

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 *IP Addresses and Services Command Reference for Cisco NCS 5500 Series, Cisco NCS 540 Series, and Cisco NCS 560 Series Routers* publication.

- Introduction to DHCP, on page 1
- Prerequisites for Configuring DHCP Relay Agent, on page 2
- Limitations for DHCP Relay Feature , on page 3
- DHCPv4 Relay Agent and Proxy Support for Segment Routing over IPv6 IPv4 L3VPN, on page 3
- How to Configure and Enable DHCP Relay Agent, on page 3
- Configure a DHCP Proxy Profile, on page 13
- DHCP Server, on page 13
- DHCP Client, on page 18
- DHCP Proxy Binding Table Reload Persistency, on page 20
- Jumbo Packet Handling for DHCPv6, on page 21
- Client ID change in DHCP IPv4 Server Profile, on page 22
- DHCP Snooping, on page 24
- Relay response on source interface, on page 30
- Configure DHCPv6 Relay Source Address, on page 31
- NTP-server option for DHCPv6, on page 33
- DHCPv6 relay subscriber ID, on page 35

Introduction to DHCP

The Dynamic Host Configuration Protocol (DHCP) is a client-server network protocol that automatically assigns IP addresses and other configuration parameters to devices on a TCP/IP network. It enables hosts to obtain necessary network settings dynamically, reducing manual configuration and simplifying network management.

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.

Table 1: Feature History Table

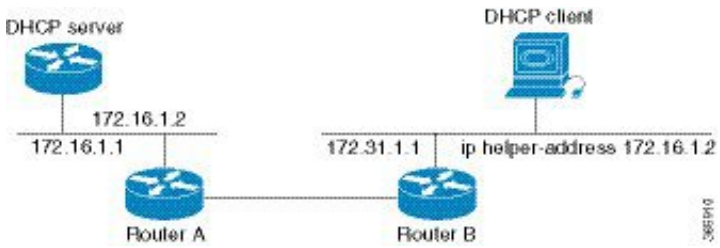
Feature Name	Release Information	Description
DHCP relay agent option 18	Release 25.4.1	<Place holder>

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.

The figure below 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



Prerequisites for Configuring DHCP Relay Agent

The following are the prerequisites 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

Limitations for DHCP Relay Feature

These are the limitations for implementing DHCP relay feature:

- The multicast addresses are not supported. The **helper-address** command in DHCP relay profile submode supports valid unicast IP address as the helper address.

Note Configuring the **helper-address** command directly (not using profile) under a interface (such as BVI interface) is not supported.

- Only interface-id and remote-id DHCP option code are added by a relay agent while forwarding the packet to a DHCP server.

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

DHCPv4 Relay Agent and Proxy Support for Segment Routing over IPv6 IPv4 L3VPN

DHCPv4 relay agent and proxy are supported on Segment Routing over IPv6 (SRv6) IPv4 L3VPN scenarios. See the How to Configure and Enable DHCP Relay Agent, on page 3 section for relay agent configuration. See the Configure a DHCP Proxy Profile, on page 13 section for proxy configuration.

For information about Segment Routing over IPv6, refer to the “Configure Segment Routing over IPv6 (SRv6)” chapter in the *Segment Routing Configuration Guide for Cisco NCS 5500 Series Routers*.

How to Configure and Enable DHCP Relay Agent

This section contains the following tasks:

Configuring and Enabling the DHCP Relay Agent

Configuration Example

```
Router# configure
/* Enters the global configuration mode */

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

Router(config-dhcpv4)# profile r1 relay
/* Enables DHCP relay profile */
```

```
Router(config-dhcpv4-relay-profile)# helper-address vrf A 10.10.10.1 giaddr 40.1.1.2
Router(config-dhcpv4-relay-profile)# broadcast-flag policy check
/* Configures VRF addresses for forwarding UDP broadcasts, including DHCP. */

Router(config-dhcpv4-relay-profile)# relay information option vpn
Router(config-dhcpv4-relay-profile)# relay information option vpn-mode rfc
/* 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 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)# exit
Router(config-dhcpv4)# interface BVI 1 relay profile r1
Router(config-dhcpv4)# commit
/* Configures DHCP relay on a BVI interface and commits the configuration */
```

Running Configuration

```
Router#show running-config
Tue May 23 10:56:14.463 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Tue May 23 10:56:08 2017 by annseque
!
dhcp ipv4
 vrf vrf1 relay profile client
 profile r1 relay
  helper-address vrf A 10.10.10.1 giaddr 40.1.1.2
  broadcast-flag policy check
  relay information option vpn
  relay information option vpn-mode rfc
  relay information option allow-untrusted
 !
```

Enabling DHCP Relay Agent on an Interface

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

Configuration Example

The DHCP relay agent is disabled by default.

```
router#configure

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

router(config-dhcpv4)#interface HundredGigE 0/2/0/2 relay profile client
/* Attaches a relay profile to an interface.
To disable the DHCP relay on the interface, use the 'no interface HundredGigE 0/2/0/2 none'
 command. */

router(config-dhcpv4-if)#commit
```

Running Configuration

```
Router#show running-config dhcp ipv4
dhcp ipv4
interface HundredGigE 0/2/0/2 relay profile client
!
```

Disabling DHCP Relay on an Interface

This task describes how to disable the DHCP relay on an interface by using the **no** keyword on the interface.

```
Router# configure terminal
Router(config)# dhcp ipv6
Router(config-dhcpv6)# no interface type name relay profile profile-name
Router(config-dhcpv6-if)# commit
```

Enabling DHCP Relay on a VRF

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

```
/CPU0:router# configure terminal
Router(config)# dhcp ipv6
Router(config-dhcpv6)# vrf vrf-name relay profile profile-name
Router(config-dhcpv6-if)# commit
```

Configure a DHCP Relay Profile with Multiple Helper Addresses

You can configure up to 16 helper IPv4 and IPv6 addresses for a DHCPv4 or DHCPv6 relay profile.

1. Enter the DHCPv4 or DHCPv6 configuration mode.

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

2. Configure the DHCPv4 or DHCPv6 relay profile.

   ```
   Router(config-dhcpv6)# profile helper relay
   ```

3. Configure helper addresses.

   Note You can configure up to 16 IPv4 and IPv6 addresses.

   ```
   Router(config-dhcpv6-relay-profile)# helper-address vrf default 2001:1:1::2
   ```

4. Confirm your configuration.

   ```
   Router(config-dhcpv6-relay-profile)# show configuration

   !! IOS XR Configuration 0.0.0
   dhcp ipv6
    profile helper relay
     helper-address vrf default 2001:1:1::2
    !
   !
   end
   ```

5. Commit your configuration.

```
Router(config-dhcpv6-relay-profile)# commit
```

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

```
Router# show dhcp ipv6 relay profile name helper
...
!
Profile: helper
Helper Addresses:
        2001:1:1::2, 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 the DHCPv6 relay helper address.

DHCP Relay Agent Notification for Prefix Delegation

DHCP 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 DHCP 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 IPv4 or 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 IPv4 or 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 Ipv4 or IPv6 routes are added when the relay agent relays a RELAY-REPLY packet, and the IPv4 or IPv6 routes are deleted when the prefix delegation lease time expires or the relay agent receives a release message. An Ipv4 or 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 IPv4 or IPv6 route on the routing table of the relay agent. This registered IPv4 or IPv6 address allows unicast reverse packet forwarding (uRPF) to work by allowing the router doing the reverse lookup to confirm that the Tpv4 or 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 DHCP Stateful Relay Agent for Prefix Delegation

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

Configuration Example

1. Configure a DHCP profile.

2. Configure the DHCP relay agent.
3. Enable IPv4 or IPv6 DHCP on an interface that acts as an IPv4 or IPv6 DHCP stateful relay agent.
4. Configure the profile name.



Note Prefix Delegation is supported with both DHCP Proxy and DHCP Relay configuration.

Configuration

```
/* Enter the global configuration mode and then enter the DHCPv6 configuration mode. */
Router# config
Router(config)# dhcp ipv6
Router(config-dhcpv6)#

/* Enter the proxy profile configuration mode and configure the DHCPv6 relay agent. */
Router(config-dhcpv6)# profile downstream proxy
Router(config-dhcpv6-profile)# helper-address 2001:db8::1 GigabitEthernet 0/1/0/1

/* Exits from the proxy profile configuration mode and enable IPv6 DHCP on an interface.
*/
Router(config-dhcpv6-profile)# exit
Router(config-dhcpv6-if)# interface GigabitEthernet 0/1/0/0 proxy

/* Configure a profile name. */

Router(config-dhcpv6-if)# profile downstream
Router(config-dhcpv6-if)# commit
```

Configuring Relay Agent Option 82 Per EFP

In forwarded BOOTREQUEST messages to a DHCP server, you can configure the relay agent to insert option 82 suboptions in the DHCP packet. Option 82 suboptions you can configure are Circuit ID and Remote ID. When the DHCP relay profile is attached to a Bridge Virtual Interface (BVI), you can assign the Option 82 circuit ID and Remote ID per EFP or per ingress Layer 2 interface. The relay agent sends the DHCP packet to the server that carries the packet's Option 82 Circuit ID or Remote ID. DHCP Relay Agent Option 82 provides security when DHCP is used to allocate network addresses. Thus, you can enable the DHCP relay agent to prevent DHCP client requests from untrusted sources.

Configuration Example

To configure a Layer 2 interface with relay agent option 82 suboptions, Circuit ID and Remote ID, use the following steps:

1. Configure DHCP for IPv4 and enter the DHCPv4 configuration submode.
2. Enable DHCP relay profile.
3. Configure VRF addresses for forwarding UDP broadcasts, including DHCP.
4. Insert the DHCP relay agent information option (option-82 field) in forwarded BOOTREQUEST messages to a DHCP server.
5. Enable DHCP for IPv4 on a BVI interface and attach the profile as the relay profile for the BVI interface.

6. Enable the DHCP relay agent to add Option 82 Circuit ID field in ascii or hex format per EFP to the DHCP packet.
7. Enable the DHCP relay agent to add Option 82 Remote ID field in ascii or hex format per EFP to the DHCP packet.

Configuration

```
/* Configure DHCP for IPv4 and enter the DHCPv4 configuration submode. */
Router# configure
Router(config)# dhcp ipv4

/* Enable DHCP relay profile. */
Router(config-dhcpv4)# profile bvi1_profile relay

/* Configure VRF addresses for forwarding UDP broadcasts, including DHCP. */
Router(config-dhcpv4-relay-profile)# helper-address 192.0.2.1

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

/* Enable DHCP for IPv4 on a BVI interface and attach the profile as the relay profile for
 the BVI interface. */
Router(config-dhcpv4)# interface BVI1 relay profile bvi1_profile

/* Enable the DHCP relay agent to add Option 82 Circuit ID field in ascii or hex format per
 EFP to the DHCP packet. */
Router(config-dhcpv4)# interface Bundle-Ether50.103 relay information option circuit-id
format-type ascii 110

/* Enable the DHCP relay agent to add Option 82 Remote ID field in ascii or hex format per
 EFP to the DHCP packet. */
Router(config-dhcpv4)# interface Bundle-Ether50.103 relay information option remote-id
format-type ascii 110
Router(config-dhcpv4)# commit
```

Running Configuration

```
Router# show running-config dhcp ipv4
Wed Jan 27 11:20:19.842 UTC
dhcp ipv4
 profile bvi1_profile relay
  helper-address vrf default 192.0.2.1
  relay information option
 !
 interface BVI1 relay profile bvi1_profile
 interface Bundle-Ether50.103 relay information option circuit-id format-type ascii 110
 interface Bundle-Ether50.103 relay information option remote-id format-type ascii 110
!
```

DHCPv6 Relay Over BVI for IANA Address Allocation

DHCPv6 Relay agents relay all packets that are coming from DHCPv6 clients over the access-interfaces towards external DHCPv6 servers to request IP addresses (::/128) through IANA allocation for the DHCPv6 clients. DHCPv6 Relay agents also receive response packets from the DHCPv6 servers and forward the packets towards DHCPv6 clients over BVI interfaces. DHCPv6 Relay agents acts as stateless, by default, for DHCPv6 clients by not maintaining any DHCPv6 binding and respective route entry for the allocated IP addresses.

You can enable a DHCPv6 client to get a particular IPv6 address assigned by the DHCPv6 server over a Bridge Virtual Interface (BVI) through Internet Assigned Numbers Authority (IANA) address allocation. Thereby, the DHCPv6 relay agent acts as a stateful relay agents and maintains DHCPv6 binding and respective route entry for the allocated IPv6 addresses.

Restrictions

- You can configure up to 500 client sessions over a BVI interface for DHCP relay.
- Each DHCPv6 relay profile can be configured with upto 8 DHCPv6 server addresses.

Configuration Example

To configure DHCPv6 Relay Over BVI for IANA Address Allocation, use the following steps.

1. Enter the interface configuration mode and configure a BVI interface.
2. Assign an IPv6 address to the BVI interface.
3. Route the L2 access interface to the L3 BVI interface of the relay agent.
4. Enter the DHCP IPv6 configuration mode and then create a DHCP IPv6 Stateful relay profile.
5. Attach the relay profile to a server address.
6. Configure a stateful relay agent by enabling route allocation through IANA.
7. Attach the BVI Interface to the DHCPv6 relay profile.

Configuration

```
/* Enter the interface configuration mode and configure a BVI interface. */
Router# configure
Router(config)# interface BVI1

Assign an IPv6 address to the BVI interface.
Router(config-if)# ipv6 address 2001:db8::2/64
Router(config-if)# commit
Router(config-if)# exit

/* Route the L2 access interface to the L3 BVI interface of the relay agent. */
Router(config)# l2vpn bridge group 1
Router(config-l2vpn-bg)# bridge-domain 1
Router(config-l2vpn-bg-bd)# interface hundredGigE 0/0/0/1.100
Router(config-l2vpn-bg-bd-ac)# commit
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# routed interface BVI1
Router(config-l2vpn-bg-bd)# exit
Router(config-l2vpn-bg)# exit
Router(config-l2vpn-bg)# exit
Router(config-l2vpn)# exit
Router(config)#

/* Enter the DHCP IPv6 configuration mode and then create a DHCP IPv6 Stateful relay profile.
 */
Router(config)# dhcp ipv6
Router(config-dhcpv6)# profile RELAY1 relay

/* Attach the relay profile to a server address. */
Router(config-dhcpv6-relay-profile)# helper-address vrf default 2001:DB8::1
```

```
/* Configure a stateful relay agent by enabling route allocation through IANA. */
Router(config-dhcpv6-relay-profile)# iana-route-add
Router(config-dhcpv6-relay-profile)# exit


/* Attach the BVI Interface to the DHCPv6 relay profile. */
Router(config-dhcpv6-relay-profile)# interface BVI1 relay profile RELAY1
Router(config-dhcpv6-relay-profile)# commit
```

Running Configuration

```
Router# show running configuration
interface BVI1
 ipv6 address 2001:db8::2/64
!
l2vpn
 bridge group 1
  bridge-domain 1
   interface HundredGigE0/0/0/1.100
   !
   routed interface BVI1
   !
  !
 !
!
dhcp ipv6
 profile RELAY1 relay
  helper-address vrf default 2001:db8::1
  iana-route-add
  !
 interface BVI1 relay profile RELAY1
!
```

Verification

Use the following command to verify that more than one DHCP client is bridged over BVI:

```
Router# show dhcp ipv6 relay binding
Thu Nov 21 05:48:38.463 UTC

Summary:
Total number of clients: 500

IPv6 Address: 2000::418f/128 (BVI31)
    Client DUID: 000100015dcf28de001094003295
    MAC Address: 0010.9400.3295
    IAID: 0x0
    VRF: default
    Lifetime: 600 secs (00:10:00)
    Expiration: 533 secs (00:08:53)
    L2Intf AC: Bundle-Ether3.1
    SERG State: NONE
    SERG Intf State: SERG-NONE
IPv6 Address: 2000::4190/128 (BVI31)
    Client DUID: 000100015dcf28de001094003296
    MAC Address: 0010.9400.3296
    IAID: 0x0
```

```
    VRF: default
    Lifetime: 600 secs (00:10:00)
    Expiration: 531 secs (00:08:51)
    L2Intf AC: Bundle-Ether3.1
    SERG State: NONE
    SERG Intf State: SERG-NONE
IPv6 Address: 2000::4191/128 (BVI31)
    Client DUID: 000100015dcf28de001094003297
    MAC Address: 0010.9400.3297
    IAID: 0x0
    VRF: default
    Lifetime: 600 secs (00:10:00)
    Expiration: 448 secs (00:07:28)
    L2Intf AC: Bundle-Ether3.1
    SERG State: NONE
    SERG Intf State: SERG-NONE
IPv6 Address: 2000::4192/128 (BVI31)
    Client DUID: 000100015dcf28de001094003298
    MAC Address: 0010.9400.3298
    IAID: 0x0
    VRF: default
    Lifetime: 600 secs (00:10:00)
    Expiration: 439 secs (00:07:19)
    L2Intf AC: Bundle-Ether3.1
    SERG State: NONE
    SERG Intf State: SERG-NONE
```

Use the following command to verify that unique IPv6 address is assigned to a client due to IANA allocation:

```
Router# show route ipv6
Mon Oct 21 06:16:43.617 UTC

Codes: C - connected, S - static, R - RIP, B - BGP, (>) - Diversion path
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR, l - LISP
       A - access/subscriber, a - Application route
       M - mobile route, r - RPL, t - Traffic Engineering, (!) - FRR Backup path

Gateway of last resort is not set

A    2000::/64
      [1/0] via fe80::1, 00:00:37, BVI700
A    2000::1/128
      [1/0] via fe80::210:94ff:fe00:8, 00:00:12, BVI700
C    2007:3019::/64 is directly connected,
      00:00:37, Loopback1
L    2007:3019::1/128 is directly connected,
      00:00:37, Loopback1
C    7001:6018::/64 is directly connected,
      00:00:37, BVI700
L    7001:6018::1/128 is directly connected,
      00:00:37, BVI700
C    7001:6019::/64 is directly connected,
      00:00:37, TenGigE0/0/0/2.2
L    7001:6019::1/128 is directly connected,
      00:00:37, TenGigE0/0/0/2.2
```

DHCP Relay Profile: Example

The following example shows how to configure the DHCP 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
 !
!
```

Configure 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.

Configuration Example

1. Enter DHCP IPv4 or DHCP IPv6 profile proxy submode.
2. Forward UDP broadcasts, including DHCP.



Note

- 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.

Configuration

```
/* Enter the DHCP IPv4 profile proxy submode. */
Router(config)# dhcp ipv4
Router(config-dhcpv4)# profile client proxy

/* Forward UDP broadcastrs, including DHCP */
Router(config-dhcpv4-proxy-profile)# helper-address vrf vrf1 foo 10.10.1.1
Router(config-dhcpv4-proxy-profile)# commit
```

DHCP Server

A 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 servers 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 servers 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.

When the DHCP server receives an address assignment request, it assigns the IP addresses from groups of IP addresses for DHCP in Distributed Address Pools (DAPS). The IP address used by the DHCP server to complete such requests is automatically excluded from the DAPS so that the DHCP server can safely assume that all the IP addresses available for its use in the DAPS are free.

DHCP Service-based Mode Selection

As part of DHCP 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.

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

The DHCPv6 server-profile-class submode supports configuring DHCP options except few (0, 12, 50, 52, 53, 54, 58, 59, 61, 82, and 255).

```
dhcp ipv6
profile DHCP_BASE base
  match option 60 41424344 profile DHCPv6_PROXY proxy
  match option 60 41424355 profile DHCPv6_SERVER server
  default profile DEFAULT_PROFILE server
  relay information authenticate inserted
   !
profile DHCPv6_PROXY proxy
  helper-address vrf default 10.10.10.1 giaddr 0.0.0.0
!
profile DHCPv6_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
```

Configuring DHCP Server Profile

You can configure routers with DHCPv4 or DHCPv6 server profile.

Perform this task to configure the DHCPv6 server profile.

```
Router# configure
Router(config)# dhcp ipv6
Router(config-dhcpv6)# profile profile-name server
Router(config-dhcpv6-server-profile)# bootfile boot-file-name
Router(config-dhcpv6-server-profile)# broadcast-flag policy unicast-always
Router(config-dhcpv6-server-profile)# class class-name
Router(config-dhcpv6-server-profile-class)# exit
Router(config-dhcpv6-server-profile)# default-router address1 address2 ... address8
Router(config-dhcpv6-server-profile)# lease {infinite | days minutes seconds }
Router(config-dhcpv6-server-profile)# limit lease {per-circuit-id | per-interface|
per-remote-id} value
Router(config-dhcpv6-server-profile)# netbios-name server address1 address2 ... address8
Router(config-dhcpv6-server-profile)# netbios-node-type {number |b-node|h-node |m-node
|p-node}
Router(config-dhcpv6-server-profile)# option option-code {ascii string | hex string |ip
address}
Router(config-dhcpv6-server-profile)# pool pool-name
Router(config-dhcpv6-server-profile)# requested-ip-address-check disable
Router(config-dhcpv6-server-profile)# commit
```

Configuring Multiple Classes with a Pool

Perform this task to configure multiple classes with a pool.

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# dhcp ipv6
RP/0/RSP0/CPU0:router(config-dhcpv6)# profile profile-name server
RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# pool pool-name
RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# class class-name
RP/0/RSP0/CPU0:router(config-dhcpv6-server-class)# pool pool-name
RP/0/RSP0/CPU0:router(config-dhcpv6-server-class)# match option option [ sub-option
sub-option] [ ascii asciiString | hex hexString ]
RP/0/RSP0/CPU0:router(config-dhcpv6-server-class)# exit
RP/0/RSP0/CPU0:router(config-dhcpv6-server-profile)# class class-name
RP/0/RSP0/CPU0:router(config-dhcpv6-server-class)# pool pool-name
RP/0/RSP0/CPU0:router(config-dhcpv6-server-class)# match vrf vrf-name
RP/0/RSP0/CPU0:router(config-dhcpv6-server-class)# commit
```

Configuring a Server Profile DAPS with Class Match Option

This section discusses configuring a server profile DAPS with class match option.

Configuration Example

```
router#configure

router(config)#dhcp ipv4
/* The 'dhcp ipv6' command configures DHCP for IPv6 and enters the DHCPv6 configuration
submode. */

router(config-dhcpv4)#profile ISP1 server
/* Enters the server profile configuration mode. */

router(config-dhcpv4-server-profile)#pool ISP1_POOL
/* Configures the DAPS pool name. */

router(config-dhcpv4-server-profile)#class ISP1_CLASS
/* Creates and enters server profile class configuration submode. */

router(config-dhcpv4-server-profile-class)#pool ISP1_CLASS_POOL
/* Configures the pool name. */

router(config-dhcpv4-server-profile-class)#match option 60 hex PXEClient_1
/* DHCP server selects a pool from a class by matching options in the received DISCOVER
packet with the match option. */

router(config-dhcpv4-server-profile-class)#exit

router(config-dhcpv4-server-profile)#exit

router(config-dhcpv4)#profile ISP2 server
/* Enters the server profile configuration mode. */

router(config-dhcpv4-server-profile)#dns-server 10.20.3.4
/* Configures the name of the DNS server or the IP address. */

router(config-dhcpv4-server-profile)#pool ISP2_POOL
/* Configures the pool name. */

router(config-dhcpv4-server-profile)#class ISP2_CLASS
/* Creates and enters the server profile class. */
```

```
router(config-dhcpv4-server-profile-class)#pool ISP2_CLASS_POOL
/* Configures the pool name. */

router(config-dhcpv4-server-profile-class)#match option 60 hex PXEClient_2
/* DHCP server selects a pool from a class by matching options in the received DISCOVER
packet with the match option. */

router(config-dhcpv4-server-profile-class)#exit

router(config-dhcpv4-server-profile)#exit

router(config-dhcpv4)#commit
```

Running Configuration

```
Router#show running-config dhcp ipv4
dhcp ipv4
profile ISP1 server
pool ISP1_POOL
class ISP1_CLASS
pool ISP1_CLASS_POOL
match option 60 hex PXEClient_1
exit
exit
profile ISP2 server
dns-server 10.20.3.4
pool ISP2_POOL
class ISP2_CLASS
pool ISP2_CLASS_POOL
match option 60 hex PXEClient_2
exit
exit
!
```

Configuring Server Profile without DAPS Pool Match Option

This section discusses configuring a server profile without DAPS pool match option.

Configuration Example

```
router#configure

router(config)#dhcp ipv4
/* The 'dhcp ipv6' command configures DHCP for IPv6 and enters the DHCPv6 configuration
submode. */

router(config-dhcpv4)#profile ISP1 server
/* Enters the server profile configuration mode. */

router(config-dhcpv4-server-profile)#dns-server ISP1.com
/* Configures the name of the DNS server or IP address. */

router(config-dhcpv4-server-profile)#exit

router(config-dhcpv4)#profile ISP2 server
/* Enters the server profile configuration mode. */

router(config-dhcpv4-server-profile)#dns-server ISP2.com
```

```
/* Configures the name of the DNS server or IP address. */

router(config-dhcpv4-server-profile)#exit

router(config-dhcpv4)#commit
```

Running Configuration

```
Router#show running-config dhcp ipv4
dhcp ipv4
 profile ISP1 server
  dns-server ISP1.com

  exit
 profile ISP2 server
  dns-server ISP2.com

  exit
!
```

Configuring an Address Pool for Each ISP on DAPS

This section discusses configuring an address pool for each ISP on DAPS.

Configuration Example

```
router#configure

router(config)#pool vrf ISP_1 ipv4 ISP1_POOL
/* Configures an IPv4 pool for the specifed VRF or all VRFs. Use the 'ipv6' keyword for
IPv6 pool. */

router(config-pool-ipv4)#network 10.10.10.0
/* Specifies network for allocation. */

router(config-pool-ipv4)#exit

router(config)#pool vrf ISP_2 ipv4 ISP2_POOL
/* Configures an IPv4 pool for the specifed VRF or all VRFs. */

router(config-pool-ipv4)#network 10.20.20.0
/* Specifies network for allocation. */

router(config-pool-ipv4)#exit

router(config-dhcpv4)#commit
```

Running Configuration

```
Router#show running-config pool
pool vrf ISP_1 ipv4 ISP1_POOL
 network 10.10.10.0
 exit
pool vrf ISP_2 ipv4 ISP2_POOL
 network 10.20.20.0
!
```

DHCP Client

The Dynamic Host Configuration Protocol (DHCP) client functionality enables the router interfaces to dynamically acquire the IPv4 or DHCPv4 or DHCPv6 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

- DHCPv4 or DHCPv6 client can be enabled only on management interfaces.
- Either DHCPv4, DHCPv6, static IPv4, or static IPv6 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
```

You can configure DHCPv6 client on BVI interfaces. You can configure different DHCPv6 client options to differentiate between clients as required. The different DHCPv6 client options are also configured to differentiate how a DHCPv6 client communicates with a DHCPv6 server. The different DHCPv6 client options that can be configured are:

- **DUID:** If the DUID DHCPv6 client option is configured on an interface, DHCPv6 client communicates with the DHCPv6 server through the link layer address.
- **Rapid Commit:** If the Rapid Commit DHCPv6 client option is configured on an interface, DHCPv6 client can obtain configuration parameters from the DHCPv6 server through a rapid two-step exchange (solicit and reply) instead of the default four-step exchange (solicit, advertise, request, and reply).
- **DHCP Options:** The various other DHCPv6 options that can be configured on a DHCPv6 client are:
  - **Option 15:** Option 15 is also known as the User Class option and it is used by a DHCPv6 client to identify the type or category of users or applications it represents.
  - **Option 16:** Option 16 is also known as the Vendor ID option and it is used by a DHCPv6 a client to identify the vendor that manufactured the hardware on which the client is running.
  - **Option 23:** Option 23 is also known as the Domain name Server (DNS) option provides a list of one or more IPv6 addresses of DNS recursive name servers to which a client's DNS resolver can send DNS queries.
  - **Option 24:** Option 24 is also known as the Domain List option and it specifies the domain search list that the client uses to resolve hostnames with the DNS.

- **DHCP Timers:** This option is used to set different timer value for DHCP client configurations. The various DHCP timer options are:
  - **Release-timeout:** It is used to set retransmission timeout value for the initial release message.
  - **Req-max-rt:** It is used to set the maximum retransmission timeout value for the request message.
  - **Req-timeout:** It is used to set the initial request timeout value of the request message.
  - **Sol-max-delay:** It is used to set the maximum delay time of the first solicit message.
  - **Sol-max-rt:** It is used to set the maximum solicit retransmission time.
  - **Sol-time-out:** It is used to set the intial timeout value of the solicit message.

Configuration Example

You can use the following steps to configure DHCPv6 client options on a BVI interface:

1. Enter the interface configuration mode, and then configure a BVI interface.
2. Enter the DHCPv6 client configuration mode, and then enter the DHCPv6 client option configuration mode.
3. Configure a DHCPv6 client option.

Configuration

```
/* Enter the interface configuration mode, and then configure a BVI interface. */
Router# configure
Router(config)# interface BVI 10

/* Enter the DHCPv6 client configuration mode, and then enter the DHCPv6 client option
configuration mode. */
Router(config-if)# ipv6 address dhcp-client-options

/* Configure a DHCP client option. */
Router(config-dhcpv6-client)# timers release-timeout 3
Router(config-dhcpv6-client)# commit
```

Verification

To verify the DHCPv6 client options Rapid Commit, DUID, User Class, Vendor ID, Domain name Server, and Domain List; use the `show dhcp ipv6 client BVI1 detail` command:

```
Router# show dhcp ipv6 client BVI1 detail
Tue Apr  7 15:13:19.272 IST

-------------------------------------------------------
Client Interface name : MgmtEth0/0/CPU0/1
Client Interface handle : 0x4040
Client MACAddr : 02f0.2b39.44be
Client State : BOUND
Client Link Local Address : fe80::f0:2bff:fe39:44be
Client IPv6 Address (Dhcp) : 600:1::12
Lease Remaining (in secs) : 74
DUID : 0003000102f02b3944be

Client Configuration
Timers
```

```
SOL_MAX_DELAY : 1 secs (00:00:01)
SOL_TIMEOUT : 1 secs (00:00:01)
SOL_MAX_RT : 120 secs (00:02:00)
REQ_TIMEOUT : 1 secs (00:00:01)
REQ_MAX_RT : 30 secs (00:00:30)
REL_TIMEOUT : 1 secs (00:00:01)

Options
RAPID-COMMIT : True
USER-CLASS : ciscoupnnp
VENDOR-CLASS : vendor
DNS-SERVERS : True
DOMAIN-LIST : True

DUID Type : DUID_LL

Server Information
Server Address : fe80::d2:a1ff:feb2:3b9f
Preference : 0
DUID : 000300010206826e2e00
Status : SUCCESS
IA-NA
Status : SUCCESS
IAID : 0x40400001
T1 : 60 secs (00:01:00)
T2 : 96 secs (00:01:36)
IA-ADDR
IA NA Address : 600:1::12
Preferred Time : 120 secs (00:02:00)
Valid Time : 120 secs (00:02:00)
Flags : 0x0
```

Associated Commands

- ipv6 address dhcp-client-options
- clear dhcp ipv6 client
- show dhcp ipv6 client
- show tech-support dhcp ipv6 client

DHCP 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 DHCPv4 or 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 DHCP Relay Binding Database Write to System Persistent Memory

Perform this task to configure the DHCP relay binding database write to the system persistent memory. This helps to recover the DHCP relay binding table after a system reload. The file names used for a full persistent file write are dhcpv4_srpb_{nodeid}_odd or dhcpv6_srpb_{nodeid}_odd and dhcpv4_srpb_{nodeid}_even or 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.

```
Router# configure
Router(config)# dhcp ipv6
Router(config-dhcpv6)# database [relay] [full-write-interval full-write-interval]
[incremental-write-interval incremental-write-interval]
Router(config-dhcpv6)# commit
```

Jumbo Packet Handling for DHCPv6

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Jumbo packet handling for DHCPv6	Release 7.4.1	This release introduces the **handle-jumbo-packet** configuration command under the dhcp ipv6 mode. This command enables processing of incoming DHCPv6 packets greater than 1280 bytes and upto 12,800 bytes in size. Prior to this release, the router discarded incoming DHCPv6 packets greater than 1280 bytes. The newly introduced command is: • handle-jumbo-packet

By default, the router allows incoming DHCPv6 packets with maximum size of 1280 bytes and drops any packet that is larger. If you configure the **handle-jumbo-packet** command under **dhcp ipv6** configuration mode, then the router allows incoming DHCPv6 packets upto 12,800 bytes in size. The router drops incoming packets larger than 12,800 bytes. You can configure this command for all modes of DHCPv6, that is, server, proxy and relay, as well as for both BNG (Broadband Network Gateway) and non-BNG networks.

Configuration Example

This example shows you how to configure **handle-jumbo-packet**:

```
Router# config
Router(config)# dhcp ipv6
Router(config-dhcpv6)# handle-jumbo-packet
Router(config-dhcpv6)# commit
```

Client ID change in DHCP IPv4 Server Profile

Table 3: Feature History Table

Feature Name	Release Information	Description
Client ID change in DHCP IPv4 Server Profile	Release 7.5.3	With this release, we've introduced the allow-client-id-change configuration under DHCP IPv4 mode. This option ensures that the client machines have only one binding with DHCP IPv4 server constantly. In scenarios where a client with active binding and valid lease time sends a discover message with a new client id, and the DHCP server approves such requests assuming it as a new client due to different client IDs. This results in multiple bindings for the same client, making the older binding redundant. This feature avoids wastage of DHCP resources due to such multiple bindings.

Starting with Cisco IOS XR Release 7.5.3, you can utilize the allow-client-id-change configuration for DHCP IPv4. When you enable this configuration, the DHCP server ensures that each client that is associated with the server has only one binding. When a client with active binding and valid lease time in the DHCP server sends a discover message with a new client id the DHCP server searches its database for the client by excluding the client id in the search criteria. In the search response, if an existing client is found in the DHCP server database, then that client entry is erased from the DHCP database. When the client does not receive a response for the discovery request, it retries with a discovery message, and the DHCP server does a database lookup again. Due to the previous clean-up, it will not find any matching node and assigns a new binding to the client.

Restrictions

- Configuration for allowing client-id change is available for DHCP IPv4 server only.

Configuration

To configure the DHCP IPv4 Server Profile to allow Client ID change, do the following:

```
Router# configure

Router(config)#dhcp ipv4
```

```
/* The 'dhcp ipv4' command configures DHCP for IPv4 and enters the DHCPv4 configuration
submode. */

Router(config-dhcpv4)#profile ISP1 server
/* Enters the server profile configuration mode. */

Router(config-dhcpv4-server-profile)#class ISP1_CLASS
/* Configures the lease for an IP address assigned from the pool. */

Router(config-dhcpv4-server-profile-class)#lease 0 1 0
/* Configures lease period for the server profile. */

Router(config-dhcpv4-server-profile-class)#pool ISP1_CLASS_POOL
/* Configures the pool name. */

Router(config-dhcpv4-server-profile-class)# delete-binding-on-discover disable
/* (Optional) When client id changes, instead of deleting the old binding, this configuration
 ensures that the old binding is reassigned to the new client id. */

Router(config-dhcpv4-server-profile-class)#exit

Router(config-dhcpv4-server-profile)# interface HundredGigE0/0/0/0 server profile server_v4
/* Configures interface for the DHCP IOv4 server profile */

Router(config-dhcpv4-server-profile)# allow-client-id-change
/* Configures allowing client id change */

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

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

Running Configuration

```
Router#show running-config dhcp ipv4
dhcp ipv4
 profile ISP2 server
  lease 0 1 0
  pool ISP2_POOL
  delete-binding-on-discover disable
 !
 interface HundredGigE0/0/0/0 server profile server_v4
 allow-client-id-change
!
```

DHCP Snooping

Table 4: Feature History Table

Feature Name	Release Information	Description
DHCP Snooping for Layer 2 networks	Release 7.9.1	With this feature, you can secure your DHCP infrastructure for Bridge Domains. DHCP Snooping operates in the Layer 2 network and prevents unauthorized DHCP servers from accessing your network. This feature mitigates the security risks due to denial-of-service from rogue DHCP servers, which disrupt networks as they compete with legitimate DHCP servers that configure hosts on the network for communication. You can use the Cisco-IOS-XR-ipv4-dhcpd-oper.yang, Cisco-IOS-XR-l2vpn-oper.yang, and Cisco-IOS-XR-um-dhcp-ipv4-cfg.yang (see GitHub, YANG Data Models Navigator) data models to configure this feature.

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.

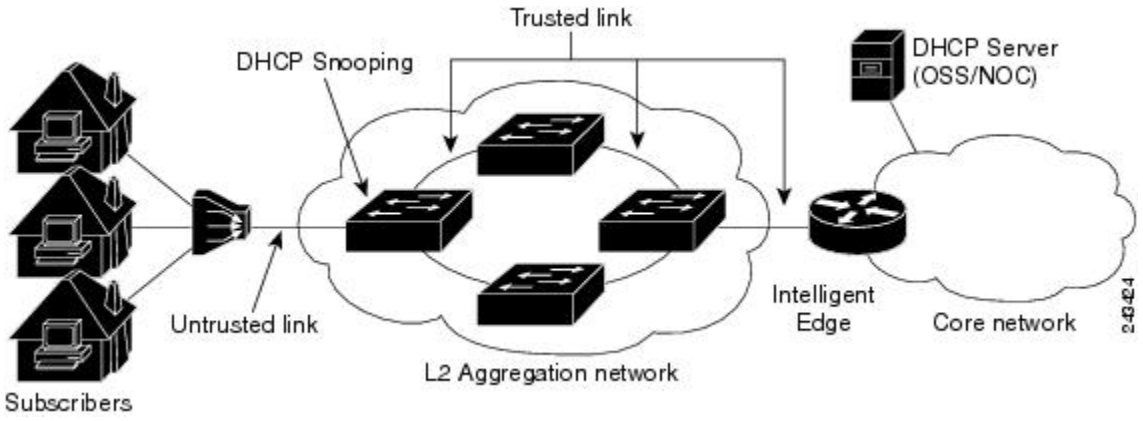
The central concept for DHCP snooping is that of trusted and untrusted links. A trusted link is one providing secure access for traffic on that link. On an untrusted link, subscriber identity and subscriber traffic can't be determined. DHCP snooping runs on untrusted links to provide subscriber identity by dynamically assigning IP address to subscriber devices on a network so it can communicate using IP. The figure *DHCP Snooping in an Aggregation Network* 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 The Layer 2 bridge feature works on Layer 2 Bridge Domaim. Here, the Ethernet Flow Points (EFP) receiving the Broadcast, Unknown-unicast and Multicast (BUM) traffic forwards it to rest of EFPs on the same bridge domain except to the EFP receiving the traffic initially.

Note The router does not forwards any DHCP packets to an EFP which has another DHCP client.

Note The NCS 5500 Series Routers supports only DHCP IPv4 Snooping for Layer 2 networks.

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 isn't tracked and the client isn't 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 client'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 aren't 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. Usually, 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 few 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, 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 don't require DHCP snooping, all should have the none profile assigned to them to disable DHCP snooping on those ports.

Prerequisites for Configuring DHCP Snooping

- 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 NCS 5500 Series Router running Cisco IOS XR software.
- A configured and running DHCP client and DHCP server.

Enabling DHCP Snooping in a Bridge Domain

The following configuration creates two ports, a client-facing port and a server-facing port. Here, an untrusted DHCP snooping profile is assigned to the client bridge port and trusted DHCP snooping profile is assigned to the server bridge port. And, an untrusted DHCP snooping profile is assigned to the bridge domain and trusted DHCP snooping profiles are assigned to server bridge ports.

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

/*   Configures an untrusted DHCP snooping profile for the client port   */
RP/0/RSP0/CPU0:router(config-dhcpv4)# profile untrustedClientProfile snoop

RP/0/RSP0/CPU0:router(config-dhcpv4)# commit
RP/0/RSP0/CPU0:router(config-dhcpv4)# exit  */


/*   Enables DHCP for IPv4 and enters DHCP IPv4 profile configuration mode
RP/0/RSP0/CPU0:router(config)# dhcp ipv4   */

/*   Configures a trusted DHCP snooping profile for the server port   */
RP/0/RSP0/CPU0:router(config-dhcpv4)# profile trustedServerProfile snoop

/*   Configures a DHCP snoop profile to be trusted
RP/0/RSP0/CPU0:router(config-dhcv4)# trusted   */
RP/0/RSP0/CPU0:router(config-dhcv4)# commit
RP/0/RSP0/CPU0:router(config-dhcv4)# exit


RP/0/RSP0/CPU0:router(config)# l2vpn

/*   Creates a bridge group to contain bridge domains and enters l2vpn bridge group
configuration submode   */
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group ccc

/*   Establishes a bridge domain   */
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ddd

/*   Identifies an interface   */
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface TenGigE0/2/0/4/1.1

/*   Attaches a trusted DHCP snoop profile to the bridge domain   */
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# dhcp ipv4 snoop profile trustedServerProfile
```

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# commit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit


/*   Identifies an interface   */
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface HundredGigE0/1/0/8.1

/*   Attaches a trusted DHCP snoop profile to the bridge domain   */
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# dhcp ipv4 snoop profile UnTrustedServerProfile

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# commit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# exit
RP/0/RSP0/CPU0:router(config-l2vpn)# exit
```

Running Configuration

```
RP/0/RSP0/CPU0:router(config)# show running config
dhcp ipv4
 profile UnTrustedClientProfile snoop
!
dhcp ipv4
 profile trustedServerProfile snoop
 trusted
!
l2vpn
 bridge group ccc
  bridge-domain ddd
   interface TenGigE0/2/0/4/1.1
   dhcp ipv4 snoop profile trustedServerProfile
 !
   interface HundredGigE0/1/0/8.1
  dhcp ipv4 snoop profile UnTrustedServerProfile
!
 !
!
```

Verification

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding detail location gigabitethernet 0/1/0/0
Bridge-domain name: bg1:bd1, id: 0, state: up
 MAC learning: enabled
 Flooding:
   Broadcast & Multicast: enabled
   Unknown unicast: enabled
 MAC aging time: 300 s, Type: inactivity
 MAC limit: 4000, Action: none, Notification: syslog
 MAC limit reached: no
 Security: disabled
 DHCPv4 snooping: enabled
IGMP snooping: disabled, flooding: disabled
 Bridge MTU: 1500 bytes
 Number of bridge ports: 1
 Number of MAC addresses: 0
 Multi-spanning tree instance: 0

  GigabitEthernet0/1/0/0, state: oper up
    Number of MAC: 0
    Statistics:
      packets: received 0, sent 0
      bytes: received 0, sent 0
    Storm control drop counters:
```

```
      packets: broadcast 0, multicast 0, unknown unicast 0
      bytes: broadcast 0, multicast 0, unknown unicast 0

RP/0/RP0/CPU0:ios# show dhcp ipv4 snoop profile name trustedServerProfile
DHCP Ipv4 Snoop Profile trustedServerProfile:

        Information Option:                 Disabled
        Information Option Allow Untrusted: Disabled
        Information Option Policy:          Replace
        Trusted:                            Enabled

        Bridge References:
        Interface References:
                TenGigE0/2/0/4/1.1

RP/0/RP0/CPU0:ios# show dhcp ipv4 snoop profile name UnTrustedServerProfile
DHCP Ipv4 Snoop Profile UnTrustedServerProfile:

        Information Option:                 Disabled
        Information Option Allow Untrusted: Disabled
        Information Option Policy:          Replace
        Trusted:                            Disabled

        Bridge References:
        Interface References:
                HundredGigE0/1/0/8.1
```

Enabling DHCP Snooping on a Specific Bridge Port

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

Configuration

```
RP/0/RSP0/CPU0:router(config)# dhcp ipv4
/*   Enters DHCP IPv4 profile configuration submode   */

RP/0/RSP0/CPU0:router(config-dhcpv4)# profile untrustedClientProfile snoop
/*   Configures an untrusted DHCP snooping profile for the client port   */

RP/0/RSP0/CPU0:router(config-dhcpv4)# exit
/*   Exits DHCP IPv4 profile configuration mode   */

RP/0/RSP0/CPU0:router(config)# dhcp ipv4
/*   Enables DHCP for IPv4 and enters DHCP IPv4 profile configuration mode   */

RP/0/RSP0/CPU0:router(config-dhcpv4)# profile trustedServerProfile snoop
/*   Configures a trusted DHCP snooping profile for the server port   */

RP/0/RSP0/CPU0:router(config-dhcv4)# trusted
/*   Configures a DHCP snoop profile to be trusted   */

RP/0/RSP0/CPU0:router(config-dhcv4)# exit
/*   Exits DHCP IPv4 profile configuration mode   */

RP/0/RSP0/CPU0:router(config)# l2vpn
/*   Enters l2vpn configuration mode   */

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   */

RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ddd
/*   Establishes a bridge domain   */
```

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gigabitethernet 0/1/0/0
/*   Identifies an interface   */

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   */

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# interface gigabitethernet 0/1/0/1
/*   Identifies an interface   */

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   */

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# commit

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit
/*   Exits the l2vpn bridge group bridge-domain interface configuration submode   */

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# commit

RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# exit
/*   Exits the l2vpn bridge group bridge-domain configuration submode   */
```

Running Configuration

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

 profile untrustedClientProfile snoop

!
dhcp ipv4

 profile trustedServerProfile snoop

  trusted

!
l2vpn

 bridge group group-name

  bridge-domain bridge-domain-name

   interface gigabitethernet 0/1/0/0
    dhcp ipv4 snoop profile untrustedClientProfile
   interface gigabitethernet 0/1/0/1
    dhcp ipv4 snoop profile trustedServerProfile
  !
 !
commit
```

Verification

```
RP/0/RSP0/CPU0:router# show l2vpn forwarding detail location gigabitethernet 0/1/0/0
Bridge-domain name: bg1:bd1, id: 0, state: up
 MAC learning: enabled
 Flooding:
   Broadcast & Multicast: enabled
   Unknown unicast: enabled
 MAC aging time: 300 s, Type: inactivity
 MAC limit: 4000, Action: none, Notification: syslog
 MAC limit reached: no
 Security: disabled
 DHCPv4 snooping: enabled
IGMP snooping: disabled, flooding: disabled
```

```
Bridge MTU: 1500 bytes
Number of bridge ports: 1
Number of MAC addresses: 0
Multi-spanning tree instance: 0

 GigabitEthernet0/1/0/0, state: oper up
   Number of MAC: 0
   Statistics:
     packets: received 0, sent 0
     bytes: received 0, sent 0
   Storm control drop counters:
     packets: broadcast 0, multicast 0, unknown unicast 0
     bytes: broadcast 0, multicast 0, unknown unicast 0
```

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 5: Feature History Table

Feature Name	Release Information	Feature Description
Configure DHCPv6 relay source address	Release 25.1.1	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: Compatibility; Native] 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. Earlier, 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 **source-interface**.

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
```

NTP-server option for DHCPv6

The NTP-server option is a functionality within the DHCPv6 (Dynamic Host Configuration Protocol for IPv6) system that allows the DHCPv6 server to provide the IPv6 address of a Network Time Protocol (NTP) server to a client device.

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
NTP-server option for DHCPv6	Release 25.3.1	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] You can now configure NTP-server option in the DHCPv6 server profile. The DHCPv6 server provides NTP-server information to a client device when the client requests this information and a user configures it. When configured, this functionality simplifies user setup and ensures accurate time synchronization across the network. The feature introduces these changes: **CLI**: • The **ntp-server** *ntp-address* command is introduced. **YANG Data Model**: Cisco-IOS-XR-ipv6-new-dhcpv6d-cfg.yang (see GitHub, YANG Data Models Navigator)

Benefits of NTP-server option

These are the benefits of configuring the NTP-server option:

- Simplified client configuration: Devices obtain NTP server information directly from the DHCPv6 server. This removes the need for manual configuration on each client.
- Consistent operation: The feature functions consistently across all platforms that run the Cisco IOS XR software.

Configuration guidelines and restrictions for NTP-server option

Configuration guidelines for NTP-server option

These guidelines apply when you configure NTP-server option:

- You must validate the reachability and correctness of the NTP-server. The router does not perform these checks.
- You can disable the feature by using the **no** version of the configuration command. For example, use **no ntp-server** *ntp-address*.

Restrictions for NTP-server option

These restrictions apply for NTP-server configuration:

- The NTP_SUBOPTION_MC_ADDR sub-option is not supported.
- The NTP_SUBOPTION_SRV_FQDN sub-option is not supported.

Configure NTP-server option

Perform these steps on the router to configure NTP-server option:

Procedure

Step 1 Enter the DHCPv6 server profile configuration mode.

Example:

```
Router# configure terminal
Router(config)# dhcp ipv6
Router(config-dhcpv6)# profile SERVE server
```

Step 2 Use the **ntp-server** command to specify the IPv6 address of the NTP server.

Example:

```
Router(config-dhcpv6-server-profile)# lease 0 1 0
Router(config-dhcpv6-relay-profile)# ntp-server 5::5
Router(config-dhcpv6-relay-profile)# !
```

Step 3 Save your changes.

Example:

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

During configuration commit, the router validates that you provide a valid IPv6 address.

Step 4 Verify the configuration.

Example:

```
Router#show dhcp ipv6 server profile name SERVE
Fri Jan 24 06:34:45.004 UTC

Profile: SERVE
DNS Addresses: 20::20
Domain Name: cisco.com
Client Lease Time: 3600 secs (01:00:00)
Framed Address Pool: IPV6_ADDR
Delegated Prefix Pool: PD_POOL
Session-limit-per-vlan: Disabled
Rapid Commit: Enabled
Dynamic-Relay-Mac-addr: Disabled
NTP-Server-addr: 10::10
Interface References:
    TenGigE0/0/0/4
    TenGigE0/0/0/3
```

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 7: 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
```