Configuring the Service Discovery Gateway

- Restrictions for Configuring the Service Discovery Gateway, on page 1
- Information about the Service Discovery Gateway and mDNS, on page 1
- How to Configure the Service Discovery Gateway, on page 4
- Monitoring Service Discovery Gateway, on page 8
- Configuration Examples, on page 8
- Where to Go Next for Configuring Services Discovery Gateway, on page 10
- Additional References for Service Discovery Gateway, on page 10
- Feature History and Information for Service Discovery Gateway, on page 10

Restrictions for Configuring the Service Discovery Gateway

The following are restrictions for configuring the Service Discovery Gateway:

- The Service Discovery Gateway does not support topologies with multiple hops. All network segments must be connected directly to it. The Service Discovery Gateway can learn services from all connected segments to build its cache and respond to requests acting as a proxy.
- The use of third-party mDNS servers or applications are not supported with this feature.

Information about the Service Discovery Gateway and mDNS

mDNS

mDNS was defined to achieve zero configuration, with zero configuration being defined as providing the following features:

- Addressing—Allocating IP addresses to hosts
- Naming—Using names to refer to hosts instead of IP addresses
- Service discovery—Finding services automatically on the network

With mDNS, network users no longer have to assign IP addresses, assign host names, or type in names to access services on the network. Users only need to ask to see what network services are available, and choose from a list.

With mDNS, *addressing* is accomplished through the use of DHCP/DHCPv6 or IPv4 and IPv6 Link Local scoped addresses. The benefit of zero-configuration occurs when no infrastructure services such as DHCP or DNS are present and self-assigned link-local addressing can be used. The client can then select a random IPv4 address in the link-local range (169.254.0.0/24) or use its IPv6 link-local address (FE80::/10) for communication.

With mDNS, *naming* (name-to-address translation on a local network using mDNS) queries are sent over the local network using link-local scoped IP multicast. Because these DNS queries are sent to a multicast address (IPv4 address 224.0.0.251 or IPv6 address FF02::FB), no single DNS server with global knowledge is required to answer the queries. When a service or device sees a query for any service it is aware of, it provides a DNS response with the information from its cache.

With mDNS, *service discovery* is accomplished by browsing. An mDNS query is sent out for a given service type and domain, and any device that is aware of matching services replies with service information. The result is a list of available services for the user to choose from.

The mDNS protocol (mDNS-RFC), together with DNS Service Discovery (DNS-SD-RFC) achieves the zero-configuration addressing, naming, and service discovery.

mDNS-SD

Multicast DNS Service Discovery (mDNS-SD) uses DNS protocol semantics and multicast over well-known multicast addresses to achieve zero configuration service discovery. DNS packets are sent to and received on port 5353 using a multicast address of 224.0.0.251 and its IPv6 equivalent FF02::FB.

Because mDNS uses a link-local multicast address, its scope is limited to a single physical or logical LAN. If the networking reach needs to be extended to a distributed campus or to a wide-area environment consisting of many different networking technologies, mDNS gateway is implemented. An mDNS gateway provides a transport for mDNS packets across Layer 3 boundaries by filtering, caching, and redistributing services from one Layer 3 domain to another.

Service Discovery Gateway

The Service Discovery Gateway feature enables multicast Domain Name System (mDNS) to operate across Layer 3 boundaries (different subnets). An mDNS gateway provides transport for service discovery across Layer 3 boundaries by filtering, caching, and redistributing services from one Layer 3 domain (subnet) to another. Prior to implementation of this feature, mDNS was limited in scope to within a subnet because of the use of link-local scoped multicast addresses. This feature enhances Bring Your Own Device (BYOD).

mDNS Gateway and Subnets

You need to enable an mDNS gateway for service discovery to operate across subnets. You can enable mDNS gateway for a device or for an interface.

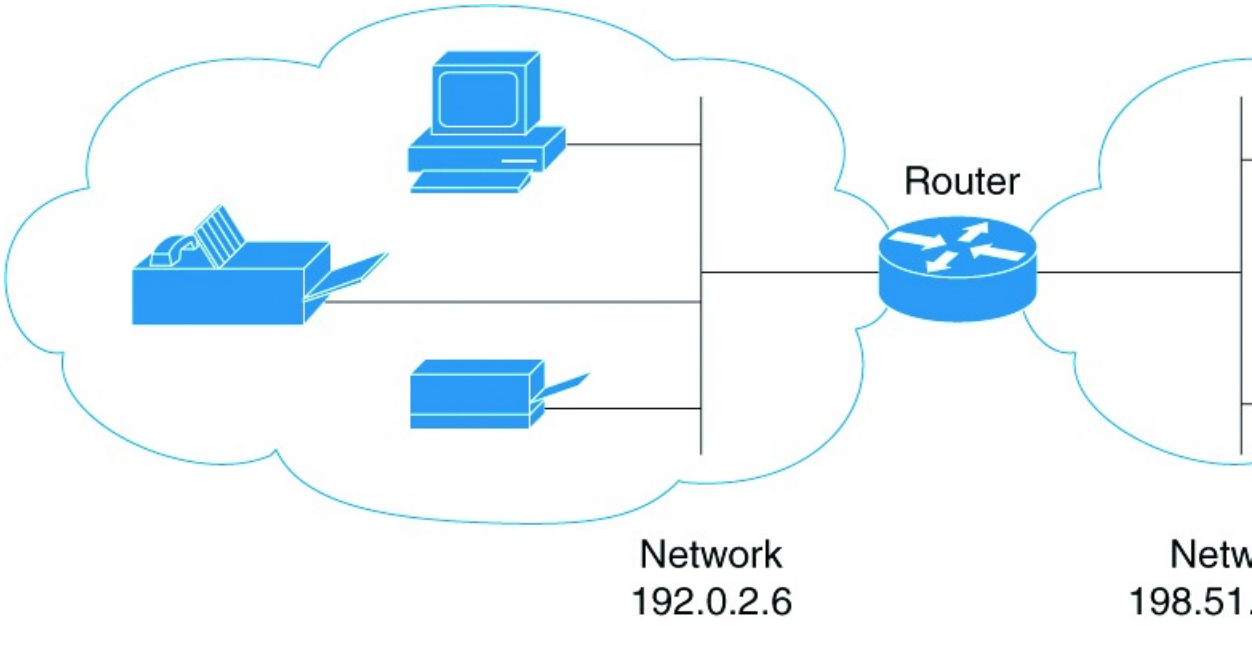
Note You need to configure service routing globally before configuring at the interface level.

After the device or interface is enabled, you can redistribute service discovery information across subnets. You can create service policies and apply filters on either incoming service discovery information (called IN-bound filtering) or outgoing service discovery information (called OUT-bound filtering).

Note If redistribution is enabled globally, global configuration is given higher priority than interface configuration.

Figure 1: Sample Networking Scenario

For example, if the mDNS gateway functionality is enabled on the router in this figure, then service information can be sent from one subnet to another and vice-versa. For example, the printer and fax service information being advertised in the network with IP address 192.0.2.6 are redistributed to the network with IP address 198.51.100.4. The printer and fax service information in the network with IP address 192.0.2.6 is learned by mDNS-enabled hosts and devices in the other network.



Filtering

After configuring the mDNS gateway and subnets, you can filter services that you want to redistribute. While creating a service list, the **permit** or **deny** command options are used:

- The **permit** command option allows you to permit or transport specific service list information.
- The **deny** option allows you to deny service list information that is available to be transported to other subnets.

You need to include a sequence number when using the **permit** or **deny** command option. The same service list name can be associated with multiple sequence numbers and each sequence number will be mapped to a rule.

Note If no filters are configured, then the default action is to deny service list information to be transported through the device or interface.

Query is another option provided when creating service lists. You can create queries using a service list. If you want to browse for a service, then active queries can be used. This function is helpful to keep the records refreshed in the cache.

Note Active queries can only be used globally and cannot be used at the interface level.

A service end-point (such as a printer or fax) sends unsolicited announcements when a service starts up. After that, it sends unsolicited announcements whenever a network change event occurs (such as an interface coming up or going down). The device always respond to queries.

After creating a service list and using the **permit** or **deny** command options, you can filter using match statements (commands) based on *service-instance*, *service-type*, or *message-type* (announcement or query).

How to Configure the Service Discovery Gateway

Configuring the Service List

This procedure describes how to create a service list, apply a filter for the service list, and configure parameters for the service list name.

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	**service-list mdns-sd** *service-list-name* {**deny** *sequence-number* | **permit** *sequence-number* | **query**}
	Example:
	`Device(config)# service-list mdns-sd sl1 permit 3`
	`Device(config)# service-list mdns-sd sl4`
		Enters mDNS service discovery service list mode. In this mode, you can:
		• Create a service list and apply a filter on the service list according to the **permit** or **deny** option applied to the sequence number.
		• Create a service list and associate a query for the service list name if the **query** option is used.

	Command or Action	Purpose
	query	**Note** The sequence number sets the priority of the rule. A rule with a lower sequence number is selected first and the service announcement or query is allowed or denied accordingly. You define the sequence number as per your network requirements.
Step 4	**match message-type** {**announcement** | **any** | **query**} **Example:** `Device(config-mdns-sd-sl)# match message-type announcement`	(Optional) Sets the message type to match. You can match the following message types: • announcement • any • query These commands configure the parameters for the service list name that is created in step 2. If the **match message-type** is an announcement, then the service list rule only allows service advertisements or announcements for the device. If the **match message-type** is a query, then only a query from the client for a certain service in the network is allowed. Multiple service maps of the same name with different sequence numbers can be created and the evaluation of the filters will be ordered on the sequence number. Service lists are an ordered sequence of individual statements, each one has a permit or deny result. Evaluation of service list consists of a list scan, in a predetermined order, and an evaluation of the criteria of each statement that matches. A list scan is stopped once the first statement match is found and an action permit/deny associated with the statement match is performed. The default action after scanning through the entire list is to deny. **Note** You cannot use the **match** command if you have used the **query** option in the previous step. The **match** command can be used only for the **permit** or **deny** option.
Step 5	**match service-instance** { *LINE* } **Example:**	(Optional) Sets the service instance to match. This command configures the parameters for the service list name that is created in step 2.

	Command or Action	Purpose
	`Device(config-mdns-sd-sl)## match service-instance servInst 1`	**Note** You cannot use the **match** command if you have used the **query** option in the previous step. The **match** command can be used only for the **permit** or **deny** option.
Step 6	**match service-type** {*LINE* } **Example:** `Device(config-mdns-sd-sl)# match service-type _ipp._tcp`	(Optional) Sets the value of the mDNS service type string to match. This command configures the parameters for the service list name that is created in step 2. **Note** You cannot use the **match** command if you have used the **query** option in the previous step. The **match** command can be used only for the **permit** or **deny** option.
Step 7	**end** **Example:** `Device(config-mdns-sd-sl)# end`	Returns to privileged EXEC mode.

What to do next

Proceed to enable the mDNS gateway and redistribution of services.

Enabling mDNS Gateway and Redistributing Services

After enabling mDNS gateway for a device, you can apply filters (apply IN-bound filtering or OUT-bound filtering) and active queries by using **service-policy** and **service-policy-query** commands, respectively. You can redistribute services and service announcements using the **redistribute mdns-sd** command, and set some part of the system memory for cache using the **cache-memory-max** command.

Note By default, mDNS gateway is disabled on all interfaces.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**service-routing mdns-sd** **Example:** `Device (config)# service-routing mdns-sd`	Enables mDNS gateway functionality for a device and enters multicast DNS configuration (config-mdns) mode. **Note** This command enables the mDNS function globally. **Note** Enter the **service-routing mdns-sd source-interface** *if-name* command in either global-config or interface-config mode, to specify an alternate source interface for outgoing mDNS packets, so its IP address can be used when there is none configured on the outgoing interface.
Step 4	**service-policy** *service-policy-name* {**IN** | **OUT**} **Example:** `Device (config-mdns)# service-policy serv-pol1 IN`	(Optional) For a service list, applies a filter on incoming service discovery information (IN-bound filtering) or outgoing service discovery information (OUT-bound filtering).
Step 5	**redistribute mdns-sd** **Example:** `Device (config-mdns)# redistribute mdns-sd`	(Optional) Redistributes services or service announcements across subnets. **Note** If redistribution is enabled globally, global configuration is given higher priority than interface configuration.
Step 6	**cache-memory-max** *cache-config-percentage* **Example:** `Device (config-mdns)# cache-memory-max 20`	(Optional) Sets some part of the system memory (in percentage) for cache. **Note** By default, 10 percent of the system memory is set aside for cache. You can override the default value by using this command.
Step 7	**service-policy-query** *service-list-query-name service-list-query-periodicity* **Example:**	(Optional) Configures service list-query periodicity.

	Command or Action	Purpose
	`Device (config-mdns)#` **`service-policy-query sl-query1 100`**	
Step 8	**exit** **Example:** `Device (config-mdns)#`**`exit`**	(Optional) Returns to global configuration mode.
Step 9	**end** **Example:** `Device(config)#` **`end`**	Returns to privileged EXEC mode.

Monitoring Service Discovery Gateway

Table 1: Monitoring Service Discovery Gateway

Command	Purpose
show mdns requests [**detail** | **name** *record-name*| **type** *record-type* [**name** *record-name*]]	This command displays information for outstanding mDNS requests, including record name and record type information.
show mdns cache [**interface** *type number* | **name** *record-name* [**type** *record-type*]| **type** *record-type*]	This command displays mDNS cache information.
show mdns statistics {**all** | **service-list** *list-name* | **service-policy** {**all** | **interface** *type number* }}	This command displays mDNS statistics.

Configuration Examples

Example: Specify Alternative Source Interface for Outgoing mDNS Packets

The following example displays how to specify an alternate source interface for outgoing mDNS packets, so its IP address can be used when there is none configured on the outgoing interface.

```
Device(config)# service-routing mdns-sd
Device(config-mdns)# source-interface if-name
```

Example: Redistribute Service Announcements

The following example displays how to redistribute service announcements received on one interface over all the interfaces or over a specific interface.

```
Device(config)# service-routing mdns-sd
Device(config-mdns)# Redistribute mdns-sd if-name
```

Example: Creating a Service-List, Applying a Filter and Configuring Parameters

The following example shows the creation of a service-list sl1. The **permit** command option is being applied on sequence number 3 and all services with message-type announcement are filtered and available for transport across various subnets associated with the device.

```
Device# configure terminal
Device(config)# service-list mdns-sd sl1 permit 3
Device(config-mdns-sd-sl)#match message-type announcement
Device(config-mdns)# exit
```

Example: Enabling mDNS Gateway and Redistributing Services

The following example shows how to enable an mDNS gateway for a device and enable redistribution of services across subnets. IN-bound filtering is applied on the service-list serv-pol1. Twenty percent of system memory is made available for cache and service-list-query periodicity is configured at 100 seconds.

```
Device# configure terminal
Device# service-routing mdns-sd
Device(config-mdns)# service-policy serv-pol1 IN
Device(config-mdns)# redistribute mdns-sd
Device(config-mdns)# cache-memory-max 20
Device(config-mdns)# service-policy-query sl-query1 100
Device(config-mdns)# exit
```

Example: Global mDNS Configuration

The following example displays how to globally configure mDNS.

```
Device# configure terminal
Device(config)# service-list mdns-sd mypermit-all permit 10
Device(config-mdns-sd-s1)# exit
Device(config)# service-list mdns-sd querier query
Device(config-mdns-sd-s1)# service-type _dns._udp
Device(config-mdns-sd-s1)# end
Device# configure terminal
Device(config)# service-routing mdns-sd
Device(config-mdns)# service-policy mypermit-all IN
Device(config-mdns)# service-policy mypermit-all OUT
```

Example: Interface mDNS Configuration

The following example displays how to configure mDNS for an interface.

```
Device(config)#interface Vlan136
Device(config-if)# description *** Mgmt VLAN ***
Device(config-if)# ip address 9.7.136.10 255.255.255.0
Device(config-if)# ip helper-address 9.1.0.100
Device(config-if)# service-routing mdns-sd
Device(config-if-mdns-sd)#  service-policy mypermit-all IN
Device(config-if-mdns-sd)# service-policy mypermit-all OUT
Device(config-if-mdns-sd)#  service-policy-query querier 60
```

Where to Go Next for Configuring Services Discovery Gateway

You can configure the following:

Additional References for Service Discovery Gateway

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the IP Multicast Routing Commands section of the *Command Reference (Catalyst 9500 Series Switches)*

Standards and RFCs

Standard/RFC	Title
RFC 6763	*DNS-Based Service Discovery*
Multicast DNS Internet-Draft	Multicast

Feature History and Information for Service Discovery Gateway

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Service Discovery Gateway	The Service Discovery Gateway feature enables multicast Domain Name System (mDNS) to operate across Layer 3 boundaries (different subnets). Support for this feature was introduced only on the C9500-12Q, C9500-16X, C9500-24Q, C9500-40X models of the Cisco Catalyst 9500 Series Switches.
Cisco IOS XE Fuji 16.8.1a	Service Discovery Gateway	Support for this feature was introduced only on the C9500-32C, C9500-32QC, C9500-48Y4C, and C9500-24Y4C models of the Cisco Catalyst 9500 Series Switches.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com