AP Location string as that of the querying client AP Location string.

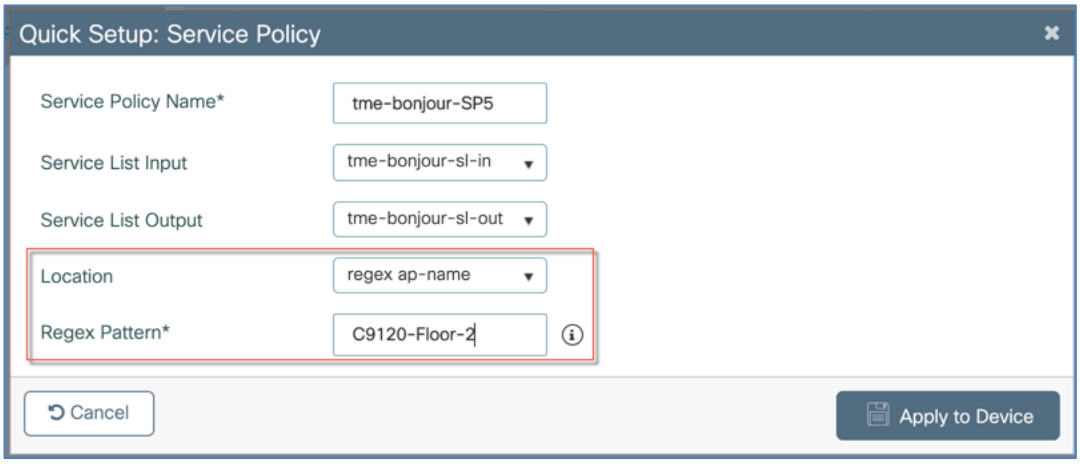
Basically, from the user perspective, only the services which share the same AP name as that of the querying client will be responded. In release 17.1 mDNS location-based filtering is enhanced to filter services from the AP Location if configured.



Based on Regular Expression matching on AP location: 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 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 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.

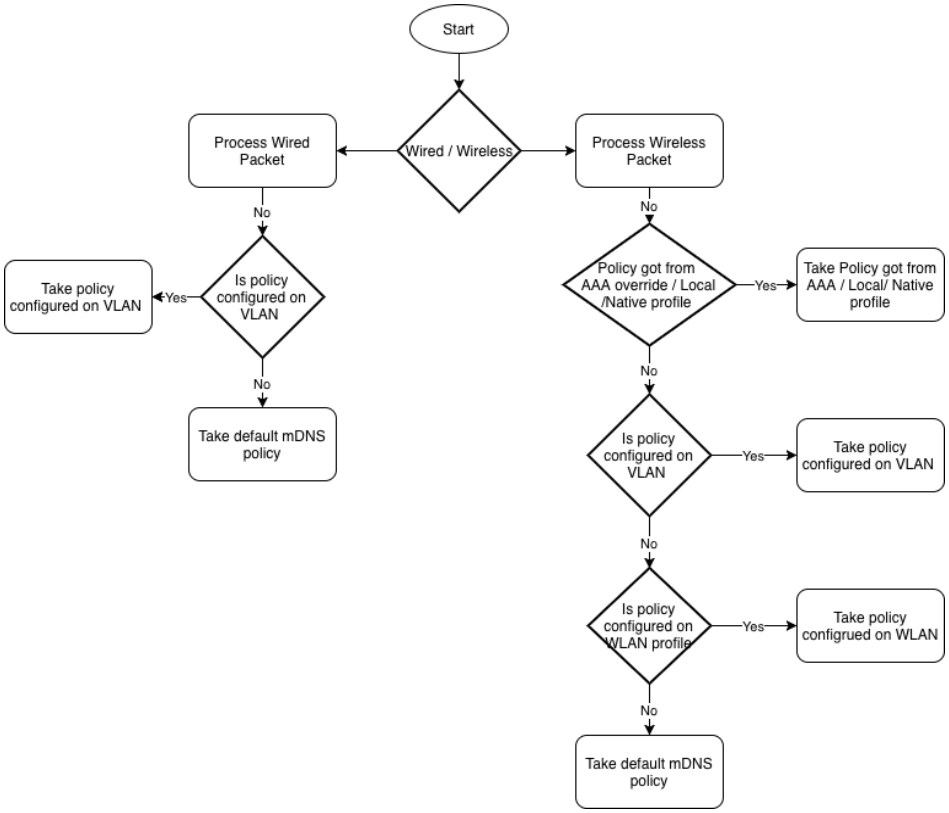
```
!
show mdns-sd service statistics
PTR-Name          Count
_ipp._tcp.local     10
_airplay._tcp.local     18
!
```

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.

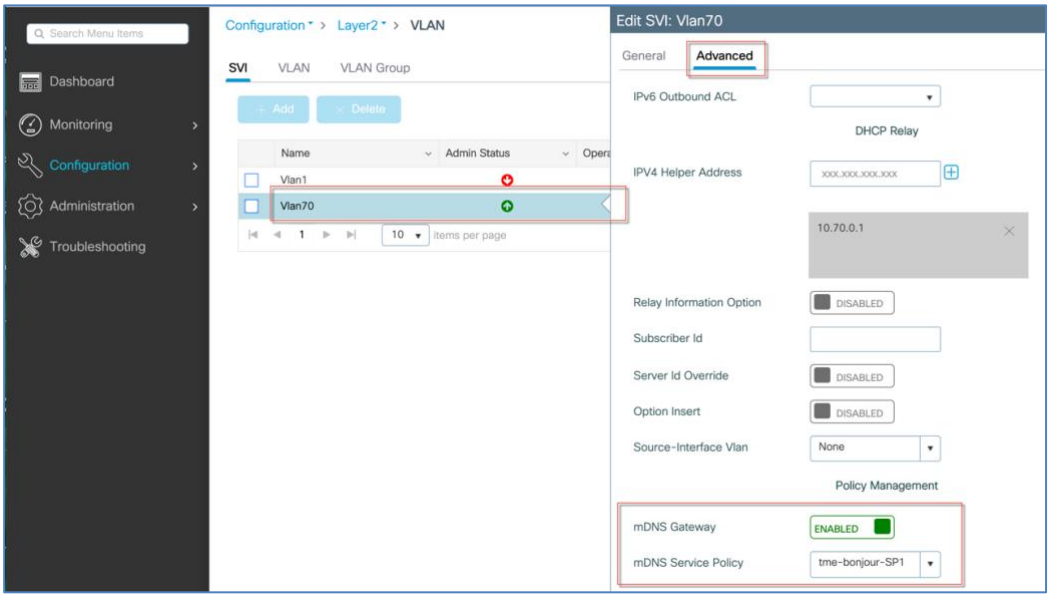
```
!
interface Vlan70
 ip address 10.70.0.211 255.255.255.0
 mdns-sd gateway
  service-policy tme-bonjour-SP1
!
```

Configuring mDNS Service Policy on VLAN SVI interface

To configure Wired Services advertised on the VLAN SVI follow the steps as shown below Configuration > Layer2 > SVI VLAN _ Advanced.

Enable the mDNS Gateway on the VLAN and choose the Service Policy as it was configured under mDNS setup earlier.

Note: In rel 17.1 Service list has to be the same under Wired and Wireless configurations.

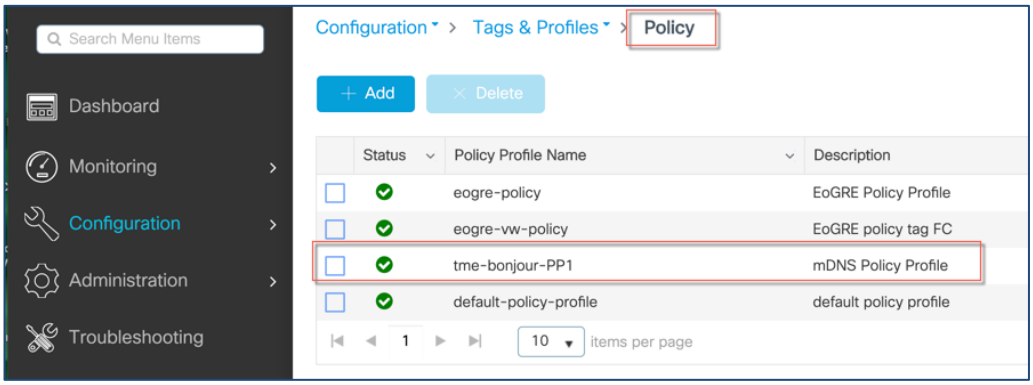


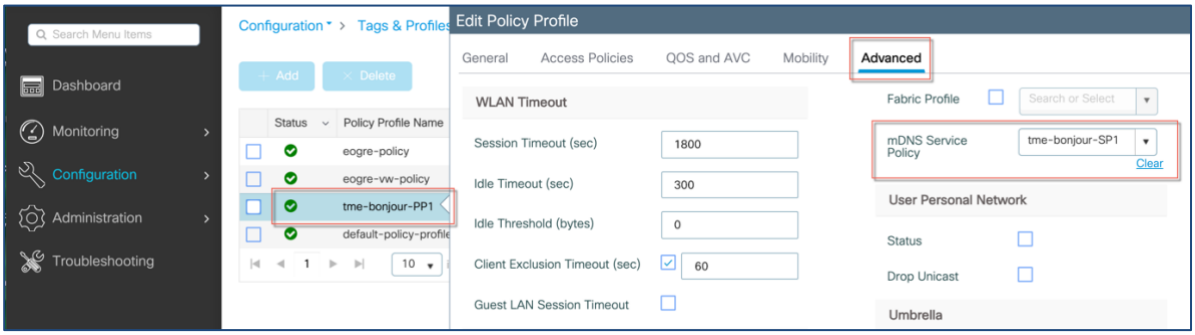
Note: Policy given by AAA override will have the highest priority followed by local / native profile followed by the policy configured under VLAN followed by Policy under wireless profile AAA Override > Local/Native > VLAN > Wireless profile.

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 “wireless profile policy” mode will associate “mdns-sd service-policy” with wireless profile policy.





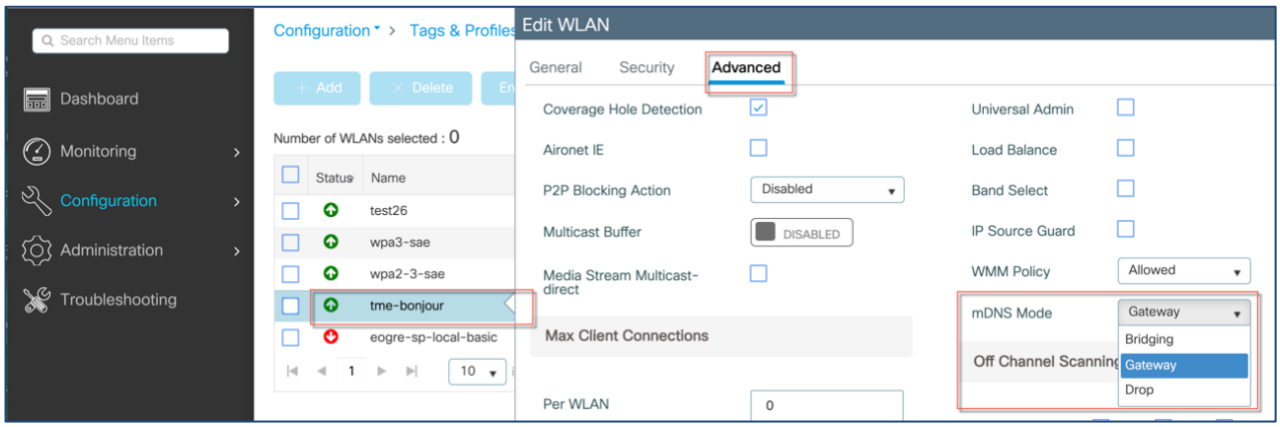
WLAN mDNS Mode is configured under WLAN tab. Different WLANs can have different Policies and Profiles, which can be also AAA overridden for more granularity.

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:

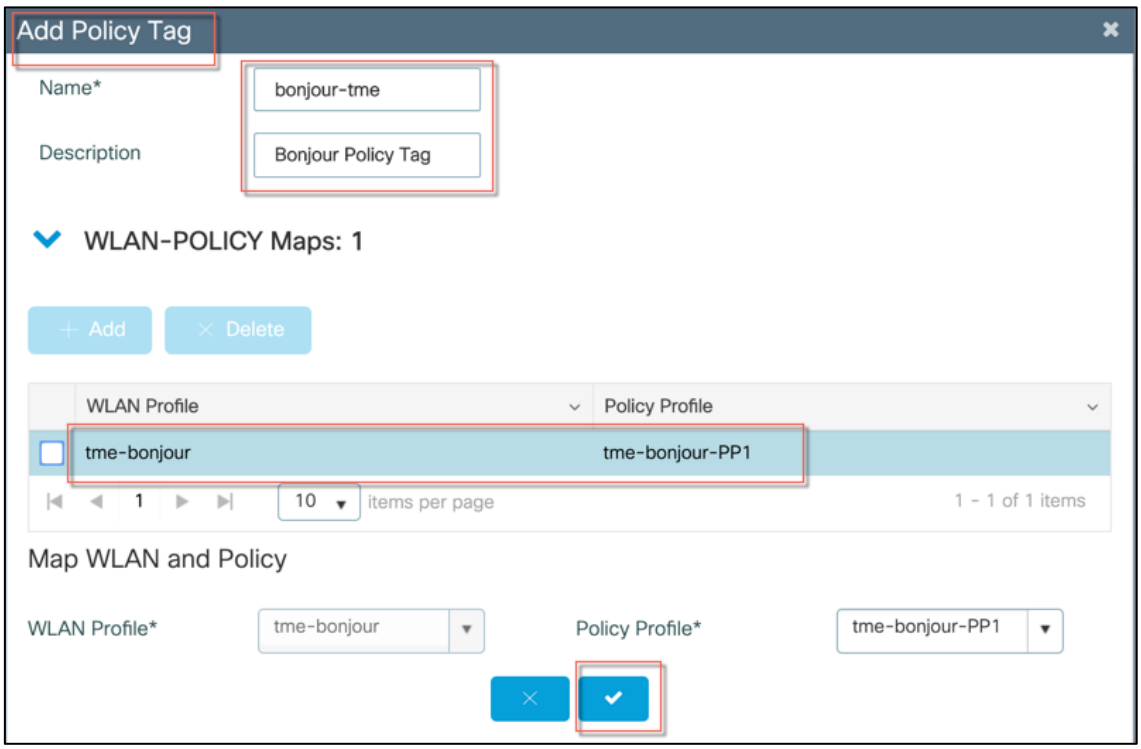
mdns-sd drop	**Disables mDNS function on that WLAN meaning , that all mDNS packets, announcements and queries, will be dropped on that WLAN**
mdns-sd gateway	This would be effective only when gateway is enabled at the global level otherwise WLAN will be effectively in drop mode
no mdns-sd gateway	No 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:

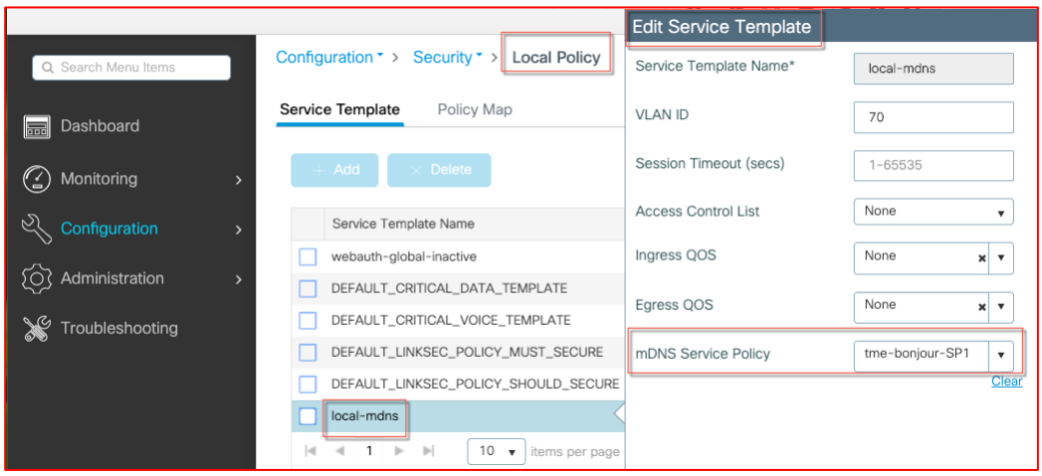


The following command under Configuration Tag>Policy , will Map the Policy Profile to the WLAN with mDNS GW mode enabled in the previous step



Local or Native mDNS Policy Profile

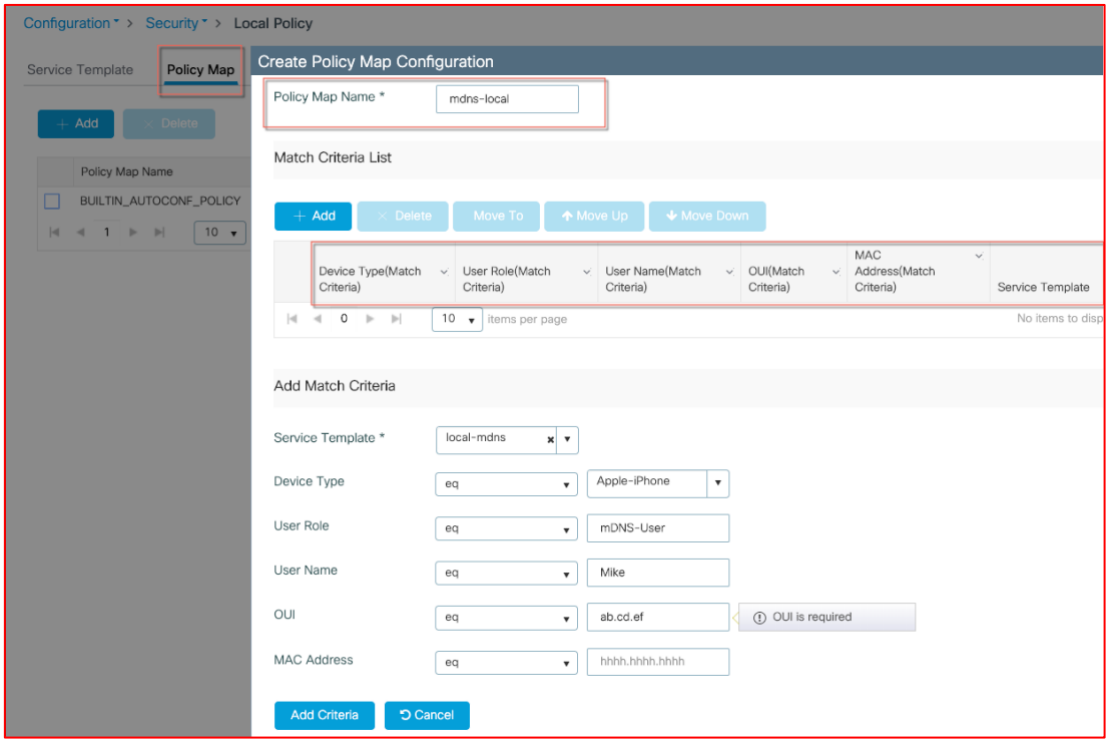
When Administrator configures local Authentication and Authorization and does not expect to get any mDNS policy from AAA server, Administrator can configure a Local or Native profile to select a mDNS policy based on user / role / device type. Currently parameters like VLAN, VNID, OpenDNS etc. are part 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.

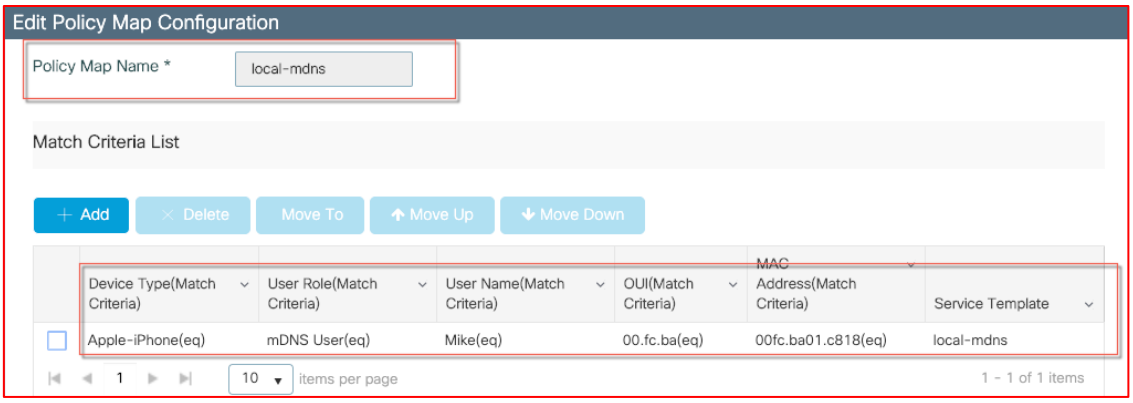


Under Policy Map the administrator can configure Service template with parameters such as

- Device Type
- User Role

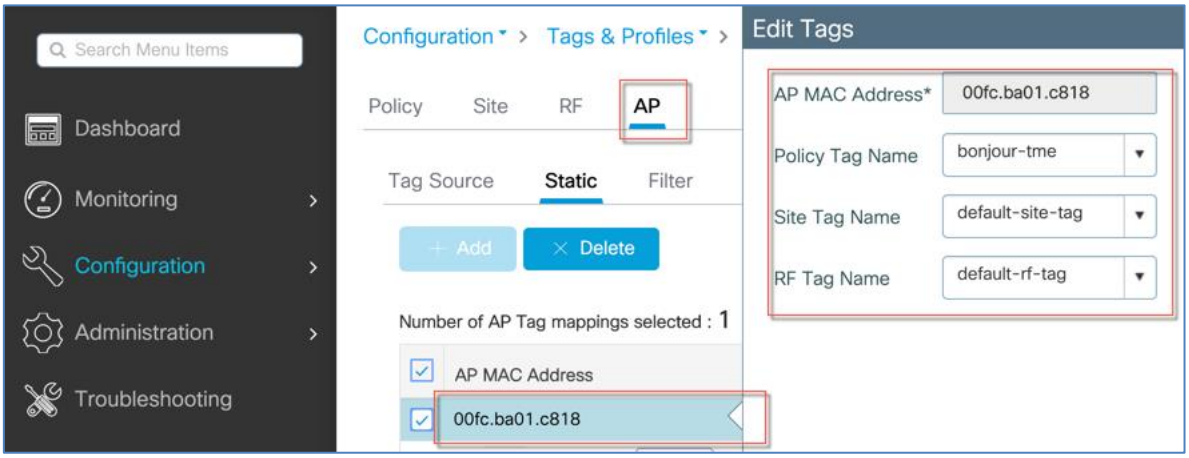
- Username
- OUI
- MAC address
- Other custom criteria



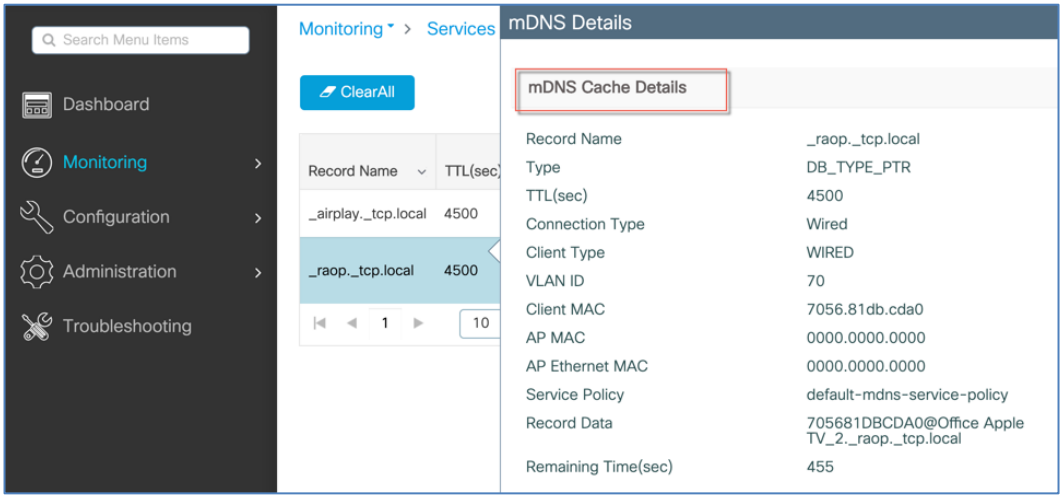


Configuring Site and Policy Tags to AP

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

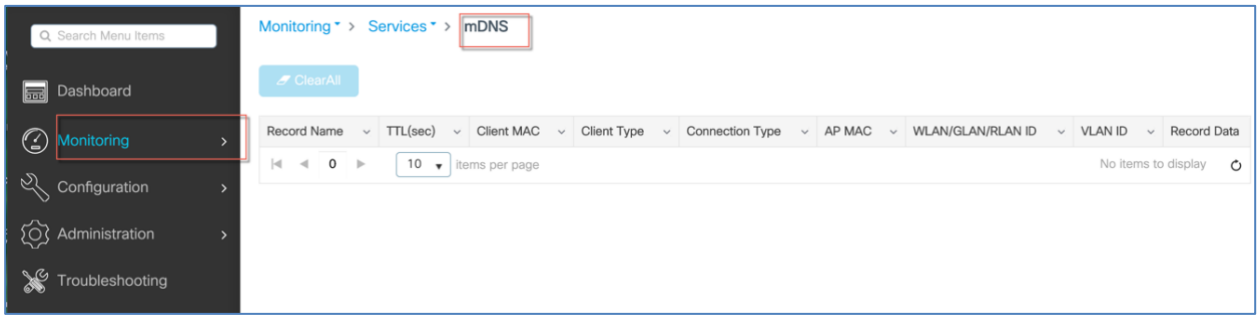


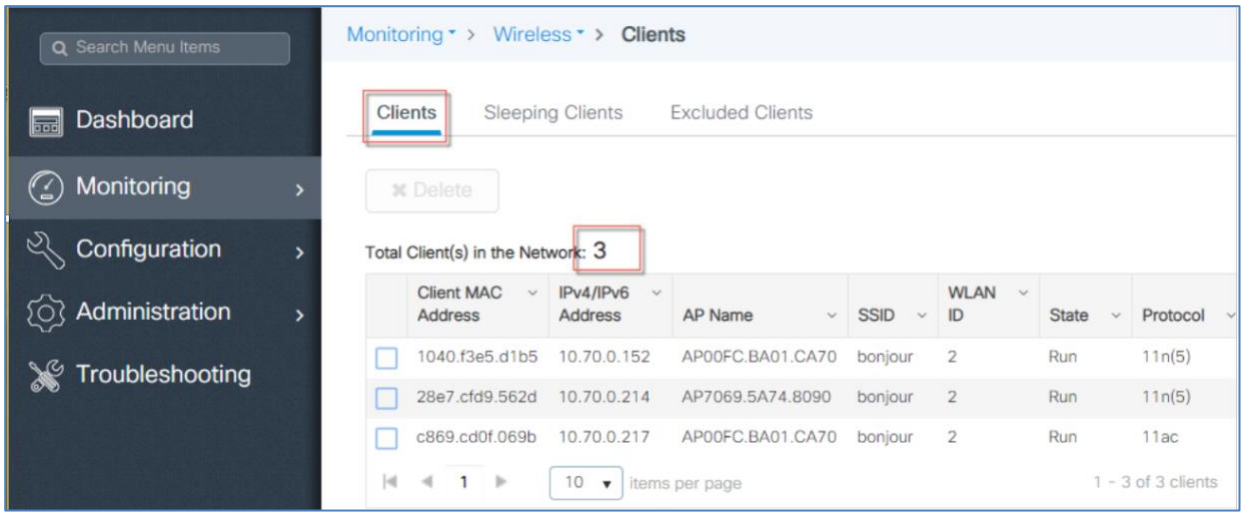
After the Site-Tag and Policy-Tags are applied to the specific AP, in the mDNS Monitoring will show under the Cached Service Type Detail



mDNS Monitoring on the C9800

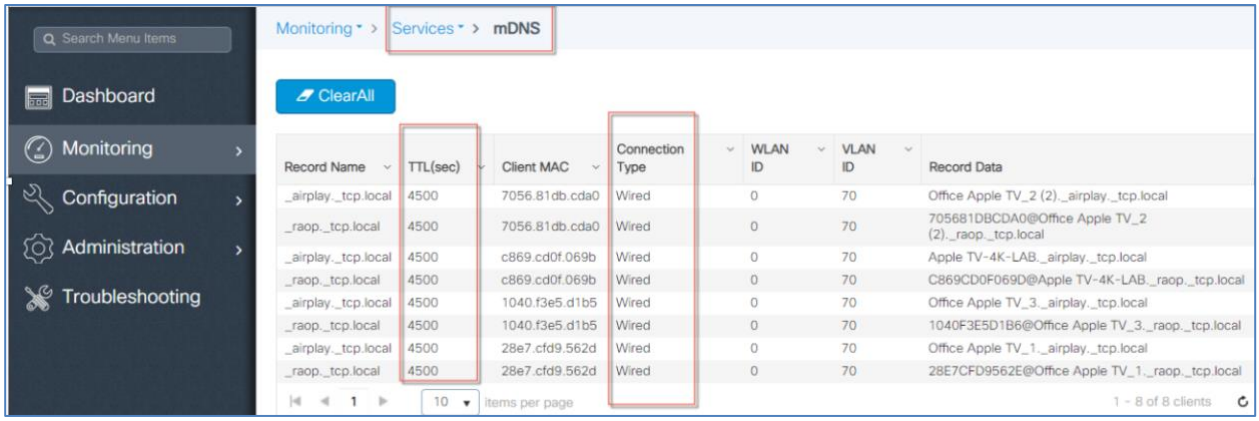
When the GW is disabled (bridging mode) on the Global and WLAN levels – no services being cached on the controller and the Service list is empty even if the Clients with Service Advertisements are connected and available as illustrated below. In this mode Controller is not doing any mDNS snooping.



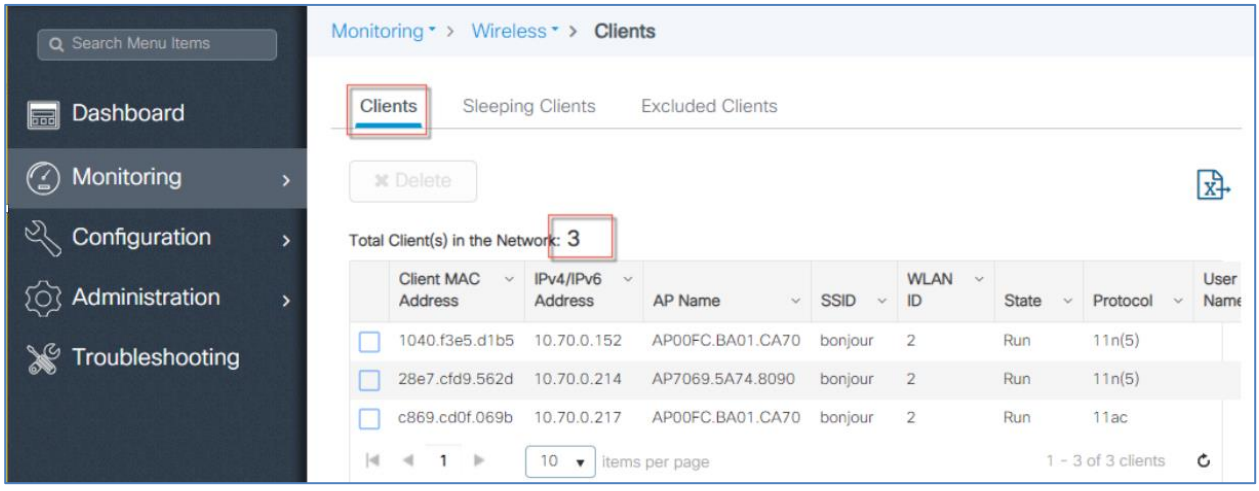


When the GW is enabled on the Global and WLAN GW is disabled – then only wired services being cached on the controller since WLAN is in bridging mode.

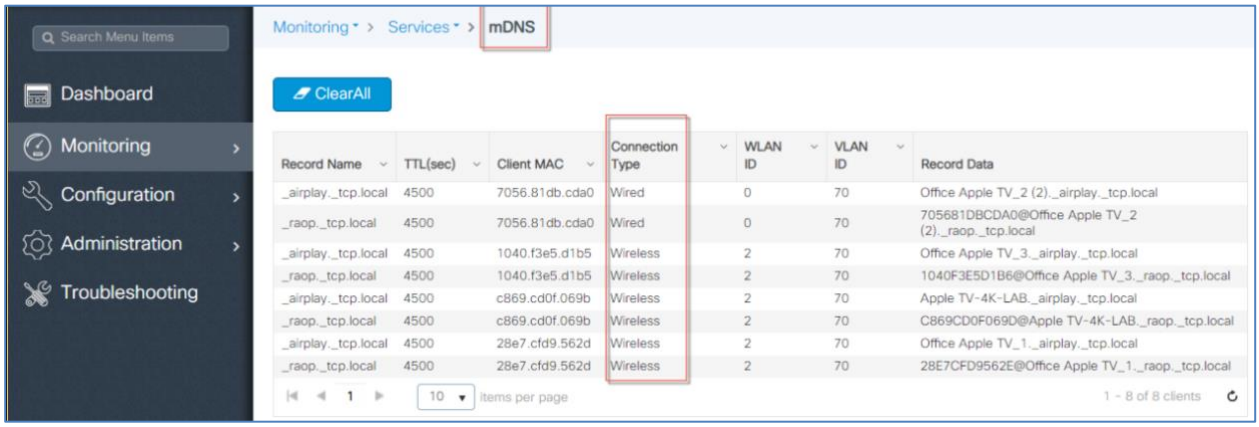
Once Services are learned they need to be refreshed as per Active-Query Timer setting – default 30min (15-120). Default TTL will be 75 min and only refreshed when TTL timer has only ~15-20 min left of the service timer.



As before only three clients are still connected.



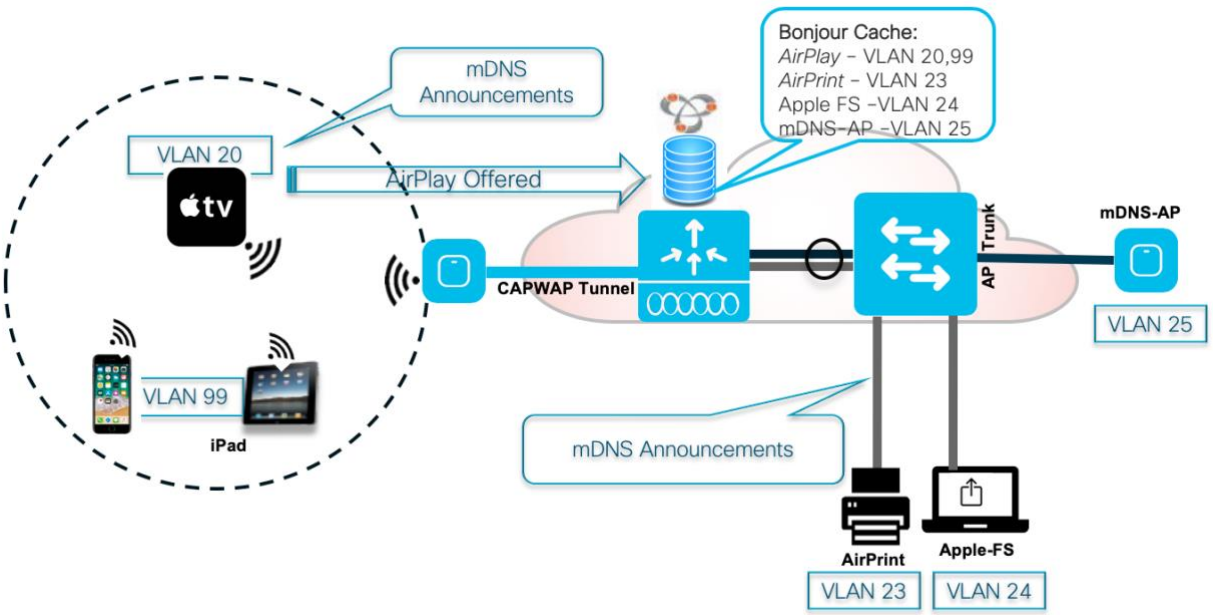
When the **GW is enabled on the Global and WLAN level**– then both wired and wireless services being cached on the controller. In setup as shown below there are 3 Wireless and 1Wired Apple TV connected.



mDNS-AP support in IOS-XE 17.1

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.

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:

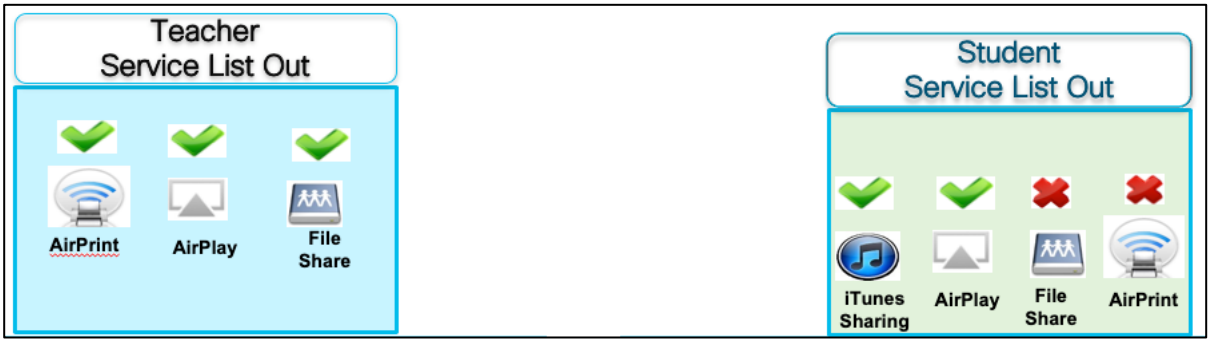
```
AP# show mdns-sd cache detail
AP#show mdns-sd cache wired
AP#show mdns-sd mdns-ap summary
```

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

mDNS Policy Example for Education with AAA Override

In this configuration example we will show how to configure mDNS Service Policy for the Teacher Service Type List-out and Students Service type List-out. Both of these configurations will be applied to the same WLAN.

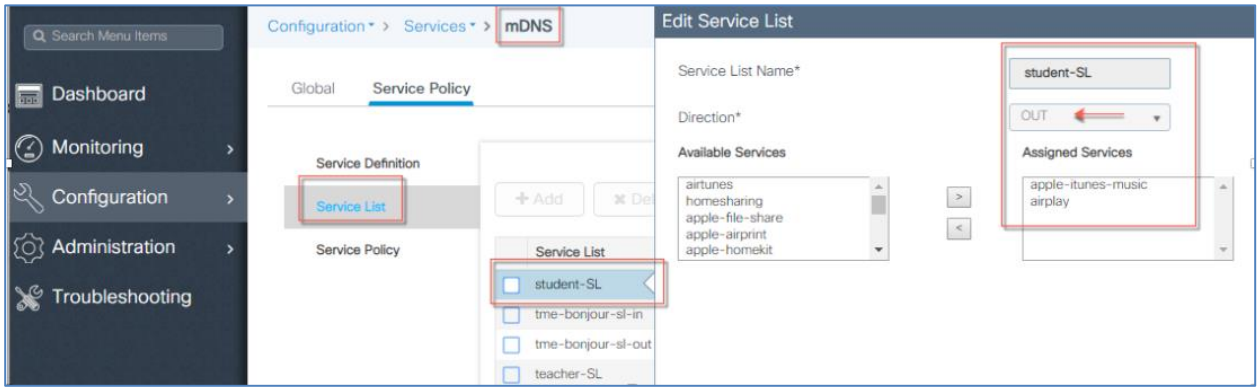
On the Teacher's network the following Service Types will be available: AirPrint, AirPlay, File Share. And on the Student's network AirPlay and iTunes Sharing services will be available.



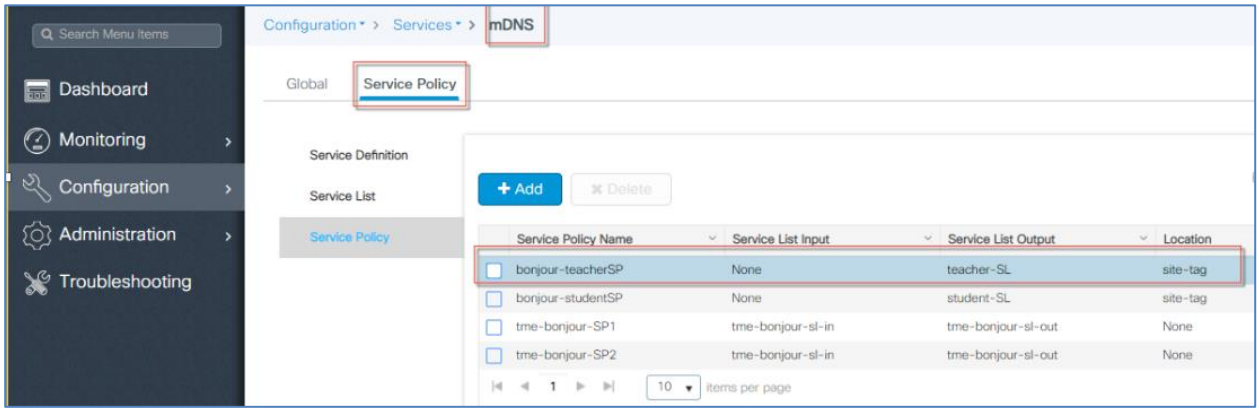
As per design example, we create an mDNS Service List for the Teacher and Student.

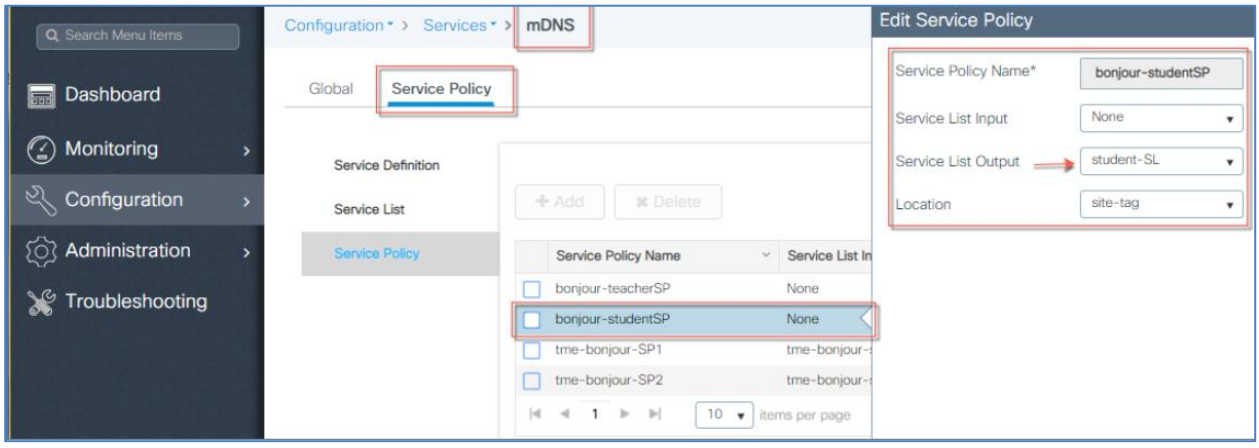
We create the teacher-SL Out and assign Services as described above.

Then create a student-SL Out and assign services as described above.



As per the design example, create Service Policy for Teacher and Student and assign the Service-List OUT to each Service Policy. Service Policy will be the AAA controlled policy and assigned based on the user authentication.

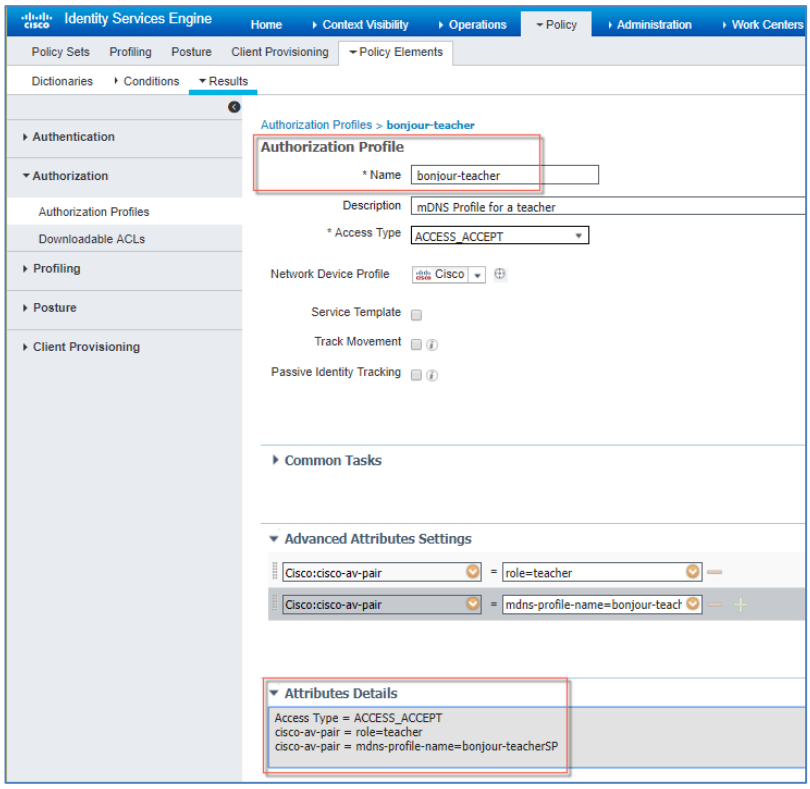




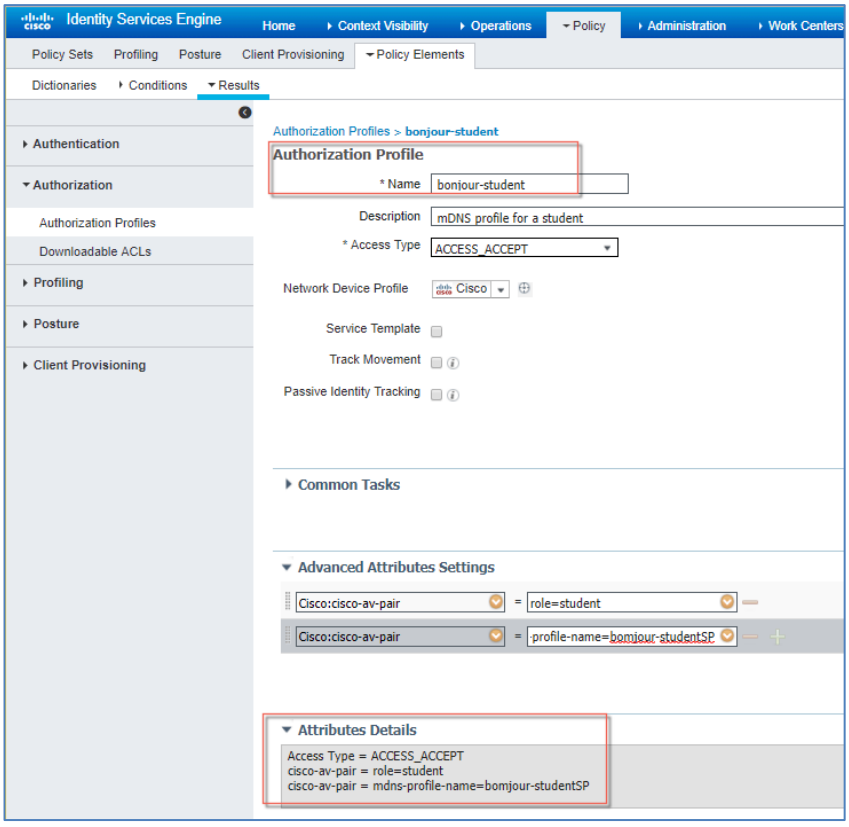
In the next configure AAA server, in this example ISE server is used. Two mDNS profiles have been created on the ISE server.

Per Authorization Profile configured on ISE, when Teachers authenticate to the WLAN "Bonjour" they will be assigned Service Policy "**bonjour-teacherSP**" and when Students authenticate to the WLAN "Bonjour" they will be assigned Service Policy "**bonjour-studentSP**"

Authorization profile for "bonjour-teacher" is shown below

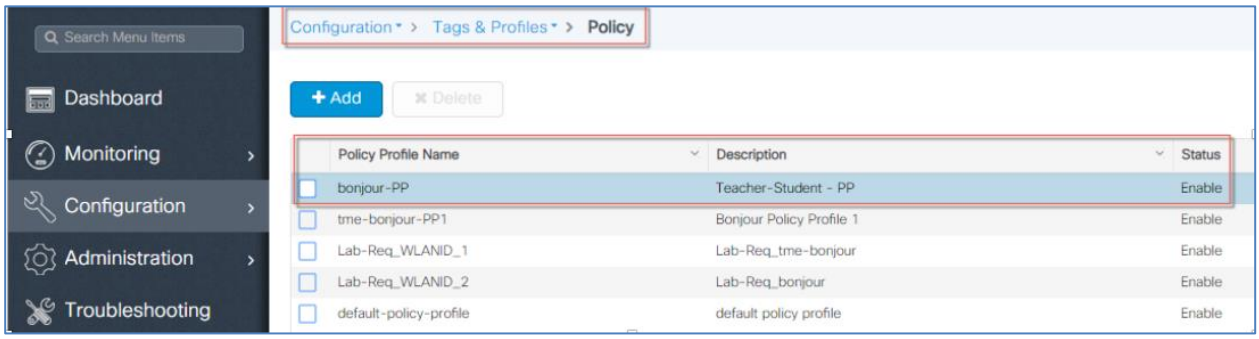


Authorization profile for “bonjour-student” is shown below

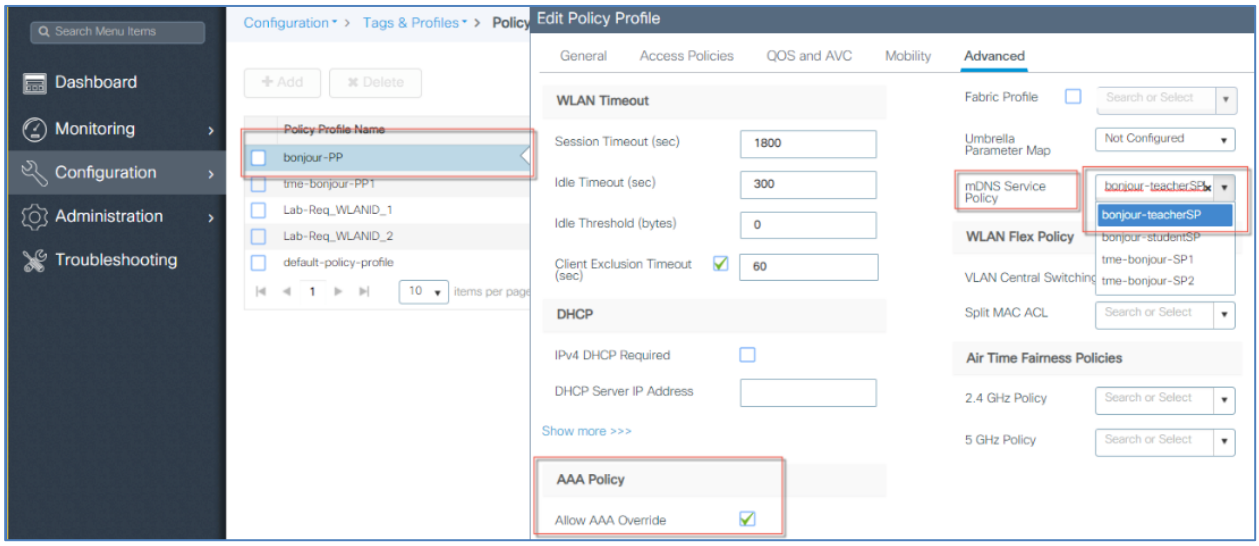


In the next step create a “bonjour-PP” Policy Profile for both Teacher and Student. Policy Profile will be assigned with Service Policy “based on

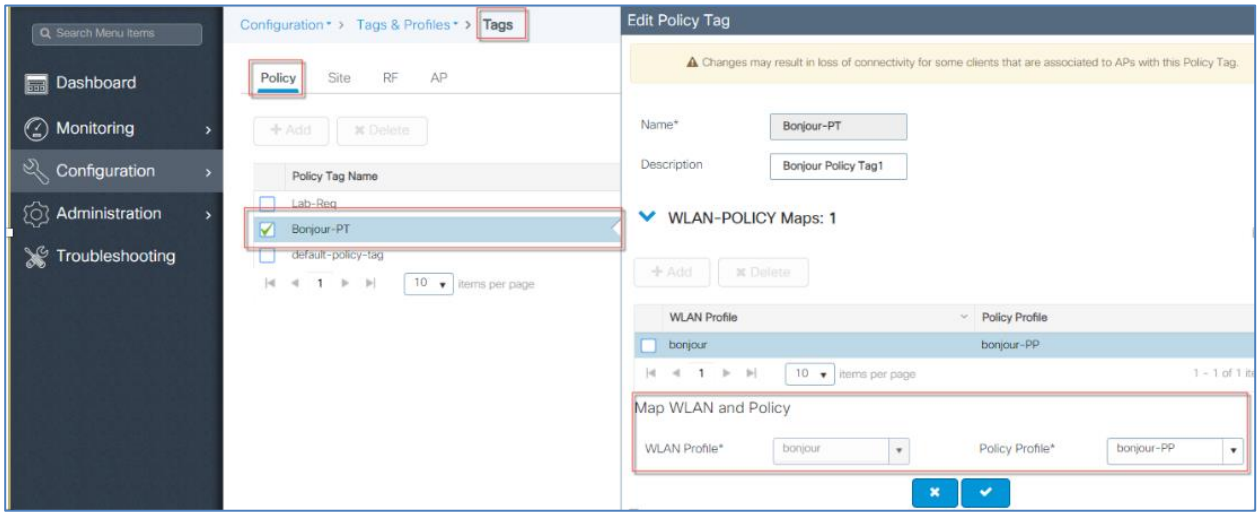
user authentication with AAA override



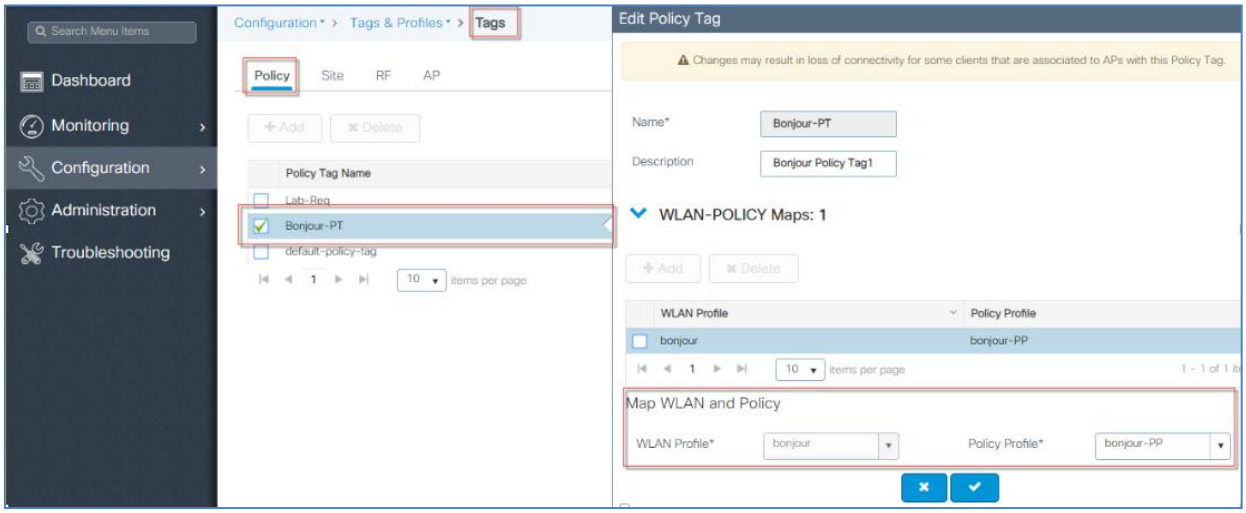
In the “bonjour-PP” policy profile the mDNS Service Policy will be selected based on the user authentication. If “teacher” gets authenticated then “bonjour-teacher-SP” gets elected and if the “student” gets authenticated then “bonjour-student-SP” gets elected in the Policy Profile.



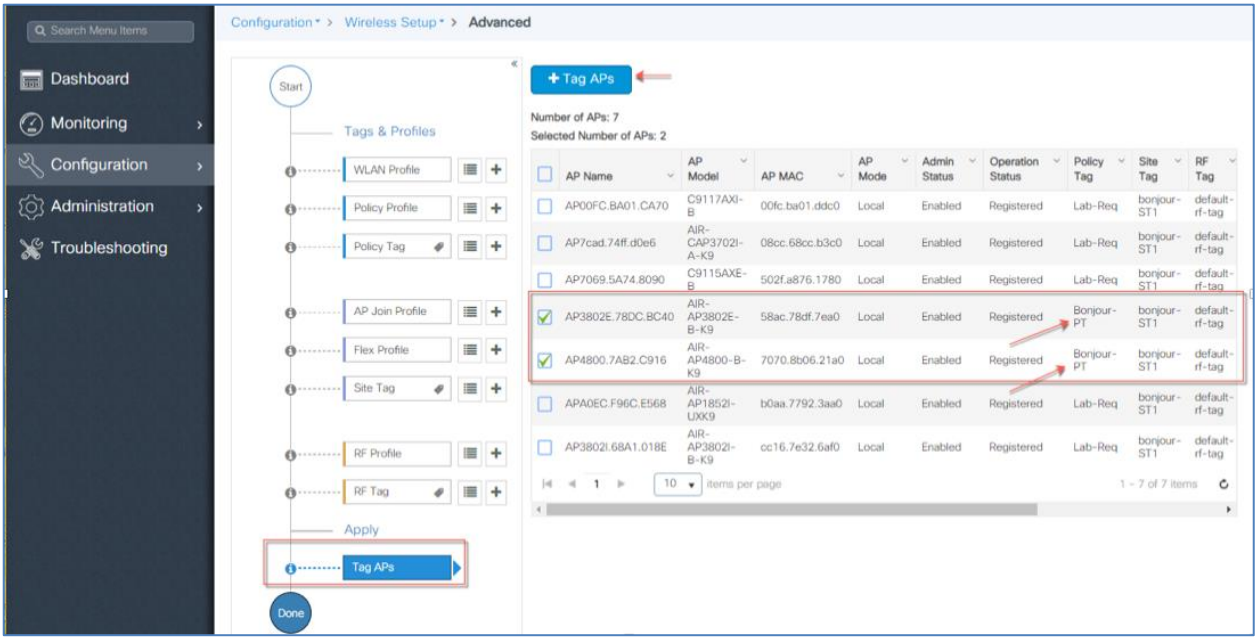
With AAA override and Policy Tag setting when Teachers connect to WLAN “bonjour” they will be assigned Service Policy “bonjour-teacherSP” via the previously configured Policy Profile.

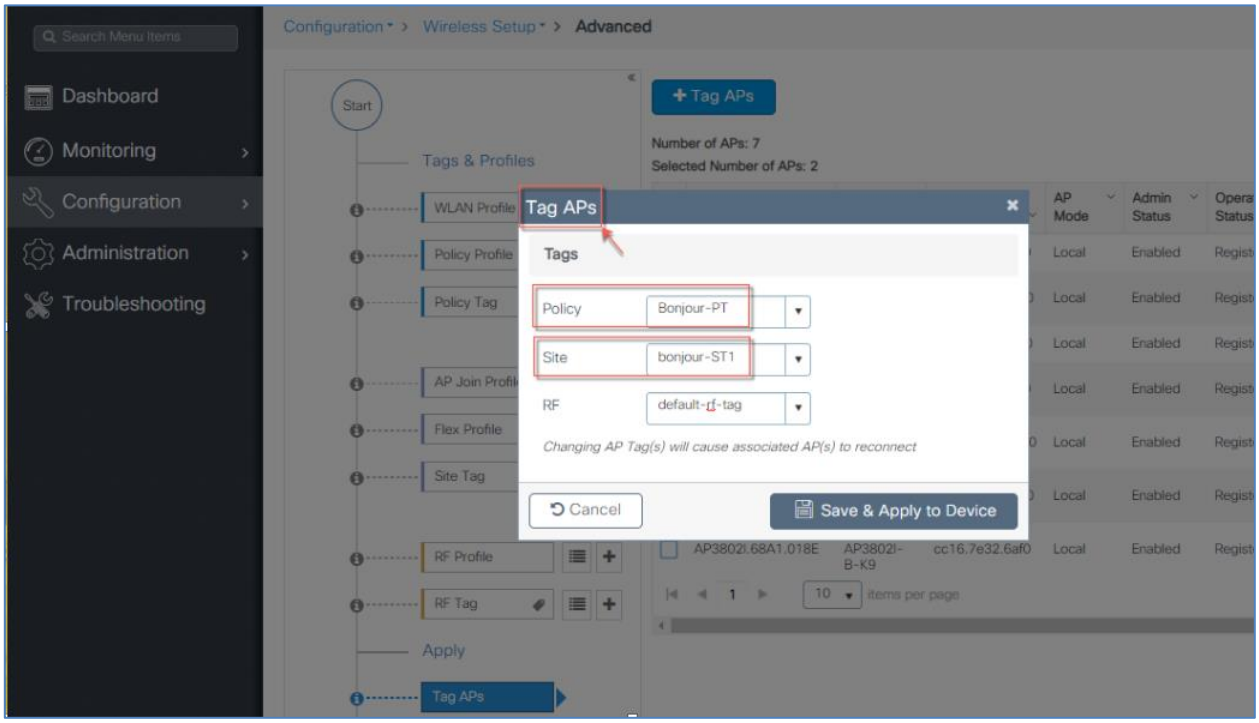


With AAA override and Policy Tag setting when Students connect to WLAN “bonjour” they will be assigned Service Policy Profile “bonjour-studentSP” via the previously configured Policy Profile.



Lastly, as per design requirement, tag all APs or selected APs with the previously configured Policy and Site Tags.




Search Menu Items
Dashboard
Monitoring
Configuration
Administration
Troubleshooting
Configuration > Wireless Setup > Advanced
Start
Tags & Profiles
WLAN Profile
Policy Profile
Policy Tag
AP Join Profile
Flex Profile
Site Tag
RF Profile
RF Tag
Apply
Tag APs
+ Tag APs
Number of APs: 7
Selected Number of APs: 2
Tag APs
Tags
Policy
Bonjour-PT
Site
bonjour-ST1
RF
default-rf-tag
Changing AP Tag(s) will cause associated AP(s) to reconnect
Cancel
Save & Apply to Device
AP Mode
Admin Status
Local
Enabled
AP3802I-B-K9
cc16.7e32.6af0
items per page

Legal Information

Cisco Trademark

Cisco Copyright