Implementing the Dynamic Host Configuration Protocol

This module describes the concepts and tasks you will use to configure Dynamic Host Configuration Protocol (DHCP).

Note For a complete description of the DHCP commands listed in this module, refer to the *Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference* publication.

Feature History for Implementing the Dynamic Host Configuration Protocol

Release	Modification
Release 3.7.2	This feature was introduced .

Prerequisites for Configuring DHCP Relay Agent

The following prerequisites are required to configure a DHCP relay agent:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- A configured and running DHCP client and DHCP server
- Connectivity between the relay agent and DHCP server

Information About DHCP Relay Agent

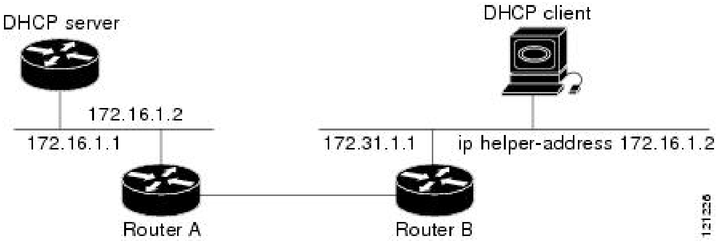
A DHCP relay agent is a host that forwards DHCP packets between clients and servers that do not reside on a shared physical subnet. Relay agent forwarding is distinct from the normal forwarding of an IP router where IP datagrams are switched between networks transparently.

DHCP clients use User Datagram Protocol (UDP) broadcasts to send DHCPDISCOVER messages when they lack information about the network to which they belong.

If a client is on a network segment that does not include a server, a relay agent is needed on that network segment to ensure that DHCP packets reach the servers on another network segment. UDP broadcast packets are not forwarded, because most routers are not configured to forward broadcast traffic. You can configure a DHCP relay agent to forward DHCP packets to a remote server by configuring a DHCP relay profile and configure one or more helper addresses in it. You can assign the profile to an interface or a VRF.

Figure 1: Forwarding UDP Broadcasts to a DHCP Server Using a Helper Address, on page 2 demonstrates the process. The DHCP client broadcasts a request for an IP address and additional configuration parameters on its local LAN. Acting as a DHCP relay agent, Router B picks up the broadcast, changes the destination address to the DHCP server's address and sends the message out on another interface. The relay agent inserts the IP address of the interface, on which the DHCP client's packets are received, into the gateway address (giaddr) field of the DHCP packet, which enables the DHCP server to determine which subnet should receive the offer and identify the appropriate IP address range. The relay agent unicasts the messages to the server address, in this case 172.16.1.2 (which is specified by the helper address in the relay profile).

Figure 1: Forwarding UDP Broadcasts to a DHCP Server Using a Helper Address



Limitations for DHCPv6 Relay Feature

These are the limitations for implementing DHCPv6 relay feature:

- The multicast addresses are not supported. The **helper-address** command in DHCPv6 relay profile submode will only support global unicast IPv6 address as the helper address.
- Only one relay is supported between client and server with an exception of Lightweight DHCPv6 Relay Agent (LRDA) being present on the access side. That is, the Layer 3 relay packets are not supported.
- Only interface-id and remote-id DHCPv6 option code are added by a relay agent while forwarding the packet to a DHCPv6 server.

Note Configuring DHCPv6 option code is not supported in DHCPv6 relay profile submode.

Secure ARP

In standalone DHCP sessions, the DHCP server adds an ARP entry when it assigns an IP address to a client. However, in IP subscriber sessions, DHCP server does not add an ARP entry. Although ARP establishes correspondences between network addresses, an untrusted device can spoof IP an address not assigned to it posing a security threat for IP subscriber sessions. You can enable the secure ARP feature and allow DHCP to add an ARP cache entry when DHCP assigns an IP address to a client. Secure ARP is disabled by default.

How to Configure and Enable DHCP Relay Agent

This section contains the following tasks:

Configuring and Enabling DHCP Relay Agent with DHCP MAC Address Verification

This section discusses how to configure and enable DHCP Relay Agent with DHCP MAC address verification.

Configuration Example

```
Router# configure

Router(config)# dhcp ipv4
/* Configures DHCP for IPv4 and enters the DHCPv4 configuration submode. */

Router(config-dhcpv4)# profile client relay
/* Enables DHCP relay profile */

Router(config-dhcpv4)# client-mac-mismatch action drop
/* Enables MAC address verification. If MAC address in the DHCPv4 protocol header does not
 match the L2 header source MAC address in the DHCPv4 relay profile,
 the frame is dropped  */

Router(config-dhcpv4-relay-profile)# relay information option
/* Inserts the DHCP relay agent information option (option-82 field) in forwarded
BOOTREQUEST messages to a DHCP server. */

Router(config-dhcpv4-relay-profile)# relay information check
/* (Optional) Configures DHCP to check the validity of the relay agent information
option in forwarded BOOTREPLY messages. */

Router(config-dhcpv4-relay-profile)# relay information policy drop
/* (Optional) Configures the reforwarding policy for a DHCP relay agent;
that is, whether the relay agent will drop or keep (using the 'keep' keyword)
 the relay information. */

Router(config-dhcpv4-relay-profile)# relay information option allow-untrusted
/* (Optional) Configures the DHCP IPv4 Relay not to discard BOOTREQUEST packets that have
an existing
relay information option and the giaddr set to zero. */

Router(config-dhcpv4-relay-profile)# giaddr policy drop
/* Drops the packet that has an existing nonzero giaddr value. Use the 'replace' keyword
 to replace the existing giaddr value with a value that it generates (the default behavior).
  */

Router(config-dhcpv4-relay-profile)# helper-address vrf vrf1 10.1.1.1
/* Forwards UDP broadcasts, including DHCP. */

Router(config-dhcpv4-relay-profile)# commit

Router(config-dhcpv4-relay-profile)# exit
Router(config-dhcpv4)# vrf vrf1 relay profile client
Router(config-dhcpv4)# commit
/* Configures DHCP Relay on a VRF and commits the entire configuration. */
```

Running Configuration

Confirm your configuration.

```
Router# show run
Thu May 11 09:00:57.839 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Thu May 11 09:00:54 2017 by annseque
!
dhcp ipv4
vrf vrf1 relay profile client
profile client relay
client-mac-match action drop
helper-address vrf vrf1 10.1.1.1
giaddr policy drop
relay information check
relay information option
relay information policy drop
relay information option allow-untrusted
!
!
```

DHCP MAC Address Verification

Use the following show command to check if DHCP MAC address is being verified on the router.

```
Router# show dhcp ipv4 relay statistics raw all
packet_drop_mac_mismatch                      :         0
```

The output validates that the DHCP MAC address of the packets is verified.

Configuring the DHCPv6 (Stateless) Relay Agent

Perform this task to specify a destination address to which client messages are forwarded and to enable Dynamic Host Configuration Protocol (DHCP) for IPv6 relay service on the interface.

Configuration Example

To configure the DHCPv6 (stateless) relay agent, you must complete the following configurations:

1. Enable the DHCP IPv6 configuration mode.
2. Configure the DHCPv6 relay profile.
3. Configure helper addresses.
4. Specify the interface for the relay profile.

Configuration

```
/* Enter the global configuration mode, and then enter the DHCP IPv6 configuration mode */
Router# configure terminal
Router(config)# dhcp ipv6
Router(config-dhcpv6)# profile test relay
Router(config-dhcpv6-relay-profile)# helper-address vrf default 2001:1::1
Router(config-dhcpv6-relay-profile)# !
Router(config-dhcpv6-relay-profile)# interface TenGigE0/0/0/0 relay profile test
Router(config-dhcpv6)# !
```

Enabling DHCP Relay Agent on an Interface

This task describes how to enable the Cisco IOS XR DHCP relay agent on an interface.

Note On Cisco IOS XR software, the DHCP relay agent is disabled by default.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **interface type name relay profile** *profile-name*
4. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters DHCP IPv4 configuration submode.

Step 3	**interface type name relay profile** *profile-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4)# interface gigabitethernet 0/0/0`
	`/0 relay profile client`
		Attaches a relay profile to an interface.

Step 4	**commit**	

Enabling DHCPv6 Relay Agent on an Interface

This task describes how to enable the DHCPv6 relay agent on an interface.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv6**
3. **interface** *type interface-instance* **relay profile** *profile-name*
4. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**dhcp ipv6** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	Configures DHCP for IPv6 and enters the DHCPv6 configuration submode.

	Command or Action	Purpose
Step 3	**interface** *type interface-instance* **relay profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# interface gigabitethernet 0/0/0/0 relay profile client`	Attaches a relay profile to an interface.
Step 4	**commit**	

Enabling DHCPv6 Relay Agent on an Interface: Example

```
 configure
 dhcp ipv6
 interface gigabitethernet 0/0/0/0 relay profile client
 !
end
```

Disabling DHCP Relay on an Interface

This task describes how to disable the DHCP relay on an interface by assigning the none profile to the interface.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **interface** *type name* **none**
4. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters DHCP IPv4 configuration submode.
Step 3	**interface** *type name* **none** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# interface gigabitethernet 0/1/4/1 none`	Disables the DHCP relay on the interface.

	Command or Action	Purpose
Step 4	**commit**	

Enabling DHCP Relay on a VRF

This task describes how to enable DHCP relay on a VRF.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **vrf** *vrf-name* **relay profile** *profile-name*
4. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters DHCP IPv4 configuration submode.

Step 3	**vrf** *vrf-name* **relay profile** *profile-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`
	`vrf default relay profile client`
		Enables DHCP relay on a VRF.

Step 4	**commit**	

Configuring the Relay Agent Information Feature

This task describes how to configure the DHCP relay agent information option processing capabilities.

A DHCP relay agent may receive a message from another DHCP relay agent that already contains relay information. By default, the relay information from the previous relay agent is replaced (using the replace option).

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **relay**
4. **relay information option**

5. **relay information check**
6. **relay information policy** {**drop** | **keep**}
7. **relay information option allow-untrusted**
8. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters DHCP IPv4 configuration submode .
Step 3	**profile** *profile-name* **relay** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile client relay`	Enters DHCP IPv4 profile relay submode .
Step 4	**relay information option** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# relay information option`	Enables the system to insert the DHCP relay agent information option (option-82 field) in forwarded BOOTREQUEST messages to a DHCP server. • This option is injected by the relay agent while forwarding client-originated DHCP packets to the server. Servers recognizing this option can use the information to implement IP address or other parameter assignment policies. When replying, the DHCP server echoes the option back to the relay agent. The relay agent removes the option before forwarding the reply to the client. • The relay agent information is organized as a single DHCP option that contains one or more suboptions. These options contain the information known by the relay agent. The supported suboptions are: • Remote ID • Circuit ID **Note** This function is disabled by default. The port field of the default circuit-ID denotes the configured bundle-ID of the bundle. If circuit IDs require

	Command or Action	Purpose
		that bundles be unique, and because the port field is 8 bits, the low-order 8 bits of configured bundle IDs must be unique. To achieve this, configure bundle-IDs within the range from 0 to 255.
Step 5	**relay information check** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# relay information check`	(Optional) Configures DHCP to check the validity of the relay agent information option in forwarded BOOTREPLY messages. If an invalid message is received, the relay agent drops the message. If a valid message is received, the relay agent removes the relay agent information option field and forwards the packet. • By default, DHCP does not check the validity of the relay agent information option field in DHCP reply packets, received from the DHCP server. **Note** Use the **relay information check** command to reenable this functionality if the functionality has been disabled.

Step 6	**relay information policy** {**drop** | **keep**}
	Example:
	`RP/0/RSP0/CPU0:router(config)# dhcp relay information policy drop`
		(Optional) Configures the reforwarding policy for a DHCP relay agent; that is, whether the relay agent will drop or keep the relay information.
		By default, the DHCP relay agent replaces the relay information option.

Step 7	**relay information option allow-untrusted** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# relay information option allow-untrusted`	(Optional) Configures the DHCP IPv4 Relay not to discard BOOTREQUEST packets that have an existing relay information option and the giaddr set to zero.
Step 8	**commit**	

Configuring Relay Agent Giaddr Policy

This task describes how to configure the DHCP relay agent's processing capabilities for received BOOTREQUEST packets that already contain a nonzero giaddr attribute.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile relay**
4. **giaddr policy** {**replace** | **drop**}
5. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enables the DHCP IPv4 configuration submode.
Step 3	**profile relay** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile client relay`	Enables profile relay submode.

Step 4	**giaddr policy** {**replace** | **drop**}
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# giaddr policy drop`
		Specifies the giaddr policy.
		• replace—Replaces the existing giaddr value with a value that it generates.
		• drop—Drops the packet that has an existing nonzero giaddr value.
		By default, the DHCP relay agent keeps the existing giaddr value.

Step 5	**commit**	

Configuring a DHCPv4 Relay Profile with Multiple Helper Addresses

You can configure up to 16 helper addresses for a DHCPv4 relay profile, as shown in the following example.

1. Enter the DHCPv4 configuration mode.

   ```
   RP/0/RSP0/CPU0:router(config)# dhcp ipv4
   ```

2. Configure the DHCPv4 relay profile.

   ```
   RP/0/RSP0/CPU0:router(config-dhcpv4)# profile helper relay
   ```

3. Configure helper addresses.

   You can configure up to 16 IPv4 addresses.

   ```
   RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 1.1.1.1
   RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 2.2.2.2
   RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 3.3.3.3
   RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 4.4.4.4
   ```

```
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 5.5.5.5
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 6.6.6.6
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 7.7.7.7
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 8.8.8.8
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 9.9.9.9
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 10.10.10.10
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 11.11.11.11
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 12.12.12.12
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 13.13.13.13
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 14.14.14.14
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 15.15.15.15
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf default 16.16.16.16
```

4. Confirm your configuration.

```
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# show configuration
Thu Feb  2 13:49:15.605 IST
Building configuration...
!! IOS XR Configuration 0.0.0
dhcp ipv4
 profile helper relay
  helper-address vrf default 1.1.1.1
  helper-address vrf default 2.2.2.2
  helper-address vrf default 3.3.3.3
  helper-address vrf default 4.4.4.4
  helper-address vrf default 5.5.5.5
  helper-address vrf default 6.6.6.6
  helper-address vrf default 7.7.7.7
  helper-address vrf default 8.8.8.8
  helper-address vrf default 9.9.9.9
  helper-address vrf default 10.10.10.10
  helper-address vrf default 11.11.11.11
  helper-address vrf default 12.12.12.12
  helper-address vrf default 13.13.13.13
  helper-address vrf default 14.14.14.14
  helper-address vrf default 15.15.15.15
  helper-address vrf default 16.16.16.16
 !
!
end
```

5. Commit your configuration.

```
RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# commit
```

6. Exit the configuration mode and verify the configured helper addresses.

```
RP/0/RSP0/CPU0:router# show dhcp ipv4 relay profile name helper
...
Profile: helper
Helper Addresses:
        1.1.1.1, vrf default
        2.2.2.2, vrf default
        3.3.3.3, vrf default
        4.4.4.4, vrf default
        5.5.5.5, vrf default
        6.6.6.6, vrf default
        7.7.7.7, vrf default
        8.8.8.8, vrf default
        9.9.9.9, vrf default
        10.10.10.10, vrf default
        10.10.10.11, vrf default
        10.10.10.13, vrf default
        10.10.10.14, vrf default
        10.10.10.15, vrf default
        10.10.10.16, vrf default
        10.10.10.17, vrf default
Information Option: Disabled
Information Option Allow Untrusted: Disabled
Information Option VPN: Disabled
Information Option VPN Mode: RFC
Information Option Policy: Replace
Information Option Check: Disabled
GIADDR Policy: Keep
Broadcast-flag Policy: Ignore
VRF References:
Interface References:
```

You have successfully configured multiple DHCPv4 relay helper addresses.

Configuring a DHCP Proxy Profile

The DHCP proxy performs all the functions of a relay and also provides some additional functions. The DHCP proxy conceals DHCP server details from DHCP clients. The DHCP proxy modifies the DHCP replies such that the client considers the proxy to be the server. In this state, the client interacts with the proxy as if it is the DHCP server.

This task describes how to configure and enable the DHCP proxy profile.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **proxy**
4. **helper-address** [**vrf** *vrf- name*] *address* [**giaddr** *gateway-address*]
5. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters DHCP IPv4 configuration submode .
Step 3	**profile** *profile-name* **proxy** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile client proxy`	Enters DHCP IPv4 profile proxy submode.

	Command or Action	Purpose
Step 4	**helper-address** [**vrf** *vrf- name*] *address* [**giaddr** *gateway-address*] **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-proxy-profile)# helper-address vrf1 10.10.1.1`	Forwards UDP broadcasts, including DHCP. • The value of the *address* argument can be a specific DHCP server address or a network address (if other DHCP servers are on the destination network segment). Using the network address enables other servers to respond to DHCP requests. • For multiple servers, configure one helper address for each server.
Step 5	**commit**	

Configuring DHCPv6 Relay Binding Database Write to System Persistent Memory

Perform this task to configure the DHCPv6 relay binding database write to the system persistent memory. This helps to recover the DHCPv6 relay binding table after a system reload. The file names used for a full persistent file write are *dhcpv6_srpb_{nodeid}_odd* and *dhcpv6_srpb_{nodeid}_even*. The *nodeid* is the actual node ID of the node where the file is written. The incremental file is named the same way as the full file, with a *_inc* appended to it.

Note With IOS XR Release 6.6.3, DHCPv6 client binding record format written to system persistent memory is changed. Due to this, when you upgrade IOS XR Software from versions lower to 6.6.3 to version 6.6.3 or above, the DHCPv6 process fails to restore the client bindings from the system persistent memory during router reload, and the router losses all the client bindings.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv6**
3. **database** [**relay**] [**full-write-interval** *full-write-interval*] [**incremental-write-interval** *incremental-write-interval*]
4. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router# configure`	
Step 2	**dhcp ipv6** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	Configures DHCP for IPv6 and enters the DHCPv6 configuration mode.
Step 3	**database** [**relay**] [**full-write-interval** *full-write-interval*] [**incremental-write-interval** *incremental-write-interval*] **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# database relay full-write-interval 20 incremental-write-interval 10`	Configures the DHCPv6 relay binding table write to the system persistent memory and specifies the time interval at which the full write and incremental file write are to be performed. The range, in minutes, for *full-write-interval* and *incremental-write-interval* is from 0 to 1440. The default value is 10 for *full-write-interval* and 1 for *incremental-write-interval*. The DHCP mode should be set as **relay**.
Step 4	**commit**	

Configuring DHCPv6 relay binding database write to system persistent memory: Example

```
 configure
 dhcp ipv6
 database relay full-write-interval 15 incremental-write-interval 5
 !
end
```

DHCPv4 Server

DHCP server accepts address assignment requests and renewals and assigns the IP addresses from predefined groups of addresses contained within Distributed Address Pools (DAPS). DHCP server can also be configured to supply additional information to the requesting client such as subnet mask, domain-name, the IP address of the DNS server, the default router, and other configuration parameters. DHCP server can accept broadcasts from locally attached LAN segments or from DHCP requests that have been forwarded by other DHCP relay agents within the network.

The DHCP proxy performs all the functions of a relay and also provides some additional functions. The DHCP proxy conceals DHCP server details from DHCP clients. The DHCP proxy modifies the DHCP replies such that the client considers the proxy to be the server. In this state, the client interacts with the proxy as if it is the DHCP server.

DHCP IPv4 service based mode selection

As part of DHCP IPv4 service based mode selection feature, a new mode called DHCP base is introduced. If an interface is configured in the DHCP base mode, then the DHCP selects either the DHCP proxy or the DHCP server mode to process the client request by matching option 60 (class-identifier) value of the client request with the configured value under the DHCP base profile.

For example:

```
dhcp ipv4
profile DHCP_BASE base
  match option 60 41424344 profile DHCP_PROXY proxy
  match option 60 41424355 profile DHCP_SERVER server
  default profile DEFAULT_PROFILE server
  relay information authenticate inserted
   !
profile DHCP_PROXY proxy
  helper-address vrf default 10.10.10.1 giaddr 0.0.0.0
!
profile DHCP_SERVER server
  lease 1 0 0
  pool IP_POOL
!
profile DEFAULT_PROFILE server
  lease 1 0 0
  pool IP_POOL
!
!
interface gigabitEthernet 0/0/0/0 base profile DHCP_BASE
```

The pool is configured under server-profile-mode and server-profile-class-sub-mode. The class-based pool selection is always given priority over profile pool selection.

The DHCPv4 server profile class sub-mode supports configuring DHCP options except few (0, 12, 50, 52, 53, 54, 58, 59, 61, 82, and 255).

Configuring DHCPv4 Server Profile

Perform this task to configure the DHCPv4 Server.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **server**
4. **bootfile** *boot-file-name*
5. **broadcast-flag policy** *unicast-always*
6. **class** *class-name*
7. **exit**
8. **default-router** *address1 address2 ... address8*
9. **lease** {**infinite** | *days minutes seconds* }
10. **limit lease** {**per-circuit-id** | **per-interface** | **per-remote-id** } *value*
11. **netbios-name server** *address1 address2 ... address8*
12. **netbios-node-type** {**number** | **b-node** | **h-node** | **m-node** | **p-node** }
13. **option** *option-code* {**ascii** *string* | **hex** *string* | **ip** *address*}
14. **pool** *pool-name*
15. **requested-ip-address-check disable**
16. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**dhcp ipv4**
	Example:
	`RP/0/RSP0/CPU0:router(config) # dhcp ipv4`
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`
		Enables DHCP for IPv4 and enters DHCP IPv4 configuration mode.

Step 3	**profile** *profile-name* **server**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4 )# profile TEST server`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Enters the server profile configuration mode.

Step 4	**bootfile** *boot-file-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# bootfile b1`	Configures the boot file.
Step 5	**broadcast-flag policy** *unicast-always* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# broadcast-flag policy unicast-always`	Configures the broadcast-flag policy to unicast-always.

Step 6	**class** *class-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# class Class_A`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile-class)`
		Creates and enters server profile class configuration submode.

Step 7	**exit** **Example:**	Exits the server profile class submode.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile-class)#`
	` exit`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`	

Step 8	**default-router** *address1 address2 ... address8*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
	` default-router 10.20.1.2`
		Configures the name of the default-router or the IP address.

Step 9	**lease** {**infinite** | *days minutes seconds* }
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
	` lease infinite`
		Configures the lease for an IP address assigned from the pool.

Step 10	**limit lease** {**per-circuit-id** | **per-interface** | **per-remote-id** } *value*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
	` limit lease per-circuit-id 23`
		Configures the limit on a lease per-circuit-id, per-interface, or per-remote-id.

Step 11	**netbios-name server** *address1 address2 ... address8*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
	` netbios-name-server 10.20.3.5`
		Configures the NetBIOS name servers.

Step 12	**netbios-node-type** {**number** | **b-node** | **h-node** | **m-node** | **p-node** }
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
	` netbios-node-type p-node`
		Configures the type of NetBIOS node.

Step 13	**option** *option-code* {**ascii** *string* | **hex** *string* | **ip** *address*} **Example:**	Configures the DHCP option code.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# option 23 ip 10.20.34.56`	
Step 14	**pool** *pool-name* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# pool pool1`	Configures the Distributed Address Pool Service (DAPS) pool name.
Step 15	**requested-ip-address-check disable** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# requested-ip-address-check disable`	Validates a requested IP address.
Step 16	**commit**	

Configuring Multiple Classes with a Pool

Perform this task to configure multiple classes with a pool.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **server**
4. **pool** *pool-name*
5. **class** *class-name*
6. **pool** *pool_name*
7. **match option** *option* [**sub-option** *sub-option*] [**ascii** *asciiString* | **hex** *hexString*]
8. **exit**
9. **class** *class-name*
10. **pool** *pool_name*
11. **match vrf** *vrf-name*
12. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**dhcp ipv4**
	Example:
	`RP/0/RSP0/CPU0:router(config) # dhcp ipv4`
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`
		Enables DHCP for IPv4 and enters DHCP IPv4 configuration mode.

Step 3	**profile** *profile-name* **server**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4 )# profile TEST server`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Enters the server profile configuration mode.

Step 4	**pool** *pool-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# pool POOL_TEST`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Configures the Distributed Address Pool Service(DAPS) pool name.

Step 5	**class** *class-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# class Class_A`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Creates and enters the server profile class.

Step 6	**pool** *pool_name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)# pool pool_A`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Configures the pool name.

	Command or Action	Purpose

Step 7	**match option** *option* [**sub-option** *sub-option*] [**ascii** *asciiString* | **hex** *hexString*]
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
	` match option 60 hex abcd`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		The DHCP server selects a pool from a class by matching options in the received DISCOVER packet with the match option. If none of the classes match, then pools configured under the profile mode are selected. The DHCP server requests DAPS to allocate an address from that pool.

Step 8	**exit**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
	` exit`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Exits the server profile class submode.

Step 9	**class** *class-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
	` class Class_B`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Creates and enters the server profile class.

Step 10	**pool** *pool_name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
	` pool pool_B`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Configures the pool name.

Step 11	**match vrf** *vrf-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
	` match vrf VRF1`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		The DHCP server selects a pool from a class by matching the options in the received DISCOVER packet with the match command. If none of the classes match, then pools configured under the profile mode are selected. The DHCP server requests DAPS to allocate an address from that pool.

Step 12	**commit**	

Configuring a server profile DAPS with class match option

Perform this task to configure a server profile DAPS with class match option.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **server**
4. **pool** *pool-name*
5. **class** *class-name*
6. **pool***pool_name*
7. **match option** *option* [**sub-option** *sub-option*] [**ascii** *asciiString* | **hex** *hexString*]
8. **exit**
9. **exit**
10. **profile** *profile-name* **server**
11. **dns-server** *address1 address2 ... address8*
12. **pool** *pool_name*
13. **class** *class-name*
14. **pool***pool_name*
15. **match option** *option* [**sub-option** *sub-option*] [**ascii** *asciiString* | **hex** *hexString*]
16. **exit**
17. **exit**
18. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**dhcp ipv4**
	Example:
	`RP/0/RSP0/CPU0:router(config) # dhcp ipv4`
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`
		Enables DHCP for IPv4 and enters DHCP IPv4 configuration mode.

Step 3	**profile** *profile-name* **server**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4 )# profile ISP1 server`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Enters the server profile configuration mode.

	Command or Action	Purpose

Step 4	**pool** *pool-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# pool ISP1_POOL`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Configures the Distributed Address Pool Service(DAPS) pool name.

Step 5	**class** *class-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# class ISP1_CLASS`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Creates and enters the server profile class.

Step 6	**pool***pool_name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)# pool ISP1_CLASS_POOL`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Configures the pool name.

Step 7	**match option** *option* [**sub-option** *sub-option*] [**ascii** *asciiString* | **hex** *hexString*]
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)# match option 60 hex PXEClient_1`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		The DHCP server selects a pool from a class by matching the options in the received DISCOVER packet with the match option. If none of the classes match, then pools configured under the profile mode will be selected. The DHCP server requests the DAPS to allocate an address from that pool.

Step 8	**exit**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)# exit`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-prfile)#`
		Exits the server profile class sub mode.

Step 9	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# exit`	Exits the server profile sub mode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`	

Step 10	**profile** *profile-name* **server**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4 )# profile ISP2 server`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Enters the server profile configuration mode.

Step 11	**dns-server** *address1 address2 ... address8*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# dns-server 10.20.3.4`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Configures the name of the DNS server or the IP address

Step 12	**pool** *pool_name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)# pool ISP2_POOL`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Configures the pool name.

Step 13	**class** *class-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# class ISP2_CLASS`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Creates and enters the server profile class.

Step 14	**pool***pool_name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)# pool ISP2_CLASS_POOL`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		Configures the pool name.

Step 15	**match option** *option* [**sub-option** *sub-option*] [**ascii** *asciiString* | **hex** *hexString*] **Example:**	The DHCP server selects a pool from a class by matching the options in the received DISCOVER packet with the match option. If none of the classes match, then pools configured under the profile mode will be selected. The

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
	` match option 60 hex PXEClient_2`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
		DHCP server requests the DAPS to allocate an address from that pool.

Step 16	**exit**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-class)#`
	` exit`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Exits the server profile class sub mode.

Step 17	**exit**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
	` exit`
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`
		Exits the server profile sub mode.

Step 18	**commit**	

Configuring Server Profile without daps pool match option

Perform this task to configure a server profile without daps pool match option.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **server**
4. **dns-server** *address1 address2 ... address8*
5. **exit**
6. **profile** *profile-name* **server**
7. **dns-server** *address1 address2 ... address8*
8. **exit**
9. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose

Step 2	**dhcp ipv4**
	Example:
	`RP/0/RSP0/CPU0:router(config) # dhcp ipv4`
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`
		Enables DHCP for IPv4 and enters DHCP IPv4 configuration mode.

Step 3	**profile** *profile-name* **server**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4 )# profile ISP1 server`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Enters the server profile configuration mode.

Step 4	**dns-server** *address1 address2 ... address8*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# dns-server ISP1.com`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Configures the name of the DNS server or IP address.

Step 5	**exit**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# exit`
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`
		Exits the server profile sub mode.

Step 6	**profile** *profile-name* **server**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4 )# profile ISP2 server`
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`
		Enters the server profile configuration mode.

Step 7	**dns-server** *address1 address2 ... address8* **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# dns-server ISP2.com`	Configures the name of the DNS server or IP address.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)#`	

Step 8	**exit**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# exit`
	`RP/0/RSP0/CPU0:router(config-dhcpv4)#`
		Exits the server profile sub mode.

Step 9	**commit**	

Configuring an address pool for each ISP on DAPS

Perform this task to configure an address pool for each ISP on Distributed Address Pool Service(DAPS).

SUMMARY STEPS

1. **configure**
2. **pool vrf** [**all** \| *vrf-name*] { **ipv4** \| **ipv6** } *pool-name*
3. **network** *address*
4. **exit**
5. **pool vrf** [**all** \| *vrf-name*] { **ipv4** \| **ipv6** } *pool-name*
6. **network** *address*
7. **exit**
8. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**pool vrf** [**all** | *vrf-name*] { **ipv4** | **ipv6** } *pool-name*
	Example:
	`RP/0/RSP0/CPU0:router(config) # pool vrf ISP_1 ipv4 ISP1_POOL`
	`RP/0/RSP0/CPU0:router(config-pool-ipv4)#`
		Configures an IPv4 pool for the specifed VRF or all vrfs.

Step 3	**network** *address* **Example:**	Specifies network for allocation.

	Command or Action	Purpose

	`RP/0/RSP0/CPU0:router(config-pool-ipv4)# network`
	`10.10.10.0`
	`RP/0/RSP0/CPU0:router(config-pool-ipv4)#`	

Step 4	**exit**
	Example:
	`RP/0/RSP0/CPU0:router(config-pool-ipv4)# exit`
	`RP/0/RSP0/CPU0:router(config)#`
		Exits the pool ipv4 configuration submode.

Step 5	**pool vrf** [**all** | *vrf-name*] { **ipv4** | **ipv6** } *pool-name*
	Example:
	`RP/0/RSP0/CPU0:router(config) # pool vrf ISP_2 ipv4`
	` ISP2_POOL`
	`RP/0/RSP0/CPU0:router(config-pool-ipv4)#`
		Configures an IPv4 pool for the specifed VRF or all vrfs.

Step 6	**network** *address*
	Example:
	`RP/0/RSP0/CPU0:router(config-pool-ipv4)# network`
	`20.20.20.0`
	`RP/0/RSP0/CPU0:router(config-pool-ipv4)#`
		Specifies network for allocation.

Step 7	**exit**
	Example:
	`RP/0/RSP0/CPU0:router(config-pool-ipv4)# exit`
	`RP/0/RSP0/CPU0:router(config)#`
		Exits the pool ipv4 configuration submode.

Step 8	**commit**	

DHCPv4 Client

The Dynamic Host Configuration Protocol (DHCP) client functionality enables the router interfaces to dynamically acquire the IPv4 address using DHCP.

The DHCP provides configuration parameters to Internet hosts. DHCP consists of two components:

- a protocol to deliver host-specific configuration parameters from a DHCP server to a host.
- a mechanism to allocate network addresses to hosts.

DHCP is built on a client-server model, where designated DHCP server hosts allocate network addresses, and deliver configuration parameters to dynamically configured hosts.

A relay agent is required if the client and server are not on the same Layer 2 network. The relay agent usually runs on the router, and is required because the client device does not know its own IP address initially. The agent sends out a Layer 2 broadcast to find a server that has this information. The router relays these broadcasts to the DHCP server, and forwards the responses back to the correct Layer 2 address so that the correct device gets the correct configuration information.

DHCP has the ability to allocate IP addresses only for a configurable period of time, called the lease period. If the client is required to retain this IP address for a longer period beyond the lease period, the lease period must be renewed before the IP address expires. The client renews the lease based on configuration that was sent from the server. The client unicasts a REQUEST message using the IP address of the server. When a server receives the REQUEST message and responds with an ACK message. The lease period of the client is extended by the lease time configured in the ACK message.

Restrictions and Limitations

- DHCP client can be enabled only on management interfaces.
- Either DHCP or static IP can be configured on an interface.

Enabling DHCP Client on an Interface

The DHCPv4 or DHCPv6 client can be enabled at an interface level. The DHCP component receives a notification when DHCPv4 or DHCPv6 is enabled or disabled on an interface.

```
Router# configure
Router(config)# interface MgmtEth rack/slot/CPU0/port
Router(config)# interface interface_name ipv6 address dhcp
```

DHCPv6 Relay Agent Notification for Prefix Delegation

DHCPv6 relay agent notification for prefix delegation allows the router working as a DHCPv6 relay agent to find prefix delegation options by reviewing the contents of a DHCPv6 RELAY-REPLY packet that is being relayed by the relay agent to the client. When the relay agent finds the prefix delegation option, the relay agent extracts the information about the prefix being delegated and inserts an IPv6 subscriber route matching the prefix delegation information onto the relay agent. Future packets destined to that prefix via relay are forwarded based on the information contained in the prefix delegation. The IPv6 subscriber route remains in the routing table until the prefix delegation lease time expires or the relay agent receives a release packet from the client releasing the prefix delegation.

The relay agent automatically does the subscriber route management.

The IPv6 routes are added when the relay agent relays a RELAY-REPLY packet, and the IPv6 routes are deleted when the prefix delegation lease time expires or the relay agent receives a release message. An IPv6 subscriber route in the routing table of the relay agent can be updated when the prefix delegation lease time is extended.

This feature leaves an IPv6 route on the routing table of the relay agent. This registered IPv6 address allows unicast reverse packet forwarding (uRPF) to work by allowing the router doing the reverse lookup to confirm that the IPv6 address on the relay agent is not malformed or spoofed. The IPv6 route in the routing table of the relay agent can be redistributed to other routing protocols to advertise the subnets to other nodes. When the client sends a DHCP_DECLINE message, the routes are removed.

Configuring DHCPv6 Stateful Relay Agent for Prefix Delegation

Perform this task to configure Dynamic Host Configuration Protocol (DHCP) IPv6 relay agent notification for prefix delegation.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv6**
3. **profile** *profile-name* **proxy**
4. **helper-address** *ipv6-address* **interface** *type interface-path-id*
5. **exit**
6. **interface** *type interface-path-id* **proxy**
7. **profile** *profile-name*
8. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**dhcp ipv6**
	Example:
	`RP/0/RSP0/CPU0:router(config) # dhcp ipv6`
	`RP/0/RSP0/CPU0:router(config-dhcpv6)#`
		Enables DHCP for IPv6 and enters DHCP IPv6 configuration mode.

Step 3	**profile** *profile-name* **proxy**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv6)# profile downstream proxy`
	`RP/0/RSP0/CPU0:router(config-dhcpv6-profile)#`
		Enters the proxy profile configuration mode.

Step 4	**helper-address** *ipv6-address* **interface** *type interface-path-id*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv6-profile)# helper-address 2001:db8::1 GigabitEthernet 0/1/0/1`
	`RP/0/RSP0/CPU0:router(config-dhcpv6-profile)`
		Configure the DHCP IPv6 relay agent.

	Command or Action	Purpose

Step 5	**exit**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv6-profile)# exit`
	`RP/0/RSP0/CPU0:router(config-dhcpv6)#`
		Exits from the profile configuration mode.

Step 6	**interface** *type interface-path-id* **proxy**
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv6)# interface GigabitEthernet 0/1/0/0 proxy`
	`RP/0/RSP0/CPU0:router(config-dhcpv6-if)#`
		Enables IPv6 DHCP on an interface and acts as an IPv6 DHCP stateful relay agent.

Step 7	**profile** *profile-name*
	Example:
	`RP/0/RSP0/CPU0:router(config-dhcpv6-if)# profile downstream`
	`RP/0/RSP0/CPU0:router(config-dhcpv6-if)#`
		Enters the profile configuration mode.

Step 8	**commit**	

Enabling Secure ARP

Secure ARP is disabled by default; this task describes how to enable secure ARP.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. Do one of the following:
    - **profile** *profile-name* **proxy**
    - **profile** *profile-name* **server**
4. **secure-arp**
5. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters DHCP IPv4 configuration mode.
Step 3	Do one of the following: • **profile** *profile-name* **proxy** • **profile** *profile-name* **server** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile profile1 server`	Enters DHCP IPv4 profile proxy or server submode.
Step 4	**secure-arp** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-server-profile)# secure-arp`	Enables secure ARP.
Step 5	**commit**	

Configuration Examples for the DHCP Relay Agent

This section provides the following configuration examples:

DHCP Relay Profile: Example

The following example shows how to configure the Cisco IOS XR relay profile:

```
dhcp ipv4
 profile client relay
  helper-address vrf foo 10.10.1.1
 !
! ...
```

DHCP Relay on an Interface: Example

The following example shows how to enable the DHCP relay agent on an interface:

```
dhcp ipv4
 interface GigabitEthernet 0/1/1/0 relay profile client
!
```

DHCP Relay on a VRF: Example

The following example shows how to enable the DHCP relay agent on a VRF:

```
dhcp ipv4
 vrf default relay profile client
!
```

Relay Agent Information Option Support: Example

The following example shows how to enable the relay agent and the insertion and removal of the DHCP relay information option:

```
dhcp ipv4
 profile client relay
relay information option

 !
!
```

Relay Agent Giaddr Policy: Example

The following example shows how to configure relay agent giaddr policy:

```
dhcp ipv4
 profile client relay
  giaddr policy drop
 !
!
```

Implementing DHCP Snooping

Prerequisites for Configuring DHCP Snooping

The following prerequisites are required example shows how to configure DHCP IPv4 snooping relay agent broadcast flag policy:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- A Cisco ASR 9000 Series Router running Cisco IOS XR software.

- A configured and running DHCP client and DHCP server.

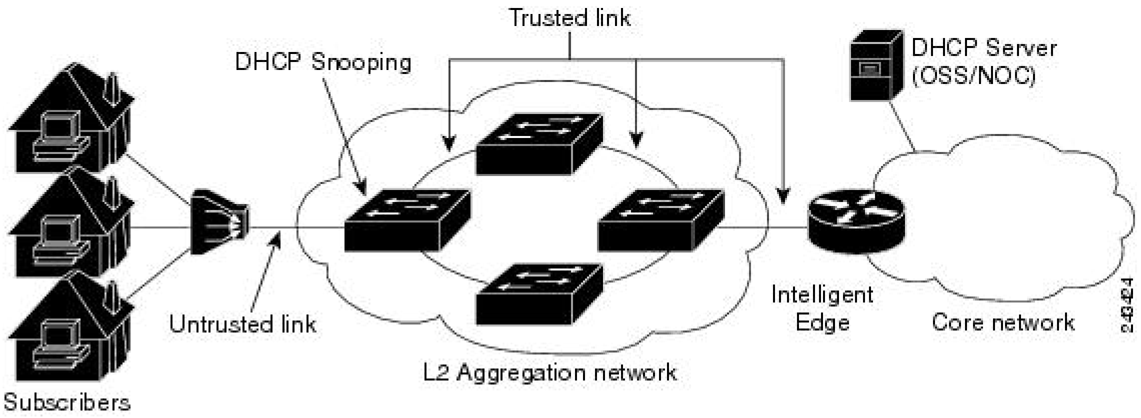
Information about DHCP Snooping

DHCP Snooping features are focused on the edge of the aggregation network. Security features are applied at the first point of entry for subscribers. Relay agent information option information is used to identify the subscriber's line, which is either the DSL line to the subscriber's home or the first port in the aggregation network.

The central concept for DHCP snooping is that of trusted and untrusted links. A trusted link is one providing secure access for traffic on that link. On an untrusted link, subscriber identity and subscriber traffic cannot be determined. DHCP snooping runs on untrusted links to provide subscriber identity. Figure 2: DHCP Snooping in an Aggregation Network, on page 34 shows an aggregation network. The link from the DSLAM to the aggregation network is untrusted and is the point of presence for DHCP snooping. The links connecting the switches in the aggregation network and the link from the aggregation network to the intelligent edge is considered trusted.

Note Enabling both DHCP relay on a BVI and DHCP snooping in a bridge domain that has a BVI can result in duplicate DHCP messages from the DHCP client to the DHCP server.

Figure 2: DHCP Snooping in an Aggregation Network



Trusted and Untrusted Ports

On trusted ports, DHCP BOOTREQUEST packets are forwarded by DHCP snooping. The client's address lease is not tracked and the client is not bound to the port. DHCP BOOTREPLY packets are forwarded.

When the first DHCP BOOTREQUEST packet from a client is received on an untrusted port, DHCP snooping binds the client to the bridge port and tracks the clients's address lease. When that address lease expires, the client is deleted from the database and is unbound from the bridge port. Packets from this client received on this bridge port are processed and forwarded as long as the binding exists. Packets that are received on another bridge port from this client are dropped while the binding exists. DHCP snooping only forwards DHCP BOOTREPLY packets for this client on the bridge port that the client is bound to. DHCP BOOTREPLY packets that are received on untrusted ports are not forwarded.

DHCP Snooping in a Bridge Domain

To enable DHCP snooping in a bridge domain, there must be at least two profiles, a trusted profile and an untrusted profile. The untrusted profile is assigned to the client-facing ports, and the trusted profile is assigned to the server-facing ports. In most cases, there are many client facing ports and few server-facing ports. The simplest example is two ports, a client-facing port and a server-facing port, with an untrusted profile explicitly assigned to the client-facing port and a trusted profile assigned to the server-facing port.

Assigning Profiles to a Bridge Domain

Because there are normally many client-facing ports and a small number of server-facing ports, the operator assigns the untrusted profile to the bridge domain. This configuration effectively assigns an untrusted profile to every port in the bridge domain. This action saves the operator from explicitly assigning the untrusted profile to all of the client-facing ports. Because there also must be server-facing ports that have trusted DHCP snooping profiles, in order for DHCP snooping to function properly, this untrusted DHCP snooping profile assignment is overridden to server-facing ports by specifically configuring trusted DHCP snooping profiles on the server-facing ports. For ports in the bridge domain that do not require DHCP snooping, all should have the **none** profile assigned to them to disable DHCP snooping on those ports.

Relay Information Options

You can configure a DHCP snooping profile to insert the relay information option (option 82) into DHCP client packets only when it is assigned to a client port. The **relay information option allow-untrusted** command addresses what to do with DHCP client packets when there is a null giaddr and a relay-information option already in the client packet when it is received. This is a different condition than a DHCP snooping trusted/untrusted port. The **relay information option allow-untrusted** command determines how the DHCP snooping application handles untrusted relay information options.

How to Configure DHCP Snooping

This section contains the following tasks:

Enabling DHCP Snooping in a Bridge Domain

The following configuration creates two ports, a client-facing port and a server-facing port. In Step 1 through Step 8, an untrusted DHCP snooping profile is assigned to the client bridge port and trusted DHCP snooping profile is assigned to the server bridge port. In Step 9 through Step 18, an untrusted DHCP snooping profile is assigned to the bridge domain and trusted DHCP snooping profiles are assigned to server bridge ports.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *untrusted-profile-name* **snoop**
4. **exit**
5. **dhcp ipv4**
6. **profile** *profile-name* **snoop**
7. **trusted**
8. **exit**
9. **l2vpn**

10. **bridge group** *group-name*
11. **bridge-domain** *bridge-domain-name*
12. **interface** *type interface-path-id*
13. **dhcp ipv4 snoop profile** *untrusted-profile-name*
14. **interface** *type interface-path-id*
15. **dhcp ipv4 snoop profile** *trusted-profile-name*
16. **exit**
17. **exit**
18. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters DHCP IPv4 profile configuration submode.
Step 3	**profile** *untrusted-profile-name* **snoop** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile untrustedClientProfile snoop`	Configures an untrusted DHCP snooping profile for the client port.
Step 4	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# exit`	Exits DHCP IPv4 profile configuration mode.
Step 5	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enables DHCP for IPv4 and enters DHCP IPv4 profile configuration mode.
Step 6	**profile** *profile-name* **snoop** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile trustedServerProfile snoop`	Configures a trusted DHCP snooping profile for the server port.
Step 7	**trusted** **Example:** `RP/0/RSP0/CPU0:router(config-dhcv4)# trusted`	Configures a DHCP snoop profile to be trusted.

	Command or Action	Purpose
Step 8	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-dhcv4)# exit`	Exits DHCP IPv4 profile configuration mode.
Step 9	**l2vpn** **Example:** `RP/0/RSP0/CPU0:router(config)# l2vpn`	Enters l2vpn configuration mode.
Step 10	**bridge group** *group-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group ccc`	Creates a bridge group to contain bridge domains and enters l2vpn bridge group configuration submode.
Step 11	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ddd`	Establishes a bridge domain.
Step 12	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gigabitethernet 0/1/0/0`	Identifies an interface.
Step 13	**dhcp ipv4 snoop profile** *untrusted-profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# dhcp ipv4 snoop profile untrustedClientProfile`	Attaches an untrusted DHCP snoop profile to the bridge port.
Step 14	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# gigabitethernet 0/1/0/1`	Identifies an interface.
Step 15	**dhcp ipv4 snoop profile** *trusted-profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# dhcp ipv4 snoop profile trustedServerProfile`	Attaches a trusted DHCP snoop profile to the bridge port.
Step 16	**exit** **Example:**	Exits the l2vpn bridge group bridge-domain interface configuration submode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit`	
Step 17	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# exit`	Exits the l2vpn bridge group bridge-domain configuration submode.
Step 18	**commit**	

Disabling DHCP Snooping on a Specific Bridge Port

The following configuration enables DHCP to snoop packets on all bridge ports in the bridge domain ISP1 except for bridge port GigabitEthernet 0/1/0/1 and GigabitEthernet 0/1/0/2. DHCP snooping is disabled on bridge port GigabitEthernet 0/1/0/1. Bridge port GigabitEthernet 0/1/0/2 is the trusted port that connects to the server. In this example, no additional features are enabled, so only DHCP snooping is running.

SUMMARY STEPS

1. **configure**
2. **l2vpn**
3. **bridge group** *group-name*
4. **bridge-domain** *bridge-domain-name*
5. **dhcp ipv4 snoop profile** *profile-name*
6. **interface** *type interface-path-id*
7. **dhcp ipv4 none**
8. **interface** *type interface-path-id*
9. **dhcp ipv4 snoop profile** *profile-name*
10. **exit**
11. **exit**
12. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**l2vpn** **Example:** `RP/0/RSP0/CPU0:router(config)# l2vpn`	Enters l2vpn configuration submode.
Step 3	**bridge group** *group-name* **Example:**	Creates a bridge group to contain bridge domains and enters l2vpn bridge group configuration submode.

	Command or Action	Purpose
	`RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group GRP1`	
Step 4	**bridge-domain** *bridge-domain-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	Establishes a bridge domain and enters l2vpn bridge group bridge-domain configuration submode.
Step 5	**dhcp ipv4 snoop profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# dhcp ipv4 snoop profile untrustedClientProfile`	Attaches the untrusted DHCP snooping profile to the bridge domain.
Step 6	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gigabitethernet 0/1/0/1`	Identifies an interface and enters l2vpn bridge group bridge-domain interface configuration submode.
Step 7	**dhcp ipv4 none** **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-if)# dhcp ipv4 none`	Disables DHCP snooping on the port.
Step 8	**interface** *type interface-path-id* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gigabitethernet 0/1/0/2`	Identifies an interface and enters l2vpn bridge group bridge-domain interface configuration submode.
Step 9	**dhcp ipv4 snoop profile** *profile-name* **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# dhcp ipv4 snoop profile trustedServerProfile`	Attaches the trusted DHCP snooping profile to a port.
Step 10	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bd-bg)# exit`	Exits l2vpn bridge-domain bridge group interface configuration submode.
Step 11	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# exit`	Exits l2vpn bridge-domain submode.

	Command or Action	Purpose
Step 12	**commit**	

Using the Relay Information Option

This task shows how to use the relay information commands to insert the relay information option (option 82) into DHCP client packets and forward DHCP packets with untrusted relay information options.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **snoop**
4. **relay information option**
5. **relay information option allow-untrusted**
6. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**dhcp ipv4** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	Enters DHCP IPv4 profile configuration submode.
Step 3	**profile** *profile-name* **snoop** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile untrustedClientProfile snoop`	Configures an untrusted DHCP snooping profile for the client port.
Step 4	**relay information option** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-snoop-profile)# relay information option`	Enables the system to insert the DHCP relay information option field in forwarded BOOTREQUEST messages to a DHCP server.
Step 5	**relay information option allow-untrusted** **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv4-snoop-profile)# relay information option allow-untrusted`	Configures DHCP IPv4 relay not to discard BOOTREQUEST packets that have an existing relay information option and the giaddr set to zero.
Step 6	**commit**	

Configuration Examples for DHCP Snooping

This section provides the following configuration examples:

Assigning a DHCP Profile to a Bridge Domain: Example

The following example shows how to enable DHCP snooping in a bridge domain:

```
l2vpn
 bridge group GRP1
  bridge-domain ISP1
   dhcp ipv4 profile untrustedClientProfile snoop
```

Disabling DHCP Snooping on a Specific Bridge Port: Example

The following example shows how to disable DHCP snooping on a specific bridge port:

```
interface gigabitethernet 0/1/0/1
 dhcp ipv4 none
```

Configuring a DHCP Profile for Trusted Bridge Ports: Example

The following example shows how to configure a DHCP profile for trusted bridge ports:

```
dhcp ipv4 profile trustedServerProfile snoop
 trusted
```

Configuring an Untrusted Profile on a Bridge Domain: Example

The following example shows how to attach a profile to a bridge domain and disable snooping on a bridge port.

```
l2vpn
 bridge group GRP1
  bridge-domain ISP1
   dhcp ipv4 profile untrustedClientProfile snoop
    interface gigabitethernet 0/1/0/1
     dhcp ipv4 none
```

Configuring a Trusted Bridge Port: Example

The following example shows ow to assign a trusted DHCP snooping profile to a bridge port:

```
l2vpn
 bridge group GRP1
  bridge-domain ISP1
   interface gigabitethernet 0/1/0/2
    dhcp ipv4 profile trustedServerProfile snoop
```

DHCPv6 Proxy Binding Table Reload Persistency

The Cisco IOS-XR Dynamic Host Configuration Protocol (DHCP) application is responsible for maintaining the DHCP binding state for the DHCP leases allocated to clients by the DHCP application. These binding states are learned by the DHCP application (proxy/relay/snooping). DHCP clients expect to maintain a DHCP lease regardless of the events that occur to the DHCP application.

Note From Release 6.2.2 onwards, 200K sessions are supported on a proxy or server running DHCPv6.

This feature enables the DHCP application to maintain bind state through the above events:

- Process restart – Local checkpoint
- RP failover – Hot standby RP through checkpoint
- LC IMDR – Local checkpoint
- LC OIR – Shadow table on RP
- System restart – Bindings saved on local disk

Configuring DHCPv6 Proxy Binding Database Write to System Persistent Memory

Perform this task to configure the DHCPv6 binding database write to the system persistent memory. This helps to recover the DHCPv6 binding table after a system reload. The file names used for a full persistent file write are *dhcpv6_srpb_{nodeid}_odd* and *dhcpv6_srpb_{nodeid}_even*. The *nodeid* is the actual node ID of the node where the file is written. The incremental file is named the same way as the full file, with a *_inc* appended to it.

Note From Release 6.2.2 onwards, 200K sessions are supported on a proxy or server running DHCPv6.

SUMMARY STEPS

1. **configure**
2. **dhcp ipv6**
3. **database** [**proxy**] [**full-write-interval** *full-write-interval*] [**incremental-write-interval** *incremental-write-interval*]
4. **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RSP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**dhcp ipv6** **Example:** `RP/0/RSP0/CPU0:router(config)# dhcp ipv6`	Configures DHCP for IPv6 and enters the DHCPv6 configuration mode.
Step 3	**database** [**proxy**] [**full-write-interval** *full-write-interval*] [**incremental-write-interval** *incremental-write-interval*] **Example:** `RP/0/RSP0/CPU0:router(config-dhcpv6)# database proxy full-write-interval 20 incremental-write-interval 10`	Configures the DHCPv6 binding table write to the system persistent memory and specifies the time interval at which the full write and incremental file write are to be performed. The range, in minutes, for *full-write-interval* and *incremental-write-interval* is from 0 to 1440. The default value is 10 for *full-write-interval* and 1 for *incremental-write-interval*. The DHCP mode should be set as **proxy**.
Step 4	**commit**	

Configuring DHCP binding database write to system persistent memory: Example

```
 configure
 dhcp ipv6
 database proxy full-write-interval 15 incremental-write-interval 5
 !
end
```

DHCP Session MAC Throttle

The ASR9K router supports the DHCP session MAC throttle feature. This feature limits the number of DHCP client requests reaching the ASR9K, based on the MAC address of the DHCP clients. This feature is supported for the DHCPv4 proxy, the DHCPv4 server, and the DHCPV6 proxy. The feature prevents a DHCP client from sending multiple DISCOVER packets to the ASR9K router, within short periods of time. This, in turn, prevents that client from impacting the session establishment of other DHCP clients.

Note From Release 6.2.2 onwards, 200K sessions are supported on a proxy or server running DHCPv6.

A unique throttle entry is created in the system for each unique MAC address received on any interface where the profile is attached.

To configure the DHCP session MAC throttle feature, use the **sessions mac throttle** command in the respective DHCP profile configuration mode.

Configuring DHCP Session MAC Throttle: Example

```
dhcp ipv4
 profile p1 server
  sessions mac throttle 300 60 40
 !
 interface GigabitEthernet0/0/0/0 server profile p1
!
```

Relay response on source interface

Setting relay response on a source interface is a functionality that allows you to send the OFFER back through the same interface that received the DISCOVER.

Benefits of setting relay response on source interface

These are the benefits of setting relay response on source interface:

- Consistent network path: Ensures that both DISCOVER and OFFER packets travel the same path, which can simplify troubleshooting and monitoring.
- Reduces configuration errors: Minimizes the risk of OFFER packets being blocked or misrouted due to unexpected interfaces or routing decisions.
- Predictable traffic flow: Keeps network traffic predictable and easy to follow, which is helpful for network administrators.

Configure relay response on source interface

Use these commands to configure relay response on source interface.

Procedure

Step 1 Enter global configuration mode.

Example:

```
Router# configure
```

Step 2 Enable the DHCP IPv4 configuration mode.

Example:

```
Router(config)# dhcp ipv4
```

Step 3 Configure the DHCPv4 relay profile.

Example:

```
Router(config-dhcpv4)# profile profile-test server
```

Step 4 Enable the relay response on source interface option.

Example:

```
Router(config-dhcpv4-server-profile)# relay-response-on-src-intf
```

Step 5 Save your changes.

Example:

```
Router(config-dhcpv4-server-profile)# commit
```

Step 6 Verify the configuration.

Example:

```
Router#show running-config dhcp ipv4 profile profile-test server
Wed Jul  9 12:16:40.594 UTC
dhcp ipv4
profile profile-test server
relay-response-on-src-intf
!
!
```

Configure DHCPv6 Relay Source Address

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Configure DHCPv6 relay source address	Release 25.1.1	You can now select an IPv6 address from the configured relay source-interface to be used as the source address for forwarding packets to a server. By selecting a fixed source address, the need to frequently update firewall rules when new, lower-value IPv6 addresses are added is minimized. Previously, the router automatically used the lowest numbered IPv6 address configured on that interface as the source address. The feature introduces these changes: **CLI**: • The *dhcpv6 relay source address* variable is introduced in the helper-address (ipv6) command. **YANG Data Model**: Cisco-IOS-XR-ipv6-new-dhcpv6d-cfg.yang (see GitHub, YANG Data Models Navigator)

The feature introduces the capability to select a specific IPv6 address from the configured relay source-interface to be used as the source address for forwarding packets to the server. Previously, the system would automatically use the lowest numbered IPv6 address configured on that interface as the source address. This prior behavior required you to update the firewall rules whenever new IPv6 addresses of lower value were added to a DHCPv6 enabled interface.

Benefits of Configuring DHCPv6 Relay Source Address

The benefits of configuring DHCPv6 relay source address are:

- Flexibility – You can now choose a specific IPv6 address from the relay source-interface, allowing more control over which address is used for packet forwarding.
- Reduced firewall updates – By selecting a fixed source address, the need to frequently update firewall rules when new, lower-value IPv6 addresses are added is minimized.
- Enhanced security – With a stable relay source address, security policies can be more consistently applied, reducing the risk of misconfigurations.
- Improved troubleshooting – Having a predictable relay source address makes it easier to track and troubleshoot network traffic issues.

Configuration Guidelines and Restrictions for Configuring DHCPv6 Relay Source Address

These restrictions apply if you configure DHCPv6 relay source address:

- Removing the configured DHCPv6 relay source-interface IP disables the feature and retains the old behaviour of automatically using the lowest numbered IPv6 address configured on that interface as the source address.
- You can configure the DHCPv6 relay source address under **helper-address** and in the **source-interface** as well.

Configuration Example

The following section details the configuration for DHCPv6 relay source address:

```
/* Enter the global configuration mode, and then enter the DHCP IPv6 configuration mode */
Router# configure terminal
Router(config)# dhcp ipv6
Router(config-dhcpv6)# profile test relay
Router(config-dhcpv6-relay-profile)# helper-address vrf default 2011::3 TenGigE0/0/0/0
1001::10
Router(config-dhcpv6-relay-profile)# helper-address vrf default 2011::4
Router(config-dhcpv6-relay-profile)# source-interface TenGigE0/1/0/0  1001::1
Router(config-dhcpv6-relay-profile)# !
Router(config-dhcpv6-relay-profile)# interface Bundle-Ether1 relay profile test
Router(config-dhcpv6-relay-profile)# commit
Router(config-dhcpv6)# !
```

In this example, when a packet is received on the profile **relay test**, the router:

- The address `1001::10`, specified as the DHCPv6 relay source address, on interface **`TenGigE0/0/0/0`** is used to send packets to server `2011::3`.
- The address `1001::1`, specified as the DHCPv6 relay source address, on interface **`TenGigE0/1/0/0`** is used to send packets to server `2011::4`.

Note To disable this feature, use the **no** form of the command.

Running Configuration

To verify the DHCPv6 relay source address configuration:

```
Router#show running-config interface TenGigE0/0/0/0
Wed Jan 15 17:48:09.635 UTC
dhcp ipv6
profile test relay
  helper-address vrf default 2011::3 TenGigE0/0/0/0  1001::10
  helper-address vrf default 2011::4
  source-interface TenGigE0/1/0/0  1001::1
!
interface Bundle-Ether1 relay profile test
```

DHCPv6 relay subscriber ID

Relay subscriber ID, also known as option 38, is a functionality within the DHCPv6 (Dynamic Host Configuration Protocol for IPv6) system that

- allows a DHCPv6 relay agent to send a unique identifier to the DHCPv6 server
- specifies the Layer 2 (L2) sub-interface where a client request originated, and
- helps the DHCPv6 server assign network addresses more efficiently.

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
DHCPv6 relay subscriber ID	Release 25.4.1	Introduced in this release on: NCS 5500 fixed port routers; NCS 5700 fixed port routers; NCS 5500 modular routers (NCS 5500 line cards; NCS 5700 line cards [Mode: Native; Compatibility] You can now configure DHCPv6 relay subscriber ID option 38 in the DHCPv6 relay profile. This feature allows DHCPv6 relay agents to send a relay subscriber ID, also known as option 38, to the DHCPv6 server. You can configure unique IDs on L2 sub-interfaces. When a client request arrives on a DHCPv6 relay, the relay agent adds option 38 into the relay-forward message, providing the DHCPv6 server with the client's originating interface for efficient address assignment. The feature introduces these changes: **CLI**: • The **relay subscriber-id** *subscriber-id-value* command is introduced. **YANG Data Model**: Cisco-IOS-XR-ipv6-new-dhcpv6d-cfg.yang (see GitHub, YANG Data Models Navigator)

This feature allows a DHCPv6 relay agent to send a unique identifier to the DHCPv6 server. This identifier, known as the relay subscriber ID, specifies the Layer 2 (L2) sub-interface where a client request originates. This enhancement helps the DHCPv6 server assign network addresses more efficiently. Option 38 is enabled on L2 sub-interfaces within a Bridge Virtual Interface (BVI) or Integrated Routing and Bridging (IRB) setup.

Benefits of relay subscriber ID

These are the benefits of using relay subscriber ID on L2 sub-interfaces:

- The relay subscriber ID option 38 improves DHCPv6 server address assignment. The server receives specific context about the client's originating L2 interface.
- It allows network administrators to configure a unique ID for each L2 sub-interface.

Configuration guidelines for relay subscriber ID

These guidelines apply when you configure relay subscriber ID option 38:

- Configure the relay subscriber ID on a per-L2 sub-interface basis.
- The subscriber ID value must be an ASCII string. The string length must be between 1 and 128 characters. The router rejects configurations outside this range.
- This feature is enabled only in DHCPv6 relay mode.
- It applies only to L2 sub-interfaces associated with a BVI or IRB.

Restrictions for relay subscriber ID

These restrictions apply for relay subscriber ID option 38:

- The relay subscriber ID string must contain a minimum of 1 character and a maximum of 128 characters.
- This feature functions only in DHCPv6 relay mode.

How relay agent subscriber ID works

When a client's DHCPv6 request arrives on an L2 sub-interface, the relay agent checks for a configured relay subscriber ID. If found, the relay agent inserts this subscriber ID as option 38 into the relay-forward message before sending it to the DHCPv6 server.

Summary

The key components involved in the process are:

- **Client:**

  Initiates the DHCPv6 request.
- **L2 sub-interface:**

  The interface where the client's request arrives at the relay agent.
- **Relay agent:**

  Processes client requests, checks for configured IDs, and inserts option 38 into relay-forward messages.
- **Relay subscriber ID:**

  The unique ASCII string configured on L2 sub-interfaces.
- **DHCPv6 option 38:**

  The specific DHCPv6 option used to carry the relay subscriber ID.
- **Encapsulation layer:**

  The new layer created by the relay agent where Option 38 is inserted.

Workflow

These stages describe how relay subscriber ID works:

1. A client packet arrives at the relay agent on an L2 sub-interface.

2. The relay agent checks for a configured relay subscriber ID on that specific L2 sub-interface.
3. If a subscriber ID is configured, the relay agent inserts DHCPv6 option 38 into the relay-forward message.
4. The relay agent adds option 38 within a new encapsulation layer it creates for the packet.
5. The relay agent then sends the relay-forward message to the DHCPv6 server.
6. If no subscriber ID is configured for the L2 sub-interface, the relay agent does not insert option 38.

Result

An internal trace indicates when option 38 is inserted. This internal trace displays the value, its length, and the access interface.

Configure relay subscriber ID

Use this procedure to configure relay subscriber ID option 38.

Procedure

Step 1 Enter global configuration mode and then enter the DHCPv6 configuration mode.

Example:

```
Router# configure
Router(config)# dhcp ipv6
```

Step 2 Define or enter an existing DHCPv6 relay profile.

Example:

```
Router(config-dhcpv6)# profile <profile-name> relay
```

Step 3 Specify the IPv6 address of the DHCPv6 server.

Example:

```
Router(config-dhcpv6-profile# helper-address <helper-ipv6-address>
```

The helper address is also known as the DHCPv6 server address.

Step 4 Exit the DHCPv6 profile configuration mode.

Example:

```
Router(config-dhcpv6-profile)# !
```

Step 5 Enter the specific L2 sub-interface configuration mode where the client traffic originates and configure the relay subscriber ID for the interface.

Example:

```
Router(config-dhcpv6)# interface <l2-sub-interface-name> relay subscriber-id <subscriber-id-value>
```

Step 6 Exit the interface configuration mode and the global configuration mode.

Example:

```
Router(config-if)# !
Router(config)# end
```

Step 7 Save the configuration.

Example:

```
Router# commit
```

Step 8 Verify the configuration and observe the system traces for the TP3911 message, which indicates the insertion of option 38. This confirms the relay agent is actively adding the subscriber ID to the relay-forward packets.

Example:

```
Router# show running-config dhcp ipv6
Router# show trace dhcpv6d
```

- Look for the configured **interface** <*l2-sub-interface-name*> **relay subscriber-id** <*subscriber-id-value*> entry.
- Look for output similar to:

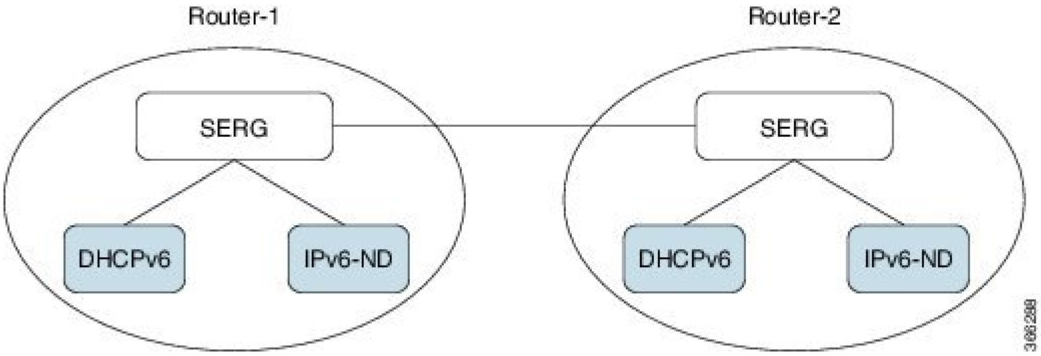
```
DHCPV6 RELAY INTERNAL: TP3911: Inserting Relay_Subscriber_ID BUNDLE(6) for interface
      Bundle-Ether10.100
```

Geographical Redundancy By Using a Session Redundancy Group (SERG)

In large scale network implementations, it becomes essential to have redundancy between routers that share the same core network (IP and MPLS), but are geographically apart. A redundancy thus achieved is known as geographical redundancy, and often consists of a switchover (SO) from the active (primary) router to the standby (subordinate) router.

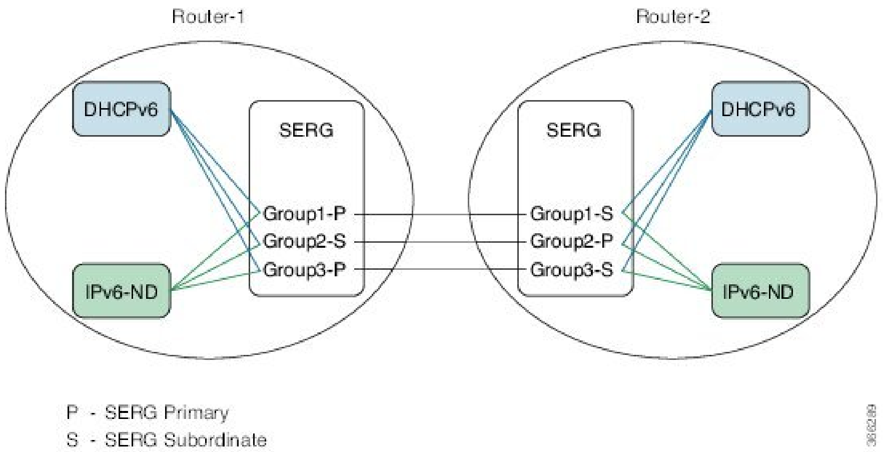
To achieve geographical redundancy for IPv6 Neighbor Discovery (ND) entries, or for DHCPv6 bindings, we use a Session Redundancy Group (SERG). A SERG comprises of sessions mapped to the access interfaces on the active RP of the router. If a single SERG is configured on the active RPs of the primary and subordinate routers, then the router hosting the primary SERG serves as the primary, and the router hosting the subordinate SERG serves as the subordinate. This is illustrated in the following figure.

Figure 3: Geo Redundancy with a Single SERG



When multiple SERGs are configured on the active RPs, you could have both primary and subordinate SERGs on a single router. This is illustrated in the following figure.

Figure 4: Geo Redundancy with Multiple SERGs



Each router has an inbuilt redundancy between the RPs. When the active RP fails, the session (s) is transferred to the standby RP. This is known as a failover (FO).

The Session Redundancy Manager (SERM) runs on the active RP of both primary and subordinate routers. The SR clients running on the routers interact with the Session Redundancy Infrastructure (Session Redundancy Agent (SRA) and the Session Redundancy Library (SRL)).

The various components and their functions are briefly described as follows:

- **Session Redundancy Manager (SERM)**: The SERM runs as a separate process on the active RP and manages the SERG configuration. The SERM peers with other routers that need to form a redundancy relationship, and establishes a point-to-multipoint communication channel to Session Redundancy Agents (SRAs) on the RP.
- **Session Redundancy Agent (SRA)**: One or more SRAs run as a a separate process on the active RP and supported line cards. A SRA acts on the SERG configuration, setting up operational context and database tables. The SRA implements the state machine for primary/subordinate selection and role change and orchestrates it using the TCP channel and provided APIs.The SRA receives the session entries on the primary router and updates its database prior to synchronizing with the database on the subordinate router. The SRA orchestrates the session context setup on the subordinate router during the FO or SO. The SRA maintains a separate session database for each session client configured in the SERG.

Note The SRA works only on specific, defined keys, such as the IPv6 address, DHCPv6 client ID, and so on. Any undefined session data is handled as opaque data by the SRA. The respective session components must provide their access library to the SRA for handling any transformation or data retrieval.

- **Session Redundancy Library (SRL)**: The SRL is used by session components for communicating with the SRA. The SRL uses IPC semantics for communicating with the SRA. SERG clients use an asychronous API for storing and retrieving the session state from the SRL.

You can configure object tracking for one or more access interfaces in the SERG to enable automatic switchovers when an interface goes down. For more information on this configuration, see the *BNG Command Reference Guide for Cisco ASR 9000 Series Routers*.

Limitations for SERG

- If the Address Cached in the primary router: When the device configured as a primary router loses connectivity with the device configured as a subordinate router, the primary router continues to provide addresses assigned to the group. If one or more subscribers restart, the previously assigned IP addresses remain in the cache until communication is restored between both SERG entities. This situation could potentially exhaust the address pool if many subscribers frequently disconnect and reconnect, if a few subscribers continuously flap, or if the **clear subscribers command** is used.
- If a device is configured as a subordinate router: This device does not provide any address if the connectivity to the primary device is lost. In case this device assumes the role of primary, for example, after a session-redundancy switchover group was executed or for any other reason, there could be situations where address duplication occurs.

Guidelines for SERG

- The connectivity between SERG devices should be stable, as communication occurs over a TCP session using port 4001. It is good practice to prioritize this traffic.
- In the event of a prolonged disconnection between devices configured for SERG, it is advisable to configure a peer removal due to the caching mechanism. During the disconnection, addresses provided by the pool are tagged as local. Once the connection is restored, they will be tagged as local and remote.

Configuring and Verifying Session Redundancy for DHCPv6 Clients

Use the following procedure to configure geo-redundancy through session redundancy for DHCPv6 clients.

In this example, we configure Router 1 as Primary and Router 2 as Subordinate.

1. On Routers R1 and R2, enter the global configuration mode and configure session redundancy by specifying Loopback 0 as the source interface.

```
Router# configure
Router(config)# session redundancy
Router(config-session-red)# hold-timer 5
Router(config-session-red)# source-interface loopback0
```

Note The hold timer values on Routers R1 and R2 must match for them to peer with each other.

2. Configure the session redundancy group by specifying the preferred role as Primary for Router R1 using the **master** keyword, and as subordinate for Router R2 using the **slave** keyword.

Router R1:

```
Router(config)# session redundancy group 1
Router(config-session-red-group)# preferred-role master
Router(config-session-red-group)# hold-timer 7
Router(config-session-red-group)# peer 2.2.2.2
Router(config-session-red-group)# revertive-timer 5 maximum 15
Router(config-session-red-group)# interface-list
Router(config-session-red-grp-intf)# interface GigabitEthernet0/1/0/0 id 1
```

Router R2:

```
Router(config)# session redundancy group 1
Router(config-session-red-group)# preferred-role slave
Router(config-session-red-group)# hold-timer 7
Router(config-session-red-group)# peer 1.1.1.1
Router(config-session-red-group)# revertive-timer 5 maximum 15
Router(config-session-red-group)# interface-list
Router(config-session-red-grp-intf)# interface GigabitEthernet0/1/0/0 id 1
```

Note The hold timer, revertive timer, and interface ID values on Routers R1 and R2 must match for them to peer with each other.

3. Exit to the global configuration mode and commit your configuration on Routers R1 and R2.

```
Router(config)# commit
```

4. Confirm your configuration on Router R1.

```
Router# show running-config session-redundancy
...
session-redundancy
 source-interface Loopback0
 hold-timer 5
 group 1
  preferred-role master
  hold-timer 7
  peer 2.2.2.2
  revertive-timer 5 maximum 15
  interface-list
   interface GigabitEthernet0/1/0/0 id 1
  !
 !
!
```

5. Confirm your configuration on Router R2.

```
Router# show running-config session-redundancy
...
session-redundancy
 source-interface Loopback0
 hold-timer 5
 group 1
  preferred-role slave
  hold-timer 7
  peer 1.1.1.1
  revertive-timer 5 maximum 15
  interface-list
   interface GigabitEthernet0/1/0/0 id 1
  !
 !
!
```

6. Verify the session redundancy group on the routers by running the following show commands.

```
Router# show session-redundancy group
...
Session Redundancy Agent Group Summary
Flags     : E - Enabled, D - Disabled, M - Preferred Master, S - Preferred Slave
            H - Hot Mode, W - Warm Mode, T - Object Tracking Enabled
P/S       : Peer Status
            I - Initialize, Y - Retry, X - Cleanup, T - Connecting
            L - Listening, R- Registered, C - Connected, E - Established
```

```
I/F Count: Interface Count
SS Count : Session Count
-----------------------------------------------------------------------------------------

   Node Name   | Group ID | Role | Flags  |         Peer Address        | P/S | I/F Count
 |  SS Count  | Sync Pending
-----------------------------------------------------------------------------------------

 0/1/CPU0              1  Master   EMH-   2.2.2.2                         E
 1          0               0
-----------------------------------------------------------------------------------------
Session Summary Count(Master/Slave/Total): 0/0/0


Router# show session-redundancy group 1
...
Session Redundancy Group ID: 1
  Description         : <<not-configured>>
  Status              : Enabled

  Init-Role           : Master
  Negotiated-Role     : Master
  Current-Role        : Master

  Hold Time           : 7
  Revert Time         : 5

  Tracking Status     : Enabled
    Core-Tracking     : <<not-configured>>
      Status          : n/a
    Access-Tracking   : <<not-configured>>
      Status          : n/a

  Peer:
    IP-address        : 2.2.2.2
    Status            : Established
    Role(Init/Neg/Cur): Slave/Slave/Slave
    Tracking Status   : Up

    Last Neg-Time     : 2017 Mar  2 18:14:42
    Last Up-Time      : 2017 Mar  2 18:14:42
    Last Down-Time    : 2017 Mar  2 18:14:26

  Switchover:
    Last Switchover   : 2017 Mar  2 18:14:42       Reason         : Peer Up
    Switchover Count  : 1
    Hold Time         : Not-Running
    Revert Time       : Not-Running

  Session Statistics:
    Count             : 0                          Slave-Upd-Fail : 0
    Pending Update    : 0                          Pending Delete : 0
    Client:
      IPv6ND          : 0
      DHCPv6          : 0

  Interface Count     : 1
    GigabitEthernet0/1/0/0                         Map-ID         : 1


Router# show session-redundancy summary interface
...
Session Redundancy Interface Summary
```

```
Status: E - Exists, F - Forward Reference
-----------------------------------------------------------------------------
       Interface Name                  | Status | Group ID | Map ID |  Role
-----------------------------------------------------------------------------
 GigabitEthernet0/1/0/0                   E              1        1   Master
-----------------------------------------------------------------------------
```

7. Verify the SRG session information on the routers.

```
Router# show session-redundancy group 1 session verbose
...
Session Redundancy Agent Group Session
       Flags: M-Master, V-Valid MAC, N-Neg Ack
        Comp: SA-Agent, ND-ipv6nd, D6-dhcpv6
  Comp Flags: U-Update, D-Delete, S-InSync, F-TxListFail, T-Dirty, C-Cleanup
    Err Info: X-xxxx-ec - H/S - Hard/Soft, xxxx - No. of Times, ec - Error Code
--------------------------------------------------------------------------------------

 Parent Interface              | Key index                                | Flags |
  Comp Flags                 | Synchronization Error Info
--------------------------------------------------------------------------------------

 GigabitEthernet0/1/0/0          00030001ca011bba000000000000000000000000   M-
SA{S} D6{S}                                       -
--------------------------------------------------------------------------------------
```

8. Verify the SRA information and statistics.

```
Router# show session-redundancy agent interface
...
Session Redundancy Agent Interface
Status   : F - Forward Referenced, S - Stale, R - Registered,
           A - CAPS Added, O - Resource Owned, P - EOMS Pending
           C - Pending CAPS Remove, U - Pending Reg Disable
Err Stats: Enable - Disable - Caps Add - Caps Remove - Attr Updated
---------------------------------------------------------------------------------------

       Interface Name          |  ID  | Group ID | Role  |  Status  | Oper  | Err Stats
---------------------------------------------------------------------------------------

 GigabitEthernet0/1/0/0               1         1   Master  --RA----   -----  0-0-0-0-0
---------------------------------------------------------------------------------------

Router# show session-redundancy agent statistics
...
Session Redundancy Agent Summary - Node 0/0/CPU0
  Process State                   : Active

  Source Interface                : Loopback0
    VRF Name                      : default
    IPv4 Address                  : 1.1.1.1
    IPv6 Address                  : 192::2

  Restart Client Sync In Progress : No
    Client Init Sync TimeStamp    : -
  Restart Peer Sync In Progress   : No
    Peer Init Sync TimeStamp      : -
  Sync in Progress                : No
  Peer Action Timer               : Not-Running
  Retry Timer                     : Not-Running
  Interface Status Statistics
      Bound to group              : 1
      Non stale                  : 0
      Pending caps remove        : 0
      Pending reg disable        : 0
      Pending other batch oper   : 0
  Sync in Progress               : No

Client Statistics:
  Status: U - Connection UP, S - Init-Sync Pending, E - Sync EOD Pending
--------------------------------------------------------------------------------
  Comp  | Status |     Up Timestamp     |    Down Timestamp    | Cleanup Timer
--------------------------------------------------------------------------------
 SERGAGT     ---    -                      -                             0
 IPv6ND     U--    2017 Mar  2 18:14:25   -                            0
 DHCPv6     U--    2017 Mar  2 18:14:25   -                            0
--------------------------------------------------------------------------------

TxList Statistics:                                     Ok   Part-Write        Clean
------------------------------------------------------------------------------------
  Marker Encode                           :             4            0            4
  Command Encode                          :             0            0            0
  Negotiation Encode                      :             0            0            0

Client Statistics:                                     Ok        NotOk
-----------------------------------------------------------------------
  Invalid Registration                    :                          0
  Invalid DeRegistration                  :                          0
  Connection Up Count                     :             2
  Connection Down Count                   :                          0
  Message CallBack Count                  :             2
  Message Received                        :             4            0
  Command Message Received                :             0            0
  Session Message Received                :             4            0
  Peer Done                               :             2

Peer Statistics:                                       Ok        NotOk
-----------------------------------------------------------------------
  Timer Handler                           :             0
  Invalid Registration                    :                          0
  Invalid DeRegistration                  :                          0
  Message CallBack Count                  :             0            0
  Command Connection Up                   :                          0
  Command Connection Down                 :                          0
  Session Connection Up                   :                          0
  Session Connection Down                 :                          0
  Peer Done                               :             0
-----------------------------------------------------------------------
```

9. Verify the DHCPv6 SR client information on the routers.

```
Router#show session-redundancy agent client dhcpv6
...
Session Redundancy Agent Client Statistics - Node 0/0/CPU0
Component - DHCPv6
Statistics:                                            Ok        NotOk
-------------------------------------------------------------------------------
Sent To Client:
  Command
    Start of Download  - SOD              :             1            0
    End of Download    - EOD              :             1            0
    End of Master Sync - EOMS             :             0            0
    Clear              - All              :             0            0
    Clear              - Selected         :             0            0
    Replay             - All              :             0            0
    Replay             - Selected          :            0            0
  Session                                  :            0            0
    Update                                 :            0            0
    Delete                                 :            0

TxList Operation:
  Encode - Complete Write                  :            0
  Encode - Partial Write                   :            0
  Cleanup CallBack                         :            0
Last Replay Count                          :            0

Received From Client:
  Command
    Start of Download  - SOD   - All       :            1
    Start of Download  - SOD   - Selected  :            0
    End of Download    - EOD   - All       :            1
    End of Download    - EOD   - Selected  :            0
    End of Master Sync - EOMS              :            0
    Clear              - All               :            0
    Clear              - Selected          :            0
    Replay             - All               :            0
    Replay             - Selected          :            0
  Session
    Update                                 :            0            0
    Delete                                 :            0            0
  Negative Acknowledgement                 :            0            0

Client Activity Statistics:
  Active                                   :            1            0
  Deactive                                 :            0            0
  Registration                             :            1            0
  DeRegistration                           :            0
  Connection Down                          :            0
  Cleanup                                  :            0
-------------------------------------------------------------------------------

Session Redundancy Agent Client Statistics - Node 0/1/CPU0
Component - DHCPv6
Statistics:                                            Ok        NotOk
-------------------------------------------------------------------------------
Sent To Client:
  Command
    Start of Download  - SOD               :            1            0
    End of Download    - EOD               :            1            0
    End of Master Sync - EOMS              :            1            0
    Clear              - All               :            0            0
    Clear              - Selected          :            0            0
    Replay             - All               :            0            0
    Replay             - Selected          :            0            0
  Session                                  :            0            0
    Update                                 :            0            0
    Delete                                 :            0

TxList Operation:
  Encode - Complete Write                  :            0
  Encode - Partial Write                   :            0
  Cleanup CallBack                         :            0
Last Replay Count                          :            0

Received From Client:
  Command
    Start of Download  - SOD   - All       :            1
    Start of Download  - SOD   - Selected  :            0
    End of Download    - EOD   - All       :            1
    End of Download    - EOD   - Selected  :            0
    End of Master Sync - EOMS              :            0
    Clear              - All               :            0
    Clear              - Selected          :            0
    Replay             - All               :            0
    Replay             - Selected          :            0
  Session
    Update                                 :            0            3
    Delete                                 :            0            2
  Negative Acknowledgement                 :            0            0

Client Activity Statistics:
  Active                                   :            1            0
  Deactive                                 :            0            0
  Registration                             :            1            0
  DeRegistration                           :            0
  Connection Down                          :            0
  Cleanup                                  :            0
-------------------------------------------------------------------------------

Session Redundancy Agent Client Statistics - Node 0/2/CPU0
Component - DHCPv6
Statistics:                                            Ok        NotOk
-------------------------------------------------------------------------------
Sent To Client:
  Command
    Start of Download  - SOD               :            1            0
    End of Download    - EOD               :            1            0
    End of Master Sync - EOMS              :            0            0
    Clear              - All               :            0            0
    Clear              - Selected          :            0            0
    Replay             - All               :            0            0
    Replay             - Selected          :            0            0
  Session                                  :            0            0
    Update                                 :            0            0
    Delete                                 :            0

TxList Operation:
  Encode - Complete Write                  :            0
  Encode - Partial Write                   :            0
  Cleanup CallBack                         :            0
Last Replay Count                          :            0

Received From Client:
  Command
    Start of Download  - SOD   - All       :            1
    Start of Download  - SOD   - Selected  :            0
    End of Download    - EOD   - All       :            1
    End of Download    - EOD   - Selected  :            0
    End of Master Sync - EOMS              :            0
    Clear              - All               :            0
    Clear              - Selected          :            0
    Replay             - All               :            0
    Replay             - Selected          :            0
  Session
    Update                                 :            0            0
    Delete                                 :            0            0
  Negative Acknowledgement                 :            0            0

Client Activity Statistics:
  Active                                   :            1            0
  Deactive                                 :            0            0
  Registration                             :            1            0
  DeRegistration                           :            0
```

```
Connection Down                          :            0
Cleanup                                  :            0
```

You have successfully configured and verified geo redundancy using session redundancy groups for DHCPv6 clients.

Managing Session Redundancy Groups

After you have configured and verified the session redundancy groups (SERGs), you can use the commands in this section to trigger a manual switchover, trigger a manual synchronization, or clear sessions for all or a specific SERG.

Triggering a Manual Switchover

After you have configured SERGs on the primary and subordinate routers, if you want to remove/replace the primary router, you can trigger a manual switchover from the primary to the subordinate by running the following commands.

Note The following commands can be executed only on the primary router.

- To trigger a redundancy switchover for all SERGs, run the following command.

```
RP/0/RSP0/CPU0:router# session redundancy switchover
```

- To trigger a redundancy switchover for a specific SERG, run the following command.

```
RP/0/RSP0/CPU0:router# session redundancy switchover group 210
```

Triggering Manual Synchronization

If the sessions between the primary and subordinate routers are not getting synchronized, either because of some change in the network topology, or some network latency, you can trigger synchronization manually by running the following commands.

Note The following commands can be executed on either the Primary or the Subordinate router.

- To trigger a redundancy synchronization for all SERGs, run the following command.

```
RP/0/RSP0/CPU0:router# session redundancy synchronize
```

- To trigger a redundancy synchronization for a specific SERG, run the following command.

```
RP/0/RSP0/CPU0:router# session redundancy synchronize group 210
```

Clearing Sessions in a SERG

If you want to clear the existing sessions on the primary and subordinate routers, either because of a switchover, or a change in network topology, you can run the following commands.

Note The following commands can be executed on either the primary or the subordinate router.

When issued on the subordinate, the session context is deleted from the router and a synchronization is requested with the primary. If the router is in hot-standby mode, the sessions are deleted on the subordinate.

When issued on the primary, the session entries are deleted first on the primary and later on the subordinate. The SRA then requests a fresh session from the SR client, which is eventually synchronized with the subordinate.

- To clear sessions for all SERGs, run the following command.

```
RP/0/RSP0/CPU0:router# clear session-redundancy
```

- To clear sessions for a specific SERG, run the following command.

```
RP/0/RSP0/CPU0:router# clear session-redundancy group 1
```

Configuring and Verifying Session Redundancy for IPv6 ND Clients

Use the following procedure to configure geo-redundancy through session redundancy for IPv6 ND clients.

In this example, we configure Router 1 as Primary and Router 2 as Subordinate.

1. On Routers R1 and R2, enter the global confiiguration mode and configure session redundancy by specifying Loopback 0 as the source interface.

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# session redundancy
RP/0/RSP0/CPU0:router(config-session-red)# hold-timer 5
RP/0/RSP0/CPU0:router(config-session-red)# source-interface loopback0
```

Note The hold timer values on Routers R1 and R2 must match for them to peer with each other.

2. Configure the session redundancy group by specifying the preferred role as Primary for Router R1 using the **master** keyword, and as subordinate for Router R2 using the **slave** keyword.

Router R1 :

```
RP/0/RSP0/CPU0:router(config)# session redundancy group 1
RP/0/RSP0/CPU0:router(config-session-red-group)# preferred-role master
RP/0/RSP0/CPU0:router(config-session-red-group)# hold-timer 7
RP/0/RSP0/CPU0:router(config-session-red-group)# peer 2.2.2.2
RP/0/RSP0/CPU0:router(config-session-red-group)# revertive-timer 5 maximum 15
RP/0/RSP0/CPU0:router(config-session-red-grp-intf)# interface GigabitEthernet0/1/0/0 id
 1
```

Router R2:

```
RP/0/RSP0/CPU0:router(config)# session redundancy group 1
RP/0/RSP0/CPU0:router(config-session-red-group)# preferred-role slave
RP/0/RSP0/CPU0:router(config-session-red-group)# hold-timer 7
RP/0/RSP0/CPU0:router(config-session-red-group)# peer 1.1.1.1
RP/0/RSP0/CPU0:router(config-session-red-group)# revertive-timer 5 maximum 15
RP/0/RSP0/CPU0:router(config-session-red-grp-intf)# interface GigabitEthernet0/1/0/0 id
 1
```

Note The hold timer, revertive timer, and interface ID values on Routers R1 and R2 must match for them to peer with each other.

3. Exit to the global configuration mode and commit your configuration on Routers R1 and R2.

```
RP/0/RSP0/CPU0:router(config)# commit
```

4. Confirm your configuration on Router R1 (Primary).

```
RP/0/RSP0/CPU0:router# show running-config session-redundancy
...
session-redundancy
 source-interface Loopback0
 hold-timer 5
 group 1
  preferred-role master
  hold-timer 7
  peer 2.2.2.2
  revertive-timer 5 maximum 15
   interface GigabitEthernet0/1/0/0 id 1
  !
 !
!
```

5. Confirm your configuration on Router R2.

```
RP/0/RSP0/CPU0:router# show running-config session-redundancy
...
session-redundancy
 source-interface Loopback0
 hold-timer 5
 group 1
  preferred-role slave
  hold-timer 7
  peer 1.1.1.1
  revertive-timer 5 maximum 15
   interface GigabitEthernet0/1/0/0 id 1
  !
 !
!
```

6. Verify the session redundancy group on the routers by running the following show commands.

```
RP/0/RSP0/CPU0:router# show session-redundancy group
...
Session Redundancy Agent Group Summary
Flags    : E - Enabled, D - Disabled, M - Preferred Master, S - Preferred Slave
           H - Hot Mode, W - Warm Mode, T - Object Tracking Enabled
P/S      : Peer Status
           I - Initialize, Y - Retry, X - Cleanup, T - Connecting
           L - Listening, R- Registered, C - Connected, E - Established
I/F Count: Interface Count
```

```
SS Count : Session Count
-------------------------------------------------------------------------------------------

   Node Name   | Group ID | Role | Flags  |         Peer Address        | P/S | I/F Count
 |  SS Count  | Sync Pending
-------------------------------------------------------------------------------------------

 0/1/CPU0              1  Master   EMH-   2.2.2.2                         E
 1           0              0
--------------------------------------------------------------------------------------------
Session Summary Count(Master/Slave/Total): 0/0/0

RP/0/RSP0/CPU0:router# show session-redundancy group 1
...
Session Redundancy Group ID: 1
  Description         : <<not-configured>>
  Status              : Enabled

  Init-Role           : Master
  Negotiated-Role     : Master
  Current-Role        : Master

  Hold Time           : 7
  Revert Time         : 5

  Tracking Status     : Enabled
    Core-Tracking     : <<not-configured>>
      Status          : n/a
    Access-Tracking   : <<not-configured>>
      Status          : n/a

  Peer:
    IP-address        : 2.2.2.2
    Status            : Established
    Role(Init/Neg/Cur): Slave/Slave/Slave
    Tracking Status   : Up

    Last Neg-Time     : 2017 Mar  2 18:14:42
    Last Up-Time      : 2017 Mar  2 18:14:42
    Last Down-Time    : 2017 Mar  2 18:14:26

  Switchover:
    Last Switchover   : 2017 Mar  2 18:14:42       Reason         : Peer Up
    Switchover Count  : 1
    Hold Time         : Not-Running
    Revert Time       : Not-Running

  Session Statistics:
    Count             : 0                          Slave-Upd-Fail : 0
    Pending Update    : 0                          Pending Delete : 0
    Client:
      IPv6ND          : 0
      DHCPv6          : 0

  Interface Count     : 1
    GigabitEthernet0/1/0/0                         Map-ID         : 1


RP/0/RSP0/CPU0:router# show session-redundancy summary interface
...
Session Redundancy Interface Summary
Status: E - Exists, F - Forward Reference
```

```
-----------------------------------------------------------------------------
       Interface Name                  | Status | Group ID | Map ID |  Role
-----------------------------------------------------------------------------
 GigabitEthernet0/1/0/0                   E            1        1   Master
-----------------------------------------------------------------------------
```

7. Verify the SRG session information on the routers.

```
RP/0/RSP0/CPU0:router# show session-redundancy group 1 session verbose
...
Session Redundancy Agent Group Session
       Flags: M-Master, V-Valid MAC, N-Neg Ack
        Comp: SA-Agent, ND-ipv6nd, D6-dhcpv6
  Comp Flags: U-Update, D-Delete, S-InSync, F-TxListFail, T-Dirty, C-Cleanup
    Err Info: X-xxxx-ec - H/S - Hard/Soft, xxxx - No. of Times, ec - Error Code
________________________________________________________________________________________

 Parent Interface              | Key index                                | Flags |
  Comp Flags                 | Synchronization Error Info
________________________________________________________________________________________

 GigabitEthernet0/1/0/0          00030001ca011bba000000000000000000000000   M-
SA{S} D6{S}                                     -
________________________________________________________________________________________
```

8. Verify the SRA information and statistics.

```
RP/0/RSP0/CPU0:router# show session-redundancy agent interface
...
Session Redundancy Agent Interface
Status   : F - Forward Referenced, S - Stale, R - Registered,
           A - CAPS Added, O - Resource Owned, P - EOMS Pending
           C - Pending CAPS Remove, U - Pending Reg Disable
Err Stats: Enable - Disable - Caps Add - Caps Remove - Attr Updated
--------------------------------------------------------------------------------------

       Interface Name          |  ID  | Group ID | Role  |  Status  | Oper  | Err Stats
--------------------------------------------------------------------------------------

 GigabitEthernet0/1/0/0              1         1   Master  --RA----   -----  0-0-0-0-0
--------------------------------------------------------------------------------------
```

```
RP/0/RSP0/CPU0:router# show session-redundancy agent statistics
...
Session Redundancy Agent Summary - Node 0/0/CPU0
  Process State                   : Active

  Source Interface                : Loopback0
    VRF Name                      : default
    IPv4 Address                  : 1.1.1.1
    IPv6 Address                  : 192::2

  Restart Client Sync In Progress : No
    Client Init Sync TimeStamp    : -
  Restart Peer Sync In Progress   : No
    Peer Init Sync TimeStamp      : -
  Sync in Progress                : No
  Peer Action Timer               : Not-Running
  Retry Timer                     : Not-Running
  Interface Status Statistics
      Bound to group              : 1
      Non stale                  : 0
      Pending caps remove        : 0
      Pending reg disable        : 0
      Pending other batch oper   : 0
  Sync in Progress               : No

Client Statistics:
  Status: U - Connection UP, S - Init-Sync Pending, E - Sync EOD Pending
--------------------------------------------------------------------------------
  Comp  | Status |     Up Timestamp     |    Down Timestamp    | Cleanup Timer
--------------------------------------------------------------------------------
 SERGAGT     ---    -                      -                             0
 IPv6ND     U--    2017 Mar  2 18:14:25   -                             0
 DHCPv6     U--    2017 Mar  2 18:14:25   -                             0
--------------------------------------------------------------------------------

TxList Statistics:                                     Ok   Part-Write        Clean
------------------------------------------------------------------------------------
  Marker Encode                            :            4            0            4
  Command Encode                           :            0            0            0
  Negotiation Encode                       :            0            0            0

Client Statistics:                                     Ok        NotOk
-----------------------------------------------------------------------
  Invalid Registration                     :                         0
  Invalid DeRegistration                   :                         0
  Connection Up Count                      :            2
  Connection Down Count                    :                         0
  Message CallBack Count                   :            2
  Message Received                         :            4            0
  Command Message Received                 :            0            0
  Session Message Received                 :            4            0
  Peer Done                                :            2

Peer Statistics:                                       Ok        NotOk
-----------------------------------------------------------------------
  Timer Handler                            :            0
  Invalid Registration                     :                         0
  Invalid DeRegistration                   :                         0
  Message CallBack Count                   :            0            0
  Command Connection Up                    :                         0
  Command Connection Down                  :                         0
  Session Connection Up                    :                         0
  Session Connection Down                  :                         0
  Peer Done                                :            0
-----------------------------------------------------------------------
...
```

9. Verify the IPv6 ND SR client information on the routers.

```
RP/0/RSP0/CPU0:router# show session-redundancy client ipv6nd
Session Redundancy Client Statistics - Node 0/0/CPU0

  Config    : True
  Status : Enabled
  Active    : True

  Connection Status         : Connected
  Last Connection Up Time   : 2017 Mar  7 10:28:03
  Last Connection Down Time : 1970 Jan  1 05:30:00

TxList Operation:
  Message CallBack                         :            2
  Encode - Complete Write                  :            7
  Encode - Partial Write                   :            0
  Cleanup CallBack                         :            0
  Decode Message Error                     :            0
  Unknown Operation Error                  :            0

TxList Statistics:                                     Ok   Part-Write        Clean
------------------------------------------------------------------------------------
  Marker Encode                            :            2            0            2
  Command Encode                           :            0            0            0
  Negotiation Encode                       :            0            0            0

Statistics:                                            Ok        NotOk
-------------------------------------------------------------------------------
Sent To Agent:
  Command
    Start of Download  - SOD               :            1            0
    End of Download    - EOD               :            1            0
    End of Master Sync - EOMS              :            0            0
    Clear              - All               :            0            0
    Clear              - Selected          :            0            0
    Replay             - All               :            0            0
    Replay             - Selected          :            0            0
  Session
    Add                                    :            7            0
    Delete                                 :            7            0
  Negative Acknowledgement
    Synchronous                            :            0            0
    Asynchronous                           :            0            0

Received From Agent:
Message                                    :            9            0
  Command                                  :            0            0
    Start of Download  - SOD   - All       :            1
    Start of Download  - SOD   - Selected  :            0
    End of Download    - EOD   - All       :            1
    End of Download    - EOD   - Selected  :            0
    End of Master Sync - EOMS              :            7
    Clear              - All               :            0
    Clear              - Selected          :            0
    Replay             - All               :            1
    Replay             - Selected          :            0
  Session                                  :            9            0
    Update                                 :            0            0
    Delete                                 :            0            0

Agent Activity Statistics:
  Active                                   :            1            0
  Deactive                                 :            0            0
  Connection Up                            :            1            0
  Connection Down                          :            0
  Peer Done                                :            8
...

RP/0/RSP0/CPU0:router# show ipv6 nd statistics

Service Attrib  Oper      Success# Failure# MinTime   MaxTime     AvgTime
                                              (usec)    (usec)      (usec)
AIB     IDB     Init      1         0         84110     84110       84110
AIB     IDB     Reg       1         0         295       295         295

CHKPT   IDB     Init      1         0         73        73          73
CHKPT   IDB     Update    47        0         23        100         27
CHKPT   NBR     Create    4         0         68        110         80
CHKPT   NBR     Del       2         0         150       1475        812
CHKPT   NBR     Init      1         0         232       232         232
CHKPT   NBR     Update    4         0         54        166         120

GSP     BATCH   Sent      4         32        0         2411        68
GSP     IDB     Init      1         0         5197      5197        5197
GSP     IDB     Reg       1         0         1816      1816        1816
GSP     IDB     Sent      24        0         13        104         15
GSP     MNODE   Sent      8         0         14        2482        322
GSP     PING    Recv      4         0         0         0           0
GSP     PING    Sent      4         0         76        1476        475
GSP     Ucast   Recv      5         0         0         0           0

IM      IDB     Create    10        0         0         0           0
IM      IDB     Init      2         0         670       4831        2750
IM      IDB     Reg       1         0         465       465         465
IM      MAC     Recv      8         0         0         0           0
IM      MAC     Reg       1         0         134       134         134
IM      MTU     Recv      8         0         0         0           0
IM      MTU     Reg       1         0         121       121         121
IM      VLAN    Recv      7         0         0         0           0
IM      VLAN    Reg       1         0         144       144         144
IM      MEMBER  Recv      24        0         0         0           0
IM      MEMBER  Reg       1         0         1603      1603        1603

LPTS    IDB     Add       8         0         2         174         25
LPTS    IDB     Init      1         0         4545      4545        4545
LPTS    IDB     Update    10        0         0         1           0

IPV6-MA BATCH   Recv      3         0         0         0           0

NETIO   IDB     Init      1         0         244787    244787      244787
NETIO   RA      Recv      14        0         0         0           0
NETIO   NA      Recv      4         0         0         0           0
NETIO   NS      Recv      4         0         0         0           0

STATS   IDB     Init      1         0         21410     21410       21410
STATS   IDB     Reg       8         0         2         19532       2449

 Session Redundancy Stats
 ---------------------------------------------

 Type                          Success   Error
 ---------------------------------------------
serg_init                        1         0
serg_shutdown                    0         0
serg_activate                    1         0
serg_active_txlist_add           0         0
serg_active_txlist_del           0         0
serg_active_txlist_encode        6         0
serg_active_txlist_clean         6         0
serg_active_replay               1         0
serg_active_cleanup              0         0
serg_standby_receive             0         0
serg_standby_sess_update         0         0
serg_standby_sess_delete         0         0
serg_standby_sess_nack           0         0
serg_standby_sess_mark           1         0
serg_standby_sess_sweep          1         0
serg_standby_cleanup             0         0
```

You have successfully configured and verified geo redundancy using session redundancy groups for IPv6 ND clients.

For information on managing SERGs, see Managing Session Redundancy Groups, on page 60.

Additional References

The following sections provide references related to implementing the Cisco IOS XR DHCP relay agent and DHCP snooping features.

Related Documents

Related Topic	Document Title
Cisco IOS XR DHCP commands	*DHCP Commands* module in the *IP Addresses and Services Command Reference for Cisco ASR 9000 Series Routers*
Getting started material	*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*
Information about user groups and task IDs	*Configuring AAA Services* module in the *System Security Configuration Guide for Cisco ASR 9000 Series Routers*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: https://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index

RFCs

RFC	Title
RFC 2131	*Dynamic Host Configuration Protocol*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport