



# Configuring Virtual Private Networks and Subnet Allocation

---

This chapter describes how to configure the Cisco CNS Network Registrar DHCP server to support virtual private networks (VPNs) and subnet allocation for on-demand address pools.

Configuring VPNs involves an adjustment to the usual DHCP host IP address designation. VPNs use private address spaces that might not be unique across the Internet. Because of this, Network Registrar supports IP addresses that are distinguished by a VPN identifier. Relay agents on routers must support this capability as well. This VPN identifier selects the VPN in which the client belongs. VPN support for DHCP is currently only supported by the Cisco Internet Operating System (IOS), the newest versions of which can include VPN IDs in the relayed DHCP messages.

Subnet allocation is a way of offering entire subnets (ranges of addresses) to relay agents so that remote access devices can provision IP addresses to DHCP client hosts. This can occur along with or instead of managing individual client addresses. Subnet allocation can vastly improve IP address provisioning, aggregation, characterization, and distribution by relying on the DHCP infrastructure to dynamically manage subnets. Subnet allocation through DHCP is currently only supported by Cisco IOS, the newest versions of which incorporate the on-demand address pools feature.

## Configuring Virtual Private Networks Through DHCP

To configure a VPN whereby a client can request IP addresses from a DHCP server using a relay agent, you must define the VPN and associate a scope with it. Specifically:

1. Ensure that the relay agents that handle DHCP VPN traffic are configured with a version of Cisco IOS that supports the *vpn-id* suboption of the *relay-agent-info* option (82) in DHCP.
2. Coordinate with the IOS relay agent administrator whether the VPN is identified by a VPN ID or a VPN Routing and Forwarding instance (VRF) name.
3. Create a scope for the VPN.



**Note**

---

In the CLI, a VPN is known as a namespace. Use the **namespace** command in the CLI.

---

## Typical Virtual Private Networks

Figure 2-13 on page 2-11 shows a VPN scenario with DHCP Client1 as part of VPN *blue* and DHCP Client2 in VPN *red*. Both have the same private network addresses 192.168.1.0/24. The DHCP relay agent has gateway addresses that are in the two VPNs as well as a global one (172.27.180.232). There are two failover DHCP servers, both of which know the relay agent through its external gateway address.

Here is the processing that takes place for the server to issue a VPN-supported address to a client:

1. DHCP client1 broadcasts a DHCPDISCOVER packet, including its MAC address, host name, and any requested DHCP options.
2. DHCP relay agent at address 192.168.1.1 picks up the broadcast packet. It adds a *relay-agent-info* option (82) to the packet and includes the *subnet-selection* suboption. This identifies 192.168.1.0 as the subnet. The packet also includes the *vpn-id* suboption that identifies the VPN as *blue*. Because the DHCP server cannot communicate directly with the requesting client, the *server-id-override* suboption contains the address of the relay agent as known by the client (192.168.1.1). The relay agent also includes in the packet its external gateway address (*giaddr*), 172.27.180.232.
3. The relay agent unicasts the DHCPDISCOVER packet to the configured DHCP server on its subnet.
4. DHCP server1 receives the packet and uses the *vpn-id* and *subnet-selection* suboptions to allocate an IP address from the proper VPN address space. It finds the available address 192.168.1.37 in the subnet and VPN, and places it in the *yiaddr* field of the packet (the address offered to the client).
5. The server unicasts a DHCPOFFER packet to the relay agent that is identified by the *giaddr* value.
6. The relay agent removes the *relay-agent-info* option and sends the packet to DHCP Client 1.
7. DHCP client1 broadcasts a DHCPREQUEST message requesting the same IP address that it was offered. The relay agent receives this broadcast message.
8. The relay agent forwards the DHCPREQUEST packet to DHCP Server1, which replies with a unicast DHCPACK packet to the client.
9. For a lease renewal, the client unicasts a DHCPRENEW packet to the IP address found in the *dhcp-server-identifier* option of the DHCPACK message. This is 192.168.1.1, the address of the relay agent. The relay agent unicasts the packet to the DHCP server. The server does its normal renewal processing, without necessarily knowing whether it was the server that gave out the original address in the first place. The server replies in a unicast DHCPACK packet. The relay agent then forwards the DHCPACK packet to the client's IP address identified by the *ciaddr* field value.

If the *server-id-override* suboption of the *relay-agent-info* option (82) exists, the DHCP server uses its value to compare to that of the *dhcp-server-identifier* option in the reply packet. Any packet that the DHCP client unicasts then goes directly to the relay agent and not to the server (which may, in fact, be inaccessible from the client). Both partners in a failover environment can renew a lease if the packet includes the *server-id-override* suboption.

## Setting Virtual Private Network Indexes

Here is what you can do as the administrator to set up the VPN and its index.

- 
- Step 1** Coordinate with the IOS relay agent administrator whether VPNs are configured by VPN ID or VRF name on the relay agent. This will determine how to identify the VPN in Network Registrar.

- Step 2** Create a VPN to include the DHCP client, with an ID number. A VPN index name can be any unique text string except the reserved words **all** or **global**. Its associated ID must also be unique. To do so:
- In the local cluster Web UI—On the Primary Navigation bar, click **DHCP**; on the Secondary Navigation bar, click **VPNs**. This opens the List/Add VPNs page. Give the VPN a numerical key identifier and a unique name in the cluster.
  - In the regional cluster Web UI—Add the local cluster containing the VPN (from the **Clusters** Primary Navigation bar and the **Cluster List** Secondary Navigation bar). Then, on the Primary Navigation bar, click **DHCP Configuration**; on the Secondary Navigation bar, click **VPNs**. This opens the List/Add VPNs page (see [Figure 19-1](#)). You can create the VPN on this page or pull the VPN from the local clusters:
    - If creating the VPN, give it a numerical key identifier and a unique name.
    - If pulling the VPN from the local clusters, click **Pull Replica VPNs** on the List/Add VPNs page, then pull specific or all the VPNs from the selected cluster.

**Figure 19-1** List/Add VPNs Page for Regional Cluster

Key*	Name*	VPN Id	VRF Name	Description
999	vpn-1	a1:3f6c		

Key	Name	VPN Id	VRF Name	Description	Push Data
VPN list is empty.					

You can also push VPNs to the clusters by clicking **Push VPN** or **Push All VPNs** on the List/Add VPNs page. You would then choose the synchronization mode and the clusters to which to push the VPNs on the Push VPN Data to Local Clusters page.

- In the CLI—Use the **namespace name create key** command, supplying its unique name string and its numerical key. For example:
 

```
nrcmd> namespace blue create 99
```

**Step 3** Specify the appropriate VPN identifier, either by VPN ID or VRF name. It is rarely both.

- If you use a VPN ID, set the *vpn-id* attribute value for the VPN. The value is usually in hexadecimal, in the form *oui:index*, per IETF RFC 2685. It consists of a three-octet VPN Organizationally Unique Identifier (OUI) that corresponds to the VPN owner or ISP, followed by a colon. It is then followed by a four-octet index number of the VPN itself.
  - In the local and regional cluster Web UI—Add the VPN ID value to the List/Add VPNs page.
  - In the CLI—Set the *vpn-id* attribute. For example:
 

```
nrcmd> namespace blue set vpn-id=a1:3f6c
```
- If you use a VPN Routing and Forwarding instance (VRF) name, set the *vrf-name* attribute value for the VPN. Cisco routers frequently use VRF names.
  - In the local and regional cluster Web UI—Add the VRF Name value to the List/Add VPNs page.
  - In the CLI—Set the *vrf-name* attribute. For example:
 

```
nrcmd> namespace blue set vrf-name=framus
```

**Step 4** In the Web UI or CLI—Add a description for the VPN, if you wish.

**Step 5** In the Web UI—Click **Add VPN**.

**Step 6** Create a scope for the VPN. Cisco recommends keeping the two names as similar as possible for identification purposes:

- In the local cluster Web UI—On the Primary Navigation bar, click **DHCP**, then click **Scopes** on the Secondary Navigation bar. This opens the List/Add DHCP Scopes page. Create a scope or edit an existing one. Under the Miscellaneous attributes, look for the *namespace-id* attribute. Choose the VPN from the drop-down list.
- In the CLI—You can identify to which VPN the scope belongs in one of three ways:
  - Its VPN name, through the *namespace* attribute (which applies the VPN ID to the scope).
  - The VPN ID itself, through the *namespace-id* attribute.
  - The default VPN name, by omitting the VPN or its ID on the command line.

You set the default VPN for the current session using the **session set current-namespace** command. You can then set the usual address range and necessary option properties for the scope. For example:

```
nrcmd> scope blue-1921681 create 192.168.1.0 255.255.255.0 namespace=blue
```

Or:

```
nrcmd> scope blue-1921681 create 192.168.1.0 255.255.255.0 namespace-id=99
```

Or:

```
nrcmd> session set current-namespace=blue
nrcmd> scope blue-1921681 create 192.168.1.0 255.255.255.0
```

Then:

```
nrcmd> scope blue-1921681 addRange 192.168.1.101 192.168.1.200
nrcmd> scope-policy blue-1921681 setOption routers 192.168.1.1
```

**Step 7** Create a nonVPN scope for nonVPN clients.

- In the local cluster Web UI—Create another scope without the Miscellaneous attribute *namespace-id* chosen, or change the choice in the drop-down list to [none].
- In the CLI—If the session's current VPN is set, unset it (using the **session unset current-namespace** command) to use the [none] VPN. Then, create a scope, which creates it without a VPN. Add ranges and options to it, as usual. For example:

```
nrcmd> session unset current-namespace
nrcmd> scope 1921682 create 192.168.2.0 255.255.255.0
```

**Step 8** Reload the DHCP server.

---

## VPN Usage with Other Objects or Processes

When you create a VPN, you must give it a name and a numerical ID, either a VPN ID or VRF name. Network Registrar stores the VPN internally by its ID. Once you set the ID, you cannot unset it, although you can change the VPN name associated with it.

In the CLI and GUI, you can include the VPN in many IP address specifications, as the VPN name prefix followed by a slash and the IP address. For example, lease names can have this syntax:

*vpn/ipaddress*

For example, red/192.168.40.0

A VPN can be any unique text string except the reserved words **global** and **all**. You can use **global** and **all** when you export address or lease data. The **global** VPN maps to the [none] VPN; the **all** VPN maps to both the specific VPN and the [none] VPN.

You can define or request the VPN or its ID for a few Network Registrar objects. If you omit the VPN or its ID in defining an object, the VPN defaults to the value set by the **session set current-namespace** command. If the current VPN is not defined, it defaults to the [none] VPN, which includes all addresses outside of any defined VPNs.

The objects that have associated VPN properties are:

- Address blocks—Define the VPN for an address block.
  - In the local and regional cluster Web UIs—On the Primary Navigation bar, click **Address Space**; on the Secondary Navigation bar, click **Address Blocks**. On the List/Add Address Blocks page, choose the VPN from the Select VPN drop-down list.
  - In the CLI—Use the **address-block** creation and attribute setting commands. For example:
 

```
nrcmd> address-block red create 192.168.50.0/24
nrcmd> address-block red set namespace=blue
nrcmd> address-block red set namespace-id=99
```
- Clients and client-classes—In some cases it is desirable to provision a VPN inside of Network Registrar instead of externally, where it might have to be configured for every Cisco Internet Operating System (IOS) device. To support this capability, you can specify a VPN for a client or client-class. Two attributes are provided:
  - *default-namespace*—VPN that the packet gets if it does not already have a *vpn-id* or *vrf-name* value in the incoming packet. You can use the attribute with clients and client-classes.
  - *override-namespace*—VPN the packet gets no matter what is provided for a *vpn-id* or *vrf-name* value in the incoming packet. You can use the attribute with clients and client-classes. Note that if you specify an override VPN on the client-class, and a default VPN for the client, the override VPN on the client-class takes precedence over the default VPN on the client.
  - In the local cluster Web UI—On the Primary Navigation bar, click **DHCP**; on the Secondary Navigation bar, click **Client-Classes**. Create or edit a client-class and enter the *default-namespace* and *override-namespace* attribute values.
  - In the regional cluster Web UI—On the Primary Navigation bar, click **DHCP Configuration**; on the Secondary Navigation bar, click **Client-Classes**. Create or pull, and then edit a client-class to enter the *default-namespace* and *override-namespace* attribute values.
  - In the CLI—Use the **client-class** creation and attribute setting commands. For example:
 

```
nrcmd> client 1,6,00:d0:ba:d3:bd:3b set default-namespace=blue
nrcmd> client-class CableModem set override-namespace=blue
```

In a cable modem deployment, for example, you can use the *override-namespace* attribute to provision the cable modems. The client-class would determine the scope for the cable modem, and the scope would determine the VPN for the uBR. User traffic through the cable modem would then have the *vpn-id* suboption set and use the specific VPN. The *override-namespace* value also overrides any *default-namespace* set for the client.

- Leases—List leases, show a lease, or get its attributes. If you had previously specified a VPN for the scope, the output shows leases in that VPN only.

In the CLI—To import leases, use the **import leases** *filename* command. Each lease entry in the file can include the VPN at the end of the line. If it is missing, Network Registrar assigns the [none] VPN.

```
nrcmd> import leases leaseimport.txt
```

To export the address or lease data to include the VPN, use the **export addresses** command with the *namespace* attribute, or the **export leases** command with the *-namespace* option. The VPN value can be the reserved word **global** or **all**:

- **Global**—Any addresses outside the defined VPNs (the [none] VPN).
- **All**—All VPNs, including the [none] VPN.

If you omit the VPN, the export uses the current one as set by the **session set current-namespace** command. If the current VPN is not set, the server uses the [none] one.

```
nrcmd> export addresses file=addressexport.txt namespace=red
nrcmd> export leases -server -namespace red leaseexport.txt
```

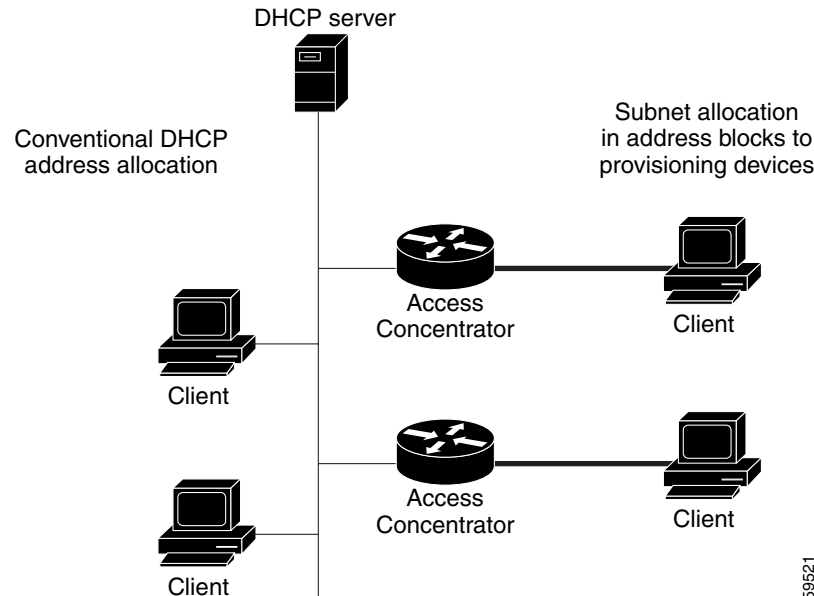
- Scopes—Scopes can include the VPN name or its ID, as described in the [“Setting Virtual Private Network Indexes”](#) section on page 19-2.
  - In the local cluster Web UI—On the Primary Navigation bar, click **DHCP**; on the Secondary Navigation bar, click **Scopes**. Create or edit a scope and set the Miscellaneous attribute *namespace-id*.
  - In the regional cluster Web UI—On the Primary Navigation bar, click **DHCP Configuration**; on the Secondary Navigation bar, click **Scope Templates**. Create or pull, and then edit a scope template to set the Miscellaneous attribute *namespace-id*.
  - In the CLI—Use the **scope** creation and attribute setting commands. For example:
 

```
nrcmd> scope examplescope1 set namespace=blue
nrcmd> scope examplescope1 set namespace-id=99
```
- Subnets—Listing subnets, showing a subnet, or getting the *namespace* or *namespace-id* attribute for a subnet shows the VPN. See the [“Configuring DHCP Subnet Allocation”](#) section on page 19-6.

## Configuring DHCP Subnet Allocation

The following section provides an example of setting up subnet allocation using the DHCP server. [Figure 19-2](#) shows a sample subnet allocation configuration with subnets assigned to provisioning devices, along with the conventional DHCP client/server configuration.

Figure 19-2 Subnet Allocation Using DHCP



**Step 1** Create an address block for a subnet, set the initial subnet mask and its increment, and set other subnet allocation request attributes. Also, associate a policy or define an embedded policy.

- If you use VPNs, you can specify a *namespace* or *namespace-id* attribute (see the “[Configuring Virtual Private Networks Through DHCP](#)” section on page 19-1). Note that unsetting the VPN ID reverts the value to the current session VPN.
- If you do not use VPNs, you can specify a *selection-tags* attribute identifying one or more subnets from which to allocate the addresses. If the relay agent sends this string, which it associates with a subnet, any address block with that string as one of its *selection-tags* attribute values uses that subnet.

For incoming subnet allocation requests that do not have a scope-selection tag, you can set the default scope-selection tag value or values on the server or VPN level.

The scope-selection tags capability is enabled by default for the server and VPNs, but you can disable it for each one. Note also that the default behavior on the server and for VPNs is that the DHCP server tries to allocate subnets to clients using address blocks that the clients already used. Disabling the *blocks-use-client-affinity* attribute causes the server to supply subnets from any suitable address block, based on other selection data in the clients’ messages.

- If you want to support configurations of multiple address blocks on a single LAN segment (analogous to using primary and secondary scopes), add a *segment-name* attribute string value to the address block. When the relay agent sends a single subnet selection address, it selects address blocks tagged with that *segment-name* string value. However, you must also explicitly enable the LAN segment capability on the server or VPN level; this is disabled by default.
- Instead of associating a policy, you can set properties for the address block’s embedded policy. As in embedded policies for clients, client-classes, and scopes, you can enable, disable, set, unset, get, and show attributes for an address block policy. You can also set, unset, get, and list any DHCP options for it, as well as set, unset, get, and list vendor options. Note that deleting an address block embedded policy unsets all the embedded policy properties.

**Step 2** Note that the server allocates subnets based on the relay agent request. If not requested, the default subnet size is a 28-bit address mask. You can change this default, if necessary, by setting the *default-subnet-size* attribute for the address block. For example:

```
nrcmd> address-block red set default-subnet-size=25
```

**Step 3** Reload the DHCP server.

**Step 4** You can control any of the subnets the DHCP server creates from the address blocks. Identify the subnet in the form *vpn-name/netipaddress/mask*, with the *vpn-name* optional. Subnet control includes activating and de-activating the subnet as you would a lease. Likewise, you can force a subnet to be available, with the condition that before you do so, check that the clients assigned the subnet are no longer using it. First, show any subnets created.

**Step 5** You can list the subnets that Network Registrar created for an address block. In the CLI:

```
nrcmd> address-block red listsubnets
```

## Tuning Parameters

Consider these tuning parameters for VPNs and on-demand address pools.

- Keep orphaned leases that have nonexistent VPNs—Network Registrar usually maintains leases that do not have an associated VPN in its state database. You can change this by enabling the DHCP attribute *delete-orphaned-leases*. The server maintains a lease state database that associates clients with leases. If a scope modification renders the existing leases invalid, the lease database then has orphaned lease entries. These are typically not removed even after the lease expires, because the server tries to use this data in the future to re-associate a client with a lease. One downside to this is that the lease database may consume excessive disk space. By enabling the *delete-orphaned-leases* attribute, such lease database entries are removed during the next server reload. However, be cautious when enabling this attribute, because rendering leases invalid can result in clients using leases that the server believes to be free. This can compromise network stability.
- Keep orphaned subnets that have nonexistent VPNs or address blocks—This is the default behavior, although you can change it by enabling the DHCP attribute *dhcp enable delete-orphaned-subnets*. As the DHCP server starts up, it reads its database of subnets and tries to locate the parent VPN and address block of each subnet. With the attribute enabled, if a subnet refers to a VPN that is no longer configured in the server, or if the server cannot locate a parent address block that contains the subnet, the server permanently deletes the subnet from the state database.
- Keep the VPN communication open—This is the default behavior, although you can change it by disabling the DHCP attribute *vpn-communication*. The server can communicate with clients that reside on a different VPN from that of the server by using an enhanced DHCP relay agent capability. This is signaled by the appearance of the *vpn-id* suboption of the *relay-agent-info* option (82). You can disable the *vpn-communication* attribute if the server is not expected to communicate with clients on a different VPN than the server. The motivation is typically to enhance network security by preventing unauthorized DHCP client access.