Multicast Domain Name System

Introduction to mDNS Gateway

Multicast Domain Name System (mDNS) is an Apple service discovery protocol which locates devices and services on a local network with the use of mDNS service records.

The Bonjour protocol operates on service announcements and queries. Each query or advertisement is sent to the Bonjour multicast address ipv4 224.0.0.251 (ipv6 FF02::FB). This protocol uses mDNS on UDP port 5353.

The address used by the Bonjour protocol is link-local multicast address and therefore is only forwarded to the local L2 network. As, multicast DNS is limited to an L2 domain for a client to discover a service it has to be part of the same L2 domain, This is not always possible in any large scale deployment or enterprise.

In order to address this issue, the Cisco Catalyst 9800 Series Wireless Controller acts as a Bonjour Gateway. The controller then listens for Bonjour services, caches these Bonjour advertisements (AirPlay, AirPrint, and so on) from the source or host. For example, Apple TV responds back to Bonjour clients when asked or requested for a service. This way you can have sources and clients in different subnets.

By default, the mDNS gateway is disabled on the controller. To enable mDNS gateway functionality, you must explicitly configure mDNS gateway using CLI or Web UI.

Prerequisite

Since the Cisco Catalyst 9800 Series Wireless Controller will respond and advertise for services cached when acting as a Bonjour Gateway, it must have an SVI interface with a valid IP address on every VLAN where mDNS is allowed or used. This will be the source IP address of those mDNS packets that are coming out from the controller acting as mDNS Gateway.

Guidelines and Restrictions for Configuring mDNS AP

- Cisco recommends deploying scalable Wide Area Bonjour to route mDNS service between Wired and Wireless networks. Cisco Catalyst 9800 Series Wireless LAN Controller (WLC) introduces a new mDNS gateway called Service-Peer mode to replace the classic mDNS flood-n-learn to support Enterprise-grade scalable, stateful, and reliable complete unicast-based mDNS service-routing with upstream gateway Cisco Catalyst 9000 Series Switches. For more information, see Part: Cisco DNA Service for Bonjour.

- The mDNS AP (classic flood-n-learn based feature) is enhanced with complete unicast-based service-routing using Cisco Wide Area Bonjour supporting flood-free Wired and Wireless networks to overcome several operational, scalable, and service resiliency challenges.

- The mDNS AP extends the mDNS flood from Wired VLANs to AP and further extends over the CAPWAP tunnel to WLC for central processing across Core network. Cisco recommends that the mDNS AP must be considered only for small network environments.

- The mDNS AP is supported only in Local and Monitor modes. If Cisco Wireless AP is in FlexConnect mode, the Fabric mode AP does not support mDNS AP feature. For more information on how to enable the mDNS service-routing for various distributed Wireless modes, see Part: Cisco DNA Service for Bonjour.

- Wireless users connected to mDNS AP may not be able to browse the Wired mDNS services across flooded Wired VLAN to mDNS AP.

- The Wired mDNS service-provider VLANs must be extended to flood the mDNS traffic up to mDNS AP ethernet port in trunk mode settings. The Wired VLAN extension to mDNS AP may include other Wired flood traffic, such as Broadcast, Unknown Unicast, and Layer 2 Multicast that impacts the mDNS AP scale and performance.

- It is recommended to have minimum one mDNS AP for each Layer 3 Access switch. All Wired mDNS traffic is flooded using alternate L2 methods, if single mDNS AP is shared between multiple Layer 3 Access switch.

- The maximum mDNS AP scale limit for each Cisco Catalyst 9800 Series Wireless LAN Controller (WLC) is limited.

- The maximum mDNS Wired VLAN count for each WLC is limited.

- The old Wired mDNS service entry continues to be advertised to all Wireless users up to 4500 seconds based on the mDNS cache timers on WLC. The stale entries require manual clearing from local cache in WLC.

- The mDNS AP does not support mDNS Query packet suppression or rate-limiter in AP. The Wired mDNS flood from all Wired VLAN is extended to WLC for central processing of policy enforcement.

- The maximum number of flooded packets for each second processing from Wired VLANs to mDNS AP is limited. The mDNS AP performance and reliability may get compromised in large network environments.

- A maximum of 10 Wired VLANs' mDNS flood can be extended to mDNS AP. Combined large Wired VLAN and mDNS AP scale may impact scale and performance in AP and WLC.

- Only one mDNS AP is supported for each Wired VLAN. Multiple mDNS APs cannot be configured to map the same Wired VLAN ID as it causes service instability and duplicate processing.

- High Availability is not supported in multiple mDNS AP. The mDNS services across Wired and Wireless network gets disrupted when connectivity to mDNS AP is lost due to any kinds of failures.

- Only one Wired mDNS service-policy is supported for all network-wide mDNS AP.

- All WLAN users can discover all flooded Wired mDNS services without granular Location-Based service. The mDNS AP in large and flooded network impacts user-experience on mobile devices.

- The mDNS AP do not support IPv6 for Wired mDNS service-provider or service-receiver. Only IPv4 is supported.

- The mDNS AP do not support role-based mDNS service filtering between Wired and Wireless networks.

- The mDNS AP do not detect and auto-resolve duplicate mDNS service-instance names across Wired VLANs. The Cisco Catalyst 9800 Series Wireless LAN Controller (WLC) discovers and records the first service instance with unique name in its local cache database. If a duplicate service instance name is discovered, the WLC rejects the duplicate name and does not distribute it to the Wireless clients.

- Wireless multicast link-local is enabled by default. When wireless link-local is enabled, only mDNS Bridging mode is supported. If you require mDNS Gateway for wired services, disable wireless link-local.

- If you have a FlexConnect AP as an mDNS gateway, ensure that you do not use "." in the service provider name, as it is not supported.

Enabling mDNS Gateway (GUI)

Procedure

Step 1	Choose **Configuration > Services > mDNS**.
Step 2	In the **Global** section, toggle the slider to enable or disable the **mDNS Gateway**.
Step 3	From the **Transport** drop-down list, choose one of the following types:

- **ipv4**
- **ipv6**
- **both**

Step 4	Enter an appropriate timer value in **Active-Query Timer**. The valid range is between 1 to 120 minutes. The default is 30 minutes.
Step 5	From the **mDNS-AP Service Policy** drop-down list, choose an mDNS service policy.

Note Service policy is optional only if mDNS-AP is configured. If mDNS-AP is not configured, the system uses default-service-policy.

Step 6	Click **Apply**.

Enabling or Disabling mDNS Gateway (GUI)

Procedure

Step 1	Choose **Configuration** > **Services** > **mDNS** > **Global**.
Step 2	Enable or disable the **mDNS Gateway** toggle button.
Step 3	Choose **ipv4** or **ipv6** or **both** from the **Transport** drop-down list.
Step 4	Enter the **Active-Query Timer**.
Step 5	Click **Apply**.

Enabling or Disabling mDNS Gateway (CLI)

Note
- mDNS gateway is disabled by default globally on the controller.

- You need both global and WLAN configurations to enable mDNS gateway.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** Device> enable	Enables privileged **EXEC** mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**mdns-sd gateway** **Example:** Device(config)# mdns-sd gateway	Enables mDNS gateway.
Step 4	**transport** {**ipv4** | **ipv6** | **both**} **Example:** Device(config-mdns-sd)# transport ipv4	Processes mDNS message on a specific transport. Here, **ipv4** signifies that the IPv4 mDNS message processing is enabled. This is the default value. **ipv6** signifies that the IPv6 mDNS message processing is enabled. **both** signifies that the IPv4 and IPv6 mDNS message is enabled for each network.
Step 5	**active-query timer** *active-query-periodicity* **Example:** Device(config-mdns-sd)# active-query timer 15	Changes the periodicity of mDNS multicast active query. **Note** An active query is a periodic mDNS query to refresh dynamic cache. Here, *active-query-periodicity* refers to the active query periodicity in Minutes. The valid range is from 1 to 120 minutes. Active query runs with a default periodicity of 30 minutes.
Step 6	**exit** **Example:** Device(config-mdns-sd)# exit	Returns to global configuration mode.

Creating Default Service Policy

When the mdns gateway is enabled on any of the WLANs by default, mdns-default-service-policy is associated with it. Default service policy consists of default-service-list and their details are explained in this section. You can override the default service policy with a custom service policy.

Procedure

Step 1	Create a service-definition if the service is not listed in the preconfigured services.
Step 2	Create a service list for IN and OUT by using the service-definitions.
Step 3	Use the existing service list to create a new service. For more information, refer to *Creating Service Policy* section.
Step 4	Attach the mdns-service-policy to the profile or VLAN that needs to be enforced.
Step 5	To check the default-mdns-service list, use the following command: **show mdns-sd default-service-list**

Creating Custom Service Definition (GUI)

Procedure

Step 1	Choose **Configuration > Services > mDNS**.
Step 2	In the **Service Definition** section, click **Add**.
Step 3	In the **Quick Setup: Service Definition** page that is displayed, enter a name and description for the service definition.
Step 4	Enter a service type and click + to add the service type.
Step 5	Click **Apply to Device**.

Creating Custom Service Definition

Service definition is a construct that provides an admin friendly name to one or more mDNS service types or A pointer (PTR) Resource Record Name.

By default, few built-in service definitions are already predefined and available for admin to use.

In addition to built-in service definitions, admin can also define custom service definitions.

You can execute the following command to view the list of all the service definitions (built-in and custom):

```
Device# show mdns-sd master-service-list
```

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged **EXEC** mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mdns-sd service-definition** *service-definition-name* **Example:** `Device(config)# mdns-sd service-definition CUSTOM1`	Configures mDNS service definition. **Note** • All the created custom service definitions are added to the primary service list. • Primary service list comprises of a list of custom and built-in service definitions.
Step 4	**service-type** *string* **Example:** `Device(config-mdns-ser-def)# service-type _custom1._tcp.local`	Configures mDNS service type.
Step 5	**exit** **Example:** `Device(config-mdns-ser-def)# exit`	Returns to global configuration mode.

Creating Service List (GUI)

Procedure

Step 1   Choose **Configuration > Services > mDNS**.

Step 2   In the **Service List** section, click **Add**.

Step 3   In the **Quick Setup: Service List** page that is displayed, enter a name for the service list.

Step 4   From the **Direction** drop-down list, choose **IN** for inbound filtering or **OUT** for outbound filtering.

Step 5   From the **Available Services** drop-down list, choose a service type to match the service list.

     Note   To allow all services, choose the **all** option.

Step 6   Click **Add Services**.

Step 7   From the **Message Type** drop-down list, choose the message type to match from the following options:

- **any**—To allow all messages.

- **announcement**—To allow only service advertisements or announcements for the device.

- **query**—To allow only a query from the client for a service in the network.

Step 8   Click **Save** to add services.

Step 9   Click **Apply to Device**.

Creating Service List

mDNS service list is a collection of service definitions.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged **EXEC** mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**mdns-sd service-list** *service-list-name* {**IN** | **OUT**}
	
	Example:
	`Device(config)# mdns-sd service-list Basic-In IN`
	`Device(config)# mdns-sd service-list Basic-Out OUT`
		Configures mDNS service list.
		
		• **IN**: Provides inbound filtering.
		
		• **Out**: Provides outbound filtering.

Step 4	**match** *service-definition-name* **message-type** {**announcement** | **any** | **query**} **Example:** `Device(config-mdns-sl-in)# match CUSTOM1 message-type query`	Matches the service to the message type. Here, *service-definition-name* refers to the names of services, such as, airplay, airserver, airtunes, and so on.

	Command or Action	Purpose
		Note To add a service, the service name must be part of the primary service list.
		If the mDNS service list is set to IN, you get to view the following command: **match service-definition-name message-type {announcement | any | query}**.
		If the mDNS service list is set to Out, you get to view the following command: **match service-definition-name**.
		(OR)
Step 5	**match all message-type {announcement | any | query}** **Example:** Device(config-mdns-sl-in)# match all message-type query	Matches all services to the message type.
		Note To add a service, the service name must be part of the primary service list.
		If the mDNS service list is set to IN, you get to view the following command: **match all message-type {announcement | any | query}**.
		If the mDNS service list is set to OUT, you get to view the following command: **match all**.
		In case of IN or OUT filter, if any of the service contains the same or subset of the message type (query or announcement), the **match all** is not allowed unless the existing services are removed.
Step 6	**show mdns-sd service-list {direction | name }**	Displays inbound or outbound direction list of the configured service-list to classify matching service-types for service-policy. The list can be filtered by name or specific direction.
Step 7	**exit** **Example:** Device(config-mdns-sl-in)# exit	Returns to global configuration mode.

Creating Service Policy (GUI)

Procedure

Step 1	Choose **Configuration > Services > mDNS**.
Step 2	In the **Service Policy** section, click **Add**.
Step 3	In the **Quick Setup: Service Policy** page that is displayed, enter a name for the service policy.
Step 4	From the **Service List Input** drop-down list, choose one of the types.
Step 5	From the **Service List Output** drop-down list, choose one of the types.
Step 6	From the **Location** drop-down list, choose the location you want to associate with the service list.
Step 7	Click **Apply to Device**.

Creating Service Policy

mDNS service policy is used for service filtering while learning services or responding to queries.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged **EXEC** mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mdns-sd service-policy** *service-policy-name* **Example:** `Device(config)# mdns-sd service-policy mdns-policy1`	Enables mDNS service policy.
Step 4	**location {lss	site-tag}** **Example:**	Filters mDNS service types based on LSS or site-tag.

	Command or Action	**Purpose**
	`Device(config-mdns-ser-pol)# location lss`	**Note** In Location Specific Services (LSS) based filtering, the mDNS gateway responds with the service instances learnt from the neighboring APs of the querying client AP. Other service instances for the rest of APs are filtered. In Site tag based filtering, the mDNS gateway responds with the service instances that belong to the same site-tag as that of querying client. The mDNS gateway responds back with wired services even if the location based filtering is configured.
Step 5	**service-list** *service-list-name* {**IN** | **OUT**} **Example:** `Device(config-mdns-ser-pol)# service-list VLAN100-list IN`	Configures various service-list names for IN and OUT directions. **Note** If an administrator decides to create or use a custom service policy, then the custom service policy must be configured with service-lists for both directions (IN and OUT); otherwise, the mDNS Gateway will not work (will not learn services if there is no IN service-list, or will not reply or announce services learned if there is no OUT service-list).
Step 6	**exit** **Example:** `Device(config-mdns-ser-pol)# exit`	Returns to global configuration mode.

Configuring a Local or Native Profile for an mDNS Policy

When an administrator configures local authentication and authorization and does not expect to get any mDNS policy from the AAA server, the administrator can configure a local or native profile to select a mDNS policy based on user, role, or device type. When this local or native profile is mapped to the wireless profile policy, mDNS service policy is applied on the mDNS packets that are processed on that WLAN.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**service-template** *template-name* **Example:** `Device(config)# service-template mdns`	Configures the service-template or identity policy.

Step 3	**mdns-service-policy** *mdns-policy-name*
	
	Example:
	
	`Device(config-service-template)#`
	`mdns-service-policy mdnsTV`
		Configures the mDNS policy.

Step 4	**exit** **Example:** `Device(config-service-template)# exit`	Returns to global configuration mode.

Configuring an mDNS Flex Profile (GUI)

Procedure

Step 1   Choose **Configuration > Services > mDNS**.

Step 2   In the **mDNS Flex Profile** section, click **Add**.
The **Add mDNS Flex Profile** window is displayed.

Step 3   In the **Profile Name** field, enter the flex mDNS profile name.

Step 4   In the **Service Cache Update Timer** field, specify the service cache update time. The default value is 1 minute. The valid range is from 1 to 100 minutes.

Step 5   In the **Statistics Update Timer** field, specify the statistics update timer. The default value is 1 minute. The valid range is from 1 to 100 minutes.

Step 6   In the **VLANs** field, specify the VLAN ID. You can enter multiple VLAN IDs separated by commas, or enter a range of VLAN IDs. Maximum number of VLANs allowed is 16.

Step 7   Click **Apply to Device**.

Configuring an mDNS Flex Profile (CLI)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 2	**mdns-sd flex-profile** *mdns-flex-profile-name*
	
	Example:
	
	`Device(config)# mdns-sd flex-profile`
	`mdns-flex-profile-name`
		Enters the mDNS Flex Profile mode.

Step 3	**update-timer service-cache** *service-cache timer-value <1-100>*
	
	Example:
	
	`Device(config-mdns-flex-profile)#`
	`update-timer service-cache 60`
		Configures the mDNS update service cache timer for the flex profile.
		
		The default value is 1 minute. Value range is between 1 minute and 100 minutes.

Step 4	**update-timer statistics** *statistics timer-value <1-100>*
	
	Example:
	
	`Device(config-mdns-flex-profile)#`
	`update-timer statistics 65`
		Configures the mDNS update statistics timer for the flex profile.
		
		The default value is 1 minute. The valid range is from 1 to 100 minutes.

Step 5	**wired-vlan-range** *wired-vlan-range value*
	
	Example:
	
	`Device(config-mdns-flex-profile)#`
	`wired-vlan-range 10 - 20`
		Configures the mDNS wired VLAN range for the flex profile.
		
		The default value is 1 minute. The valid range is from 1 minute to 100 minutes.

Applying an mDNS Flex Profile to a Wireless Flex Connect Profile (GUI)

Procedure

Step 1	Choose **Configuration > Tags & Profiles > Flex**.
Step 2	Click **Add**. The **Add Flex Profile** window is displayed.
Step 3	Under the **General** tab, from the **mDNS Flex Profile** drop-down list, choose a flex profile name from the list.

Step 4   Click **Apply to Device**.

Applying an mDNS Flex Profile to a Wireless Flex Connect Profile (CLI)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 2	**wireless profile flex** *wireless-flex-profile-name*
	
	Example:
	`Device# wireless profile flex`
	`wireless-flex-profile-name`
		Enters wireless flex profile configuration mode.

Step 3	**mdns-sd** *mdns-flex-profile*
	
	Example:
	`Device(config-wireless-flex-profile)#`
	`mdns-sd mdns-flex-profile-name`
		Enables the mDNS features for all the APs in the profile

Enabling the mDNS Gateway on the VLAN Interface

This procedure configures the mDNS service policy for a specific VLAN. This allows the administrator to configure different settings to the mDNS packets on per VLAN interface basis and not on per WLAN basis.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface vlan** *vlan-interface-number* **Example:** `Device(config)# interface vlan 200`	Configures a VLAN ID and enters interface configuration mode.
Step 3	**ip address** *ip-address subnet-mask* **Example:**	Configures the IP address for the interface.

	Command or Action	Purpose
	`Device(config-if)# ip address 111.1.1.1` `255.255.255.0`	
Step 4	**mdns-sd gateway** **Example:** `Device(config-if)# mdns-sd gateway`	Enables mDNS configuration on a VLAN interface.

Step 5	**service-policy** *service-policy-name*
	
	Example:
	`Device(config-if-mdns-sd)# service-policy`
	`test-mDNS-service-policy`
		Configures the service policy.
		
		Note If specific *service-policy-name* is not defined, the VLAN will use the **default-mdns-service-policy** by default.
		
		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.

Step 6	**end** **Example:** `Device(config-if-mdns-sd)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Location-Based Service Filtering

Prerequisite for Location-Based Service Filtering

You need to create the Service Definition and Service Policy. For more information, see Creating Custom Service Definition section and Creating Service Policy section.

Configuring mDNS Location-Based Filtering Using SSID

When a service policy is configured with the SSID as the location name, the response to the query will be the services that were learnt on that SSID.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	**Purpose**
Step 2	**mdns-sd service-policy** *service-policy-name* **Example:** `Device(config)# mdns-sd service-policy mdns-policy1`	Configures the service policy.
Step 3	**location ssid** **Example:** `Device(config-mdns-ser-pol)# location ssid`	Configures location-based filtering using SSID.
Step 4	**end** **Example:** `Device(config-mdns-ser-pol)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configuring mDNS Location-Based Filtering Using AP Name

When a service policy is configured with the AP name as the location, the response to the query will be the services that were learnt on that AP.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mdns-sd service-policy** *service-policy-name* **Example:** `Device(config)# mdns-sd service-policy mdns-policy1`	Configures the service policy.
Step 3	**location ap-name** **Example:** `Device(config-mdns-ser-pol)# location ap-name`	Configures location-based filtering using an AP name.
Step 4	**end** **Example:** `Device(config-mdns-ser-pol)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configuring mDNS Location-Based Filtering Using AP Location

When a service policy is configured with location as the AP-location, the response to the query will be the services that were learnt on all the APs using the same AP "location" name (not to be confused with "site-tag").

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mdns-sd service-policy** *service-policy-name* **Example:** `Device(config)# mdns-sd service-policy mdns-policy1`	Configures the service policy.
Step 3	**location ap-location** **Example:** `Device(config-mdns-ser-pol)# location ap-location`	Configures location-based filtering using the AP location.
Step 4	**end** **Example:** `Device(config-mdns-ser-pol)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configuring mDNS Location-Based Filtering Using Regular Expression

- When a service policy is configured with the location as a regular expression that matches the corresponding AP name, the response to the query will be the services that were learnt on a group of APs based on the AP name.

- When a service policy is configured with the location as a regular expression that matches the corresponding AP location, the response to the query will be the services that were learnt on a group of APs based on the AP location.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mdns-sd service-policy** *service-policy-name* **Example:** `Device(config)# mdns-sd service-policy mdns-policy1`	Configures the service policy.
Step 3	**location regex** {**ap-location** *regular-expression* | **ap-name** *regular-expression*} **Example:**	Configures location-based filtering using regular expression.

	Command or Action	Purpose

	`Device(config-mdns-ser-pol)# location`
	`regex ap-location dns_location`
	
	`Device(config-mdns-ser-pol)# location`
	`regex ap-name dns_name`	

Step 4	**end** **Example:** `Device(config-mdns-ser-pol)# end` **Note** To filter the services for which AP names have the specific keyword such as *AP-2FLR-SJC-123*, you can use the regex AP name as *AP-2FLR-* to match the services that are learnt from the set of access points.	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Configuring mDNS AP

In most of the deployments, the services may be available in VLANs that the APs can hear in the wired side (allowed in the switchport where the AP is directly connected: its own VLAN, or even more VLANs if switchport is a trunk).

The following procedure shows how to configure mDNS AP:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mdns-sd gateway** **Example:** `Device(config)# mdns-sd gateway`	Configures the mDNS gateway.
Step 3	**ap name** *ap-name* **mdns-ap enable vlan** *vlan-id* **Example:** `Device# ap name ap1 mdns-ap enable vlan 22`	Enables mDNS on the AP, and configures a VLAN for the mDNS AP.
Step 4	**ap name** *ap-name* **mdns-ap vlan add** *vlan-id* **Example:** `Device# ap name ap1 mdns-ap vlan add 200`	Adds a VLAN to the mDNS AP. *vlan-id* ranges from 1 to 4096.

	Command or Action	Purpose
Step 5	**ap name** *ap-name* **mdns-ap vlan del** *vlan-id* **Example:** `Device# ap name ap1 mdns-ap vlan del 2`	Deletes a VLAN from the mDNS AP.
Step 6	**ap name** *ap-name* **mdns-ap disable** **Example:** `Device# ap name ap1 mdns-ap disable`	(Optional) Disables the mDNS AP.
Step 7	**end** **Example:** `Device# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode. **Note** You can configure a maximum of 10 VLANs per AP.

Enabling mDNS Gateway on the RLAN Interface

By configuring the mDNS gateway mode on the RLAN interface, you can configure the mDNS service policy for a specific RLAN.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ap remote-lan profile-name** *remote-lan-profile-name rlan-id* **Example:** `Device(config)# ap remote-lan profile-name rlan_test_1 1`	Configures a remote LAN profile. • *remote-lan-profile*: Remote LAN profile name. Range is from 1 to 32 alphanumeric characters. • *rlan-id*: Remote LAN identifier. Range is from 1 to 128. **Note** You can create a maximum of 128 RLANs. Also, you cannot use the *rlan-id* of an existing RLAN while creating another RLAN.
Step 3	**mdns-sd-interface {gateway | drop}** **Example:** mdns-sd-interface	Enables mDNS configuration on an RLAN interface.

	Command or Action	**Purpose**
	`Device(config-remote-lan)#` `mdns-sd-interface gateway`	
Step 4	**no shutdown** **Example:** `Device(config-remote-lan)# no shutdown`	Restarts the RLAN profile.
Step 5	**exit** **Example:** `Device(config-remote-lan)# exit`	Exits remote LAN configuration mode.

Step 6	**ap remote-lan-policy policy-name** *profile name*
	
	Example:
	`Device(config)# ap remote-lan-policy`
	`policy-name rlan_named_pp1`
		Configures the RLAN policy profile and enters wireless policy configuration mode.

Step 7	**mdns-sd service-policy** *service-policy-name*
	
	Example:
	`Device(config-remote-lan-policy)#`
	`mdns-sd service-policy mdnsTV6`
		Enables an mDNS service policy.

Step 8	**central switching**
	
	Example:
	`Device(config-remote-lan-policy)#`
	`central switching`
		Configures the RLAN for central switching.

Step 9	**central dhcp**
	
	Example:
	`Device(config-remote-lan-policy)#`
	`central dhcp`
		Configures the central DHCP for centrally switched clients.

Step 10	**vlan** *vlan-name*
	
	Example:
	`Device(config-remote-lan-policy)# vlan`
	`141`
		Assigns the profile policy to a VLAN.

Step 11	**no shutdown**
	
	Example:
	`Device(config-remote-lan-policy)# no`
	`shutdown`
		Restarts the RLAN profile.

Step 12	**wireless tag policy** *policy-tag-name*
	
	Example:
	`Device(config)# wireless tag policy`
	`rlan_pt_1`
		Configures a policy tag.

	Command or Action	Purpose

Step 13	**remote-lan** *remote-lan-profile-name* **policy** *rlan-policy-profile-name* **port-id** *port-id*
	
	Example:
	`Device(config-policy-tag)# remote-lan rlan_test_1 policy rlan_named_pp1 port-id 1`
	
	`Device(config-policy-tag)# remote-lan rlan_test_1 policy rlan_named_pp1 port-id 2`
	
	`Device(config-policy-tag)# remote-lan rlan_test_1 policy rlan_named_pp1 port-id 3`
	
	`Device(config-policy-tag)# remote-lan rlan_test_1 policy rlan_named_pp1 port-id 4`
		Maps the RLAN policy profile to the RLAN profile.
		
		• *remote-lan-profile-name*: Name of the RLAN profile.
		
		• *rlan-policy-profile-name*: Name of the policy profile.
		
		• *port-id*: LAN port number on the access point. Range is from 1 to 4.

Step 14	**exit** **Example:** `Device(config-policy-tag)# exit`	Returns to global configuration mode.
Step 15	**ap** *mac-address* **Example:** `Device (config)# ap 0042.5AB6.0EF0`	Configures the AP and enters the AP tag configuration mode. **Note** Use the Ethernet MAC address.
Step 16	**policy-tag** *policy-tag-name* **Example:** `Device (config-ap-tag)# policy-tag rlan_pt_1`	Maps a policy tag to the AP.
Step 17	**end** **Example:** `Device(config-guest-lan)# end`	Returns to privileged EXEC mode.

Enabling mDNS Gateway on Guest LAN Interface

By configuring the mDNS gateway mode on a Guest LAN interface, you can configure the mDNS service policy for a specific Guest LAN interface.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 2	**guest-lan profile-name** *guest_lan_profile_name num* **wired-vlan** *wired_vlan_num*
	
	Example:
	`Device(config)# guest-lan profile-name open 1 wired-vlan 666`
		Configures guest LAN profile with a wired VLAN.
		
		Note Configures the wired VLAN only for the Guest Foreign controller.
		
		    • *num*: Guest LAN identifier. The valid range is from 1 to 5.
		
		    • *wired_vlan_num*: Wired VLAN number. The valid range is from 1 to 4094.

Step 3	**guest-lan profile-name** *guest_lan_profile_name num* **Example:** `Device(config)# guest-lan profile-name open 1`	Configures the guest LAN profile without a VLAN for the Guest Anchor controller.
Step 4	**mdns-sd-interface {gateway	drop}** **Example:** `Device(config-guest-lan)# mdns-sd gateway`	Configures the mDNS gateway for a Guest LAN. **Note** You need to enable mDNS gateway globally for the Guest LAN to work.
Step 5	**end** **Example:** `Device(config-guest-lan)# end`	Returns to privileged EXEC mode. Alternatively, you can also press **Ctrl-Z** to exit global configuration mode.

Associating mDNS Service Policy with Wireless Profile Policy (GUI)

Procedure

Step 1   Choose **Configuration** > **Tags & Profiles** > **Policy**.

Step 2   Click the **policy profile** name.

Step 3   In the **Advanced** tab, choose the mDNS service policy from the **mDNS Service Policy** drop-down list.

Step 4   Click **Update & Apply to Device**.

Associating mDNS Service Policy with Wireless Profile Policy

✎

Note You must globally configure the mDNS service policy before associating it with the wireless profile policy.

A default mDNS service policy is already attached once the wireless profile policy is created. You can use the following commands to override the default mDNS service policy with any of your service policy:

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wireless profile policy** *profile-policy* **Example:** `Device(config)# wireless profile policy default-policy-profile`	Configures wireless profile policy. Here, *profile-policy* refers to the name of the WLAN policy profile.
Step 3	**mdns-sd service-policy** *custom-mdns-service-policy* **Example:** `Device(config-wireless-policy)# mdns-sd service-policy custom-mdns-service-policy`	Associates an mDNS service policy with the wireless profile policy. The default mDNS service policy name is **default-mdns-service-policy**.

Command or Action	Purpose
	Note

Command or Action	Purpose
	The **default-mdns-profile-policy** uses **default-mdns-service-list** configuration for filtering mDNS service announcement and queries.
	In wireless network, the mDNS packets are consumed by the mDNS gateway and clients or device is deprived of learning this service. To share the service with the device and provide ease of configuration to the administrator, a list of few standard service types are shared by default on the wireless network. The list of such standard service types is termed as default service policy that comprises a set of service types.
	The table covers a sample service list in the default service policy.

Table 1: Default Name and mDNS Service Type

Default Name	mDNS Service Type
Apple TV	_airplay._tcp.local _raop._tcp.local
Apple HomeSharing	_home-sharing._tcp.local
Printer-IPPS	_ipps._tcp.local
Apple-airprint	_ipp._tcp.local _universal._sub._ipp._tcp.local
Google-chromecast	_googlecast._tcp.local _googlerpc._tcp.local _googlezone._tcp.local
Apple-remote-login	_sftp-ssh._tcp.local _ssh._tcp.local
Apple-screen-share	_rfb._tcp.local
Google-expeditions	_googexpeditions._tcp.local

	Command or Action	Purpose

		Default Name	mDNS Service Type
		Multifunction-printer	_fax-ipp._tcp.local
			_ipp._tcp.local
			_scanner._tcp.local
		Apple-windows-fileshare	_smb._tcp.local

		Note	• Location would be disabled on mDNS default service policy. • You cannot change the contents of the mDNS default service policy. However, you can create separate mDNS service policies and associate them under the wireless policy profile.

Step 4	**exit** **Example:** `Device(config-wireless-policy)# exit`	Returns to global configuration mode.

Enabling or Disabling mDNS Gateway for WLAN (GUI)

Procedure

Step 1	Choose **Configuration** > **Tags & Profiles** > **WLANs**.
Step 2	Click on the WLAN.
Step 3	In the **Advanced** tab, choose the mode in **mDNS Mode** drop-down list.
Step 4	Click **Update & Apply to Device**.

Enabling or Disabling mDNS Gateway for WLAN

Note   Bridging is the default behaviour. This means that the mDNS packets are always bridged.

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**wlan** *profile-name wlan-id ssid-name* **Example:** `Device(config)# wlan test 24 ssid1`	Specifies the WLAN name and ID. • *profile-name* is the WLAN name which can contain 32 alphanumeric characters • *wlan-id* is the wireless LAN identifier. The valid range is from 1 to 4096. • *ssid-name* is the SSID which can contain 32 alphanumeric characters. **Note** Global configuration must be in place for mDNS gateway to work.

Step 3	**mdns-sd-interface** {**gateway** | **drop**}
	
	Example:
	
	`Device(config-wlan)# mdns-sd gateway`
	
	`Device(config-wlan)# mdns-sd drop`
		Enables or disables mDNS gateway and bridge functions on WLAN.

Step 4	**exit** **Example:** `Device(config-wlan)# exit`	Returns to global configuration mode.
Step 5	**show wlan name** *wlan-name* | **show wlan all** **Example:** `Device# show wlan name test	show wlan all`	Verifies the status of mDNS on WLAN.
Step 6	**show wireless profile policy** **Example:** `Device# show wireless profile policy`	Verifies the service policy configured in WLAN.

mDNS Gateway with Guest Anchor Support and mDNS Bridging

When mDNS Gateway is enabled on both Anchor and Foreign controller, the mDNS gateway functionality is supported in guest anchor deployment where clients on guest LAN or WLAN with guest anchor enabled will be responded with any services or cache from export foreign controller itself. All advertisements received on Guest LAN or WLAN on export foreign are learnt on the export foreign itself. All queries received on guest LAN or WLAN are responded by the export foreign itself.

When mDNS Gateway is enabled on Anchor and Disabled on Foreign controller [Bridging Mode], the mDNS gateway functionality is supported in guest anchor deployment where clients on guest LAN or WLAN with guest anchor enabled will be responded with any services or cache from export Anchor even though the clients are connected on Foreign. All advertisements received on guest LAN or WLAN on export foreign is forwarded to Anchor and the cache is stored on the Anchor itself. All queries received on guest LAN or WLAN are responded by the export Anchor itself.

Note
- You must configure the guest-LAN to a wireless profile policy which is configured with the required mDNS service-policy.
- To configure non guest LAN mDNS gateway, see the mDNS Gateway chapter.

Configuring mDNS Gateway on Guest Anchor

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**guest-lan profile-name** *guest-lan-profile-name guest-lan-id* **Example:** `Device(config)# guest-lan profile-name g-lanpro 2`	Configures the guest LAN profile with a wired VLAN.
Step 3	**mdns-sd gateway** **Example:** `Device(config-guest-lan)# mdns-sd gateway`	Enables mDNS gateway on the guest LAN.

Configuring mDNS Gateway on Guest Foreign (Guest LAN)

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**guest-lan profile-name** *guest-lan-profile-name* *guest-lan-id* **wired-vlan** *vlan-id* **Example:** `Device(config)# guest-lan profile-name g-lanpro 2 wired-vlan 230`	Configures guest LAN profile with a wired VLAN. **Note** Configure the wired VLAN only for the Guest Foreign controller.
Step 3	**mdns-sd gateway** **Example:** `Device(config-guest-lan)# mdns-sd gateway`	Enables mDNS gateway on the guest LAN.
Step 4	**exit** **Example:** `Device(config-wireless-policy)# exit`	Returns to global configuration mode.

Configuring mDNS Gateway on Guest Anchor

Procedure

	Command or Action	**Purpose**
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**guest-wlan profile-name** *guest-lan-profile-name guest-wlan-id* **Example:** `Device(config)# guest-wlan profile-name g-lanpro 2`	Configures the guest WLAN profile with a wired VLAN.
Step 3	**mdns-sd gateway** **Example:** `Device(config-guest-wlan)# mdns-sd gateway`	Enables mDNS gateway on the guest WLAN.

Configuring mDNS Gateway on Guest Foreign (Guest WLAN)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**guest-wlan profile-name** *guest-lan-profile-name guest-wlan-id* **wired-vlan** *vlan-id* **Example:** `Device(config)# guest-wlan profile-name g-lanpro 2 wired-vlan 230`	Configures guest WLAN profile with a wired VLAN. **Note** Configure the wired VLAN only for the Guest Foreign controller.
Step 3	**mdns-sd gateway** **Example:** `Device(config-guest-wlan)# mdns-sd gateway`	Enables mDNS gateway on the guest WLAN.
Step 4	**exit** **Example:** `Device(config-wireless-policy)# exit`	Returns to global configuration mode.

Verifying mDNS Gateway Configurations

To verify the mDNS summary, use the following command:

```
Device# show mdns-sd summary
mDNS Gateway: Enabled
Active Query: Enabled
  Periodicity (in minutes): 30
Transport Type: IPv4
```

To verify the mDNS cache, use the following command:

```
Device# show mdns-sd cache
-------------------------------------------------------- PTR Records
------------------------------------
RECORD-NAME                    TTL    WLAN   CLIENT-MAC      RR-RECORD-DATA

---------------------------------------------------------------------------------------------------
_airplay._tcp.local            4500   30     07c5.a4f2.dc01  CUST1._airplay._tcp.local

_ipp._tcp.local                4500   30     04c5.a4f2.dc01  CUST3._ipp._tcp.local2

_ipp._tcp.local                4500   15     04c5.a4f2.dc01  CUST3._ipp._tcp.local4
```

```
_ipp._tcp.local                         4500    10      04c5.a4f2.dc01   CUST3._ipp._tcp.local6

_veer_custom._tcp.local                 4500    10      05c5.a4f2.dc01
CUST2._veer_custom._tcp.local8
```

To verify the mDNS cache from wired service provider, use the following command:

```
Device# show mdns-sd cache wired
------------------------------------------------------- PTR Records
-------------------------------------
RECORD-NAME                                  TTL     VLAN    CLIENT-MAC       RR-RECORD-DATA

--------------------------------------------------------------------------------------------------
_airplay._tcp.local                          4500    16      0866.98ec.97af
wiredapple._airplay._tcp.local
_raop._tcp.local                             4500    16      0866.98ec.97af
086698EC97AF@wiredapple._raop._tcp.local


------------------------------------------------------- SRV Records
-----------------------------------------
RECORD-NAME                                  TTL     VLAN    CLIENT-MAC       RR-RECORD-DATA

--------------------------------------------------------------------------------------------------
wiredapple._airplay._tcp.local               4500    16      0866.98ec.97af   0 0 7000
wiredapple.local
086698EC97AF@wiredapple._raop._tcp.local     4500    16      0866.98ec.97af   0 0 7000
wiredapple.local


------------------------------------------------------- A/AAAA Records
-----------------------------------------
RECORD-NAME                                  TTL     VLAN    CLIENT-MAC       RR-RECORD-DATA

--------------------------------------------------------------------------------------------------
wiredapple.local                             4500    16      0866.98ec.97af
2001:8:16:16:e5:c446:3218:7437


------------------------------------------------------- TXT Records
-----------------------------------------
RECORD-NAME                                  TTL     VLAN    CLIENT-MAC       RR-RECORD-DATA

--------------------------------------------------------------------------------------------------
wiredapple._airplay._tcp.local               4500    16      0866.98ec.97af
[343]'acl=0''deviceid=08:66:98:EC:97:AF''features=
086698EC97AF@wiredapple._raop._tcp.local     4500    16      0866.98ec.97af
[193]'cn=0,1,2,3''da=true''et=0,3,5''ft=0x5A7FFFF7
```

To verify the mdns-sd type PTR, use the following command:

```
Device# show mdns-sd cache type {PTR | SRV | A-AAA | TXT}
RECORD-NAME                                  TTL     WLAN    CLIENT-MAC
RR-Record-Data
-------------------------------------------------------------------------------------------
_custom1._tcp.local                          4500    2       c869.cda8.77d6
service_t1._custom1._tcp.local
_custom1._tcp.local                          4500    2       c869.cda8.77d6
vk11._custom1._tcp.local
_ipp._tcp.local                              4500    2       c869.cda8.77d6
service-4._ipp._tcp.local
```

To verify the mdns-sd cache for a client MAC, use the following command:

```
Device# show mdns-sd cache {ap-mac <ap-mac> | client-mac <client-mac> | glan-id <glan-id>
| mdns-ap <mac-address> | rlan-id <rlan-id> | wlan-id <wlan-id> | wired}
RECORD-NAME                                  TTL     WLAN    CLIENT-MAC
RR-Record-Data
-------------------------------------------------------------------------------------------
```

```
_custom1._tcp.local                                    4500    2      c869.cda8.77d6
service_t1._custom1._tcp.local
_custom1._tcp.local                                    4500    2      c869.cda8.77d6
vk11._custom1._tcp.local
_ipp._tcp.local                                        4500    2      c869.cda8.77d6
service-4._ipp._tcp.local


----------------------------------------------------------- SRV Records
-----------------------------------------------------------
RECORD-NAME                                            TTL     WLAN   CLIENT-MAC
RR-Record-Data
---------------------------------------------------------------------------------
service-4._ipp._tcp.local                             4500    2      c869.cda8.77d6   0 0 1212
 mDNS-Client1s-275.local
vk11._custom1._tcp.local                              4500    2      c869.cda8.77d6   0 0 987
 mDNS-Client1s-275.local
service_t1._custom1._tcp.local                        4500    2      c869.cda8.77d6   0 0 197
 mDNS-Client1s-275.local


----------------------------------------------------------- A/AAAA Records
-----------------------------------------------------------
RECORD-NAME                                            TTL     WLAN   CLIENT-MAC
RR-Record-Data
---------------------------------------------------------------------------------
mDNS-Client1s-275.local                               4500    2      c869.cda8.77d6   120.1.1.33


----------------------------------------------------------- TXT Records
-----------------------------------------------------------
RECORD-NAME                                            TTL     WLAN   CLIENT-MAC
RR-Record-Data
---------------------------------------------------------------------------------
service-4._ipp._tcp.local                             4500    2      c869.cda8.77d6   'CLient1'

vk11._custom1._tcp.local                              4500    2      c869.cda8.77d6
'txtvers=11'
service_t1._custom1._tcp.local                        4500    2      c869.cda8.77d6
'txtvers=12'
```

To verify the mdns-sd cache with respect to the RLAN ID, use the following command:

```
Device# show mdns-sd cache rlan-id 1 detail

Name: _printer._tcp.local

  Type: PTR
  TTL: 4500
  RLAN: 1
  RLAN Name: rlan_test_1
  VLAN: 141
  Client MAC: 000e.c688.3942
  AP Ethernet MAC: 0042.5ab6.0ef0
  Remaining-Time: 4485
  Site-Tag: default-site-tag
  mDNS Service Policy: mdnsTV6
  Overriding mDNS Service Policy: NO
  UPN-Status: Disabled
  Rdata: printer._printer._tcp.local

Name: lab-47-187.local
  Type: A/AAAA
  TTL: 4500
  RLAN: 1
  RLAN Name: rlan_test_1
  VLAN: 141
```

```
Client MAC: 000e.c688.3942
AP Ethernet MAC: 0042.5ab6.0ef0
Remaining-Time: 4485
Site-Tag: default-site-tag
mDNS Service Policy: mdnsTV6
Overriding mDNS Service Policy: NO
UPN-Status: Disabled
Rdata: 10.15.141.124
```

To verify the mdns-sd cache with respect to mDNS-AP, use the following command:

```
Device# show mdns-sd cache mdns-ap 706b.b97d.b060 detail
Name: _printer._tcp.local

  Type: PTR
  TTL: 4500
  VLAN: 145
  Client MAC: 0050.b626.5bfa
  mDNS AP Radio MAC: 706b.b97d.b060
  mDNS AP Ethernet MAC: 706b.b97c.5208
  Remaining-Time: 4480
  mDNS Service Policy: mdnsTV
  Rdata: printer._printer._tcp.local

Name: Client-46-153.local
  Type: A/AAAA
  TTL: 4500
  VLAN: 145
  Client MAC: 0050.b626.5bfa
  mDNS AP Radio MAC: 706b.b97d.b060
  mDNS AP Ethernet MAC: 706b.b97c.5208
  Remaining-Time: 4480
  mDNS Service Policy: mdnsTV
  Rdata: 10.15.145.103
```

To verify the mdns-sd cache in detail, use the following command:

```
Device# show mdns-sd cache detail

Name: _custom1._tcp.local
  Type: PTR
  TTL: 4500
  WLAN: 2
  WLAN Name: mdns120
  VLAN: 120
  Client MAC: c869.cda8.77d6
  AP Ethernet MAC: 7069.5ab8.33d0
  Expiry-Time: 09/09/18 21:50:47
  Site-Tag: default-site-tag
  Rdata: service_t1._custom1._tcp.local
```

To verify the mdns-sd cache statistics, use the following command:

```
Device# show mdns-sd cache statistics

mDNS Cache Stats

Total number of Services: 4191
```

To verify the mdns-sd statistics, use the following command:

```
Device# show mdns-sd statistics

-----------------------------------------------------
```

```
Consolidated mDNS Packet Statistics
--------------------------------------------------------
mDNS stats last reset time: 03/11/19 04:17:35
mDNS packets sent: 61045
  IPv4 sent: 30790
    IPv4 advertisements sent: 234
    IPv4 queries sent: 30556
  IPv6 sent: 30255
    IPv6 advertisements sent: 17
    IPv6 queries sent: 30238
  Multicast sent: 57558
    IPv4 sent: 28938
    IPv6 sent: 28620
mDNS packets received: 72796
  advertisements received: 13604
  queries received: 59192
  IPv4 received: 40600
    IPv4 advertisements received: 6542
    IPv4 queries received: 34058
  IPv6 received: 32196
    IPv6 advertisements received: 7062
    IPv6 queries received: 25134
mDNS packets dropped: 87


--------------------------------------------------------
Wired mDNS Packet Statistics
--------------------------------------------------------
mDNS stats last reset time: 03/11/19 04:17:35
mDNS packets sent: 61033
  IPv4 sent: 30778
    IPv4 advertisements sent: 222
    IPv4 queries sent: 30556
  IPv6 sent: 30255
    IPv6 advertisements sent: 17
    IPv6 queries sent: 30238
  Multicast sent: 57558
    IPv4 sent: 28938
    IPv6 sent: 28620
mDNS packets received: 52623
  advertisements received: 1247
  queries received: 51376
  IPv4 received: 32276
    IPv4 advertisements received: 727
    IPv4 queries received: 31549
  IPv6 received: 20347
    IPv6 advertisements received: 520
    IPv6 queries received: 19827
mDNS packets dropped: 63


--------------------------------------------------------
mDNS Packet Statistics, for WLAN: 2
--------------------------------------------------------
mDNS stats last reset time: 03/11/19 04:17:35
mDNS packets sent: 12
  IPv4 sent: 12
    IPv4 advertisements sent: 12
    IPv4 queries sent: 0
  IPv6 sent: 0
    IPv6 advertisements sent: 0
    IPv6 queries sent: 0
  Multicast sent: 0
    IPv4 sent: 0
    IPv6 sent: 0
mDNS packets received: 20173
```

```
  advertisements received: 12357
  queries received: 7816
  IPv4 received: 8324
    IPv4 advertisements received: 5815
    IPv4 queries received: 2509
  IPv6 received: 11849
    IPv6 advertisements received: 6542
    IPv6 queries received: 5307
mDNS packets dropped: 24
```

To verify the default service list details, use the following command:

```
Device# show mdns-sd default-service-list
-------------------------------------------
mDNS Default Service List
-------------------------------------------

Service Definition: apple-tv
Service Names: _airplay._tcp.local
_raop._tcp.local

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

Service Definition: printer-ipps
Service Names: _ipps._tcp.local

Service Definition: apple-airprint
Service Names: _ipp._tcp.local
_universal._sub._ipp._tcp.local

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

Service Definition: apple-remote-login
Service Names: _sftp-ssh._tcp.local
_ssh._tcp.local

Service Definition: apple-screen-share
Service Names: _rfb._tcp.local

Service Definition: google-expeditions
Service Names: _googexpeditions._tcp.local

Service Definition: multifunction-printer
Service Names: _fax-ipp._tcp.local
_ipp._tcp.local
_scanner._tcp.local

Service Definition: apple-windows-fileshare
Service Names: _smb._tcp.local
```

To verify the primary service list details, use the following command:

```
Device# show mdns-sd master-service-list


-------------------------------------------
        mDNS Master Service List
-------------------------------------------

Service Definition: fax
Service Names: _fax-ipp._tcp.local
```

```
Service Definition: roku
Service Names: _rsp._tcp.local

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

Service Definition: scanner
Service Names: _scanner._tcp.local

Service Definition: spotify
Service Names: _spotify-connect._tcp.local

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

Service Definition: airserver
Service Names: _airplay._tcp.local
               _airserver._tcp.local

.
.
.

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

To verify the mdns-sd service statistics on the controller, use the following command:

```
Device# show mdns-sd service statistics

Service Name                                                      Service Count
--------------------------------------------------------------------------
_atc._tcp.local                                                   137
_hap._tcp.local                                                   149
_ipp._tcp.local                                                   149
_rfb._tcp.local                                                   141
_smb._tcp.local                                                   133
_ssh._tcp.local                                                   142
_daap._tcp.local                                                  149
_dpap._tcp.local                                                  149
_eppc._tcp.local                                                  138
_adisk._tcp.local                                                 149
```

To verify the mDNS-AP configured on the controller and VLAN(s) associated with it, use the following command:

```
Device# show mdns-sd ap

Number of mDNS APs................................... 1

AP Name    Ethernet MAC    Number of Vlans    Vlanidentifiers
------------------------------------------------------------------------------------------------
AP3600-1   7069.5ab8.33d0           1         300
```

Further Debug

To debug mDNS further, use the following procedure:

1. Run this command at the controller:

   ```
   set platform software trace wncd <0-7> chassis active R0 mdns debug
   ```

2. Reproduce the issue.

3. Run this command to gather the traces enabled:

```
show wireless loadbalance ap affinity wncd 0

AP MAC    Discovery Timestamp  Join Timestamp             Tag     Vlanidentifiers
-------------------------------------------------------------------------------
0cd0.f894.0600        06/30/21 12:39:48    06/30/21 12:40:021 default-site-tag     300
```