



Managing Leases

Leases are at the center of the Dynamic Host Configuration Protocol (DHCP). They are the addresses allocated to individual clients for a certain time period. The DHCP server automatically allocates these leases with properly configured scopes that include valid address ranges. No two clients can have the same leased address.

This chapter describes how to manage leases in a network.

Configuring Leases in Scopes

After setting the address ranges for a scope, you can monitor and adjust the leases that result from DHCP assignments. While there is no limit to the number of leases that you can configure per scope, if you have one with several thousand leases, it can take Network Registrar awhile to sort them.

Viewing Leases

You can view the current state of leases for the scope address ranges.

In the local cluster Web UI, create a scope, as described in the [“Defining and Configuring Scopes” section on page 19-2](#). You can view the leases in two ways:

- Click the View icon (🔍) in the Leases column for the scope on the List/Add DHCP Scopes page.
- Click the name of the scope on the List/Add DHCP Scopes page to open the Edit DHCP Scope page. Click **List Leases** in the Leases area of the page.

Both of these actions open the List DHCP Leases for Scope page, where you can click each lease to manage it.

In the CLI, use **lease name show** to show the properties of a particular lease. Use **scope name listLeases** to show all the leases for a scope. The output is nearly identical for both commands. Note that you cannot list leases in a particular virtual private network (VPN); all the leases in all the VPNs are listed.

You can show the most recent MAC address associated with a lease or what lease is associated with a MAC address. The **lease addr macaddr** command shows the MAC address of the lease whether or not that lease is reserved or leased out. The **lease addr list –macaddr** command lists the lease data only if the IP address for that MAC address was actually leased out (and not reserved). You can also list leases by LAN segment and subnet using **lease addr list –subnet network netaddr netmask**.

Lease States

A lease can be in one of these states:

- Available—Leasable.
- Unavailable—Not leasable. See the [“Handling Leases Marked as Unavailable”](#) section on page 21-12 for ways the DHCP server might set a lease to unavailable.
- Leased—Held by a client.
- Offered—Offered to the client.
- Expired—Available when the lease grace period expires.
- Deactivated—Not renewable or leasable after the lease expires. See the [“Deactivating Leases”](#) section on page 21-5.
- Pending available—Failover-related. See [Chapter 26, “Configuring DHCP Failover.”](#)



Note

When the state of an existing lease changes (for example, when it is configured as a reserved address or is deactivated), the change is not reported as an IP lease history change to the regional servers. A change in lease history is recorded only when the lease expires or the address is leased to another client. (See the [“Running IP Lease Histories”](#) section on page 21-15.) Lease state changes are logged only if you enable the *ip-history-detail* attribute for the server.

Guidelines for Lease Times

To define appropriate values for lease times, consider these events on your network:

- Frequency of changes to DHCP options and default values
- Number of available addresses compared to clients requesting them
- Number of network interface failures
- Frequency at which computers are added to and removed from the network
- Frequency of subnet changes by users

All these events can cause clients to release addresses or the leases to expire at the DHCP server. Consequently, the addresses may return to the free-address pool for reuse. If many changes occur on your network, Cisco recommends a lease time between four and ten days. Assigning such a lease time more quickly reassigns addresses as clients leave the subnet.

Another important factor is the ratio of available addresses to connected computers. For example, the demand for reusing addresses is low in a class C network having 254 available addresses, of which only 40 are used. A long lease time, such as two months, might be appropriate in such a situation. The demand would be much higher if there were 240 to 260 clients trying to connect at one time. In this situation, you should try to configure more address space. Until you do, keep the DHCP lease time to under a hour.



Tip

Short lease periods increase the demand that the DHCP server be continuously available, as clients will be renewing their leases more frequently. The DHCP failover functionality can help guarantee such levels of availability.

Create policies that have permanent leases carefully. There is a certain amount of turnover among clients even in a stable environment. Portable hosts might be added and removed, desktop hosts moved, and network adapter cards replaced. If you remove a client with a permanent lease, it requires manual intervention in the server configuration to reclaim the address. It would be better to create a long lease, such as six months, which ensures that addresses are ultimately recovered without administrator intervention.

Recommendations for lease durations include:

- Set cable modem lease times to seven days (604800 seconds). The leases should come from private address space, and the cable modems should seldom move around.
- Leases for customer premises equipment (CPE) or laptops should come from public address space and should match the habits of your user population, with as long a lease as possible to reduce load on the server.
- Shorter lease times require more DHCP request and response buffers. Set the request and response buffers for optimal throughput (see the [“Setting DHCP Request and Response Packet Buffers”](#) section on page 26-16).

Importing and Exporting Lease Data

You can use the CLI to import lease data to, and export from, text files.

Import Prerequisites

Before you can import leases, you must perform several configuration steps:

1. Configure scopes in the DHCP server for the leases that you plan to import.
2. If you want the host names for the leases dynamically entered into DNS as part of the import, configure zones in the DNS server to allow dynamic updates from the DHCP server.
3. Set the DHCP server to import mode so that it does not respond to other lease requests during the lease importing.
4. For all the time fields, use either the number of seconds since midnight GMT January 1, 1970, or day, month, date, time, year format (Mon Apr 15 16:35:48 2002).
5. After you import the leases, take the DHCP server out of import mode so that it can respond to other lease requests.

**Note**

Importing permanent leases will fail if you disable the permanent leases option, so enable this option using **policy name enable permanent-leases**, as necessary.

Import and Export Commands

The **import leases** and **export leases** commands use the following file format. Each record, or line, in the file represents one DHCP client:

```
field-1|field-2|field-3|...|field-13
```

There are no spaces between the vertical line (|) delimiter and the field values. You must include at least the first four required fields. If including more, so you must delimit all the remaining null fields with the vertical line (|) so that there are 13 fields.

The fields are, in order:

1. MAC address in *aa:bb:cc:dd:ee:ff* format (required)
2. MAC address type (required)
3. MAC address length (required)
4. IP address in dotted decimal format, *a.b.c.d* (required)
5. Start of lease time (Greenwich Mean Time, GMT) (optional)
6. Lease expiration time (GMT) (optional)
7. Allowable extension time (GMT) (optional)
8. Last transaction time (GMT) (optional)
9. IP address of the DHCP server (optional)
10. Host name (without domain) (optional)
11. Domain name (optional)
12. Client ID (optional)
13. VPN name (optional; if omitted, the global VPN is used)

For all the time fields, use either the number of seconds since 1970, or the *day-month-day-time-year* format (such as Mon Apr 10 16:35:48 2000).

When importing leases, the DHCP server might not accept a lease, or a communication failure might drop the lease packet. In the latter case, the server retries the import several times, and after about a minute, reports a failure. If the import fails, check the DHCP server log file to find the lease that caused the error. Then go back to the import file, delete all lease entries up to and including the offending one, and repeat the lease import.

When you use **export leases**, you can choose between writing the state of all current and expired leases, or just the current leases, to the output file. [Example 21-1](#) shows part of a lease data export from a Network Registrar DHCP server. The blank lines between records appear in the example for clarity; they are not in the actual output.

Example 21-1 Lease Data Export

```
00:60:97:40:c1:96|1|6|204.253.96.103|Wed Aug 30 08:36:57 2000|Fri Sep 01 13:34:05 2000|
Wed Aug 30 08:36:57 2000|Fri Sep 01 09:34:05 2000|204.253.96.57|nomad|cisco.com|
00:d0:ba:d3:bd:3b|blue-vpn

00:d0:ba:d3:bd:3b|1|6|204.253.96.77|Thu Aug 17 13:10:11 2000|Fri Sep 01 14:24:46 2000|
Thu Aug 17 13:10:11 2000|Fri Sep 01 10:09:46 2000|
204.253.96.57|NPI9F6AF8|cisco.com|blue-vpn

00:d0:ba:d3:bd:3b|1|6|204.253.96.78|Fri Jun 23 15:02:18 2000|Fri Sep 01 14:11:40 2000|
Fri Jun 23 15:02:18 2000|Fri Sep 01 09:56:40 2000|
204.253.96.57|JTB-LOCAL|cisco.com|blue-vpn
```

Lease Times in Import Files

The client is given the lesser of these lease times:

- In the import file.
- What the client would receive if it were to acquire a lease using the existing configuration of the DHCP server.

For example, it is 2:00 p.m. and your scope is configured for a one-hour lease. According to the file that you import, the lease time does not expire until 5:00 p.m. After you import the file, the lease expires at 3:00 p.m. and not at 5:00 p.m.

**Caution**

If your import file specifies a DNS zone name, the server does not use the zone name when it updates DNS. If the file specifies a host name, then the server uses the host name when updating DNS, unless the host name was overridden by a host name specification in a client or client-class entry.

The only way to indicate to the DHCP server that the client's host name should be in a zone other than the zone associated with the DNS update configuration object used for the DNS update is to specify that zone in a client or client-class entry.

Pinging Hosts Before Offering Address

You can have the DHCP server use the Internet Control Message Protocol (ICMP) echo message capability (also known as **ping**) to see if anyone responds to an address before assigning it. This allows the DHCP server to check that an address is not in use before assigning it. Using **ping** can help prevent two clients from using the same address. If a client responds to the ping, the DHCP server marks that address as *unavailable* and offers a different address. This works only for powered-up clients; it is perfectly valid for clients to have a lease and be powered down.

**Tip**

The ping timeout period is important. Because pinging helps to ensure that no client is using a particular address, each ping must wait the entire timeout period. This ping timeout period comes before an offer, so the time specified has a considerable effect on server performance. If you set this time too long, it slows down the lease offering process. If you set this time too short, it reduces the effectiveness of the ping packet's ability to detect another client using the address.

To implement pinging hosts before offering addresses, modify the scope by:

- Enabling the *ping-clients* attribute. It is disabled by default. In the CLI, use **scope name enable ping-clients**.
- Setting the *ping-timeout* attribute. It is 300 milliseconds by default. In the CLI, use **scope name set ping-timeout**.


The server will make unavailable any address for which it receives a successful ECHO reply. You can control this by enabling the DHCP server attribute *ignore-icmp-errors* (**dhcp enable ignore-icmp-errors**), which is the default. If disabled, the DHCP server also treats ICMP DEST_UNREACHABLE and TTL_EXPIRED error messages that it receives after sending ICMP ECHO requests as grounds for making an address unavailable.

Deactivating Leases

The reason you would deactivate a lease is to move a client off of it. If the lease is available, deactivating it prevents Network Registrar from giving it to a client. If the lease is active (held by a client), deactivating it prevents the client from renewing it and the lease being given to another client. You can only deactivate a lease if the server is running. Network Registrar deactivates the lease immediately.

**Tip**

To force a Windows client to release its lease, run **ipconfig /release** on the client machine.

-
- Step 1** In the local Web UI, click **DHCP**, then **Scopes** to open the List/Add DHCP Scopes page.
 - Step 2** Click the View icon () under the Lease column for the scope that has leases.
 - Step 3** Click the name of the lease on the List DHCP Leases for Scope page.
 - Step 4** On the Manage DHCP Lease page, click **Deactivate**. The lease now shows as deactivated. To reactivate the lease, click **Activate** on the Manage DHCP Lease page.
In the CLI, use **lease ipaddr deactivate**. To reactivate a lease, use **lease ipaddr activate**.
 - Step 5** If you set staged edit mode for the scope, reload the server.
-

Excluding Addresses from Ranges


Addresses ranges, by definition, must be contiguous, without any holes in the range. Therefore, to exclude an address from an existing range, you must divide the range into two smaller ranges. The new ranges then consist of the addresses between the original starting and ending range addresses and the address which you want to exclude.

However, if the address being excluded currently has an active lease, you should first follow the steps in the “[Deactivating Leases](#)” section on page 21-5. You will get a warning message otherwise.



Caution

Deleting an active lease can cause a duplicate address on the network if the deleted address is subsequently reconfigured and then reassigned. Information about that lease will no longer exist after you reload the server.

- Step 1** In the Web UI, on the Edit DHCP Scope page, in the Ranges area, click the Delete icon () next to the address range you want to remove.
In the CLI, discover the lease range (**scope name listRanges**), deactivate the lease (**lease ipaddr deactivate**), then remove the range of just that address (**scope name removeRange**). The resulting ranges are then split appropriately.
- Step 2** In the Web UI, Add a range that ends just before the excluded address.
- Step 3** Click **Add Range**.
- Step 4** Add another range that begins just after the excluded address.
- Step 5** Click **Add Range**.
- Step 6** Click **Modify Scope**.

The following example in the CLI removes the 192.168.1.55 address from the range. Note that if the lease is in a scope with a defined VPN, you must explicitly define that VPN for the session, or you can include the VPN prefix in the **lease** command:

```
nrcmd> session set current-vpn=red
nrcmd> scope examplescope1 listRanges
nrcmd> lease red/192.168.1.55 deactivate
nrcmd> scope examplescope1 removeRange 192.168.1.55 192.168.1.55
nrcmd> scope examplescope1 listRanges
```

- Step 7** If you set staged edit mode for the scope, reload the server.
-

Reserving Leases

To ensure that a client always gets the same lease, you can create a lease reservation. You must reserve leases for DHCP clients whose addresses must remain constant.



Caution


If multiple DHCP servers distribute addresses on the same subnet, the client reservations must be identical. If not, a client for whom a lease reservation exists can receive offers of different addresses from different servers.

A lease reservation consists of an IP and MAC address pairing. You can choose any valid address on the network. It does not necessarily have to be in one of the scope's ranges. In fact, you can use the addresses in the scope's range for dynamic leases and the others not in the range for reserved leases.



Note

Even though a reserved address may not be in the scope's range, the policy associated with the scope still applies to the address.

- Step 1** In the Web UI, on the Edit DHCP Scope page, in the Reservations area, add the IP address to reserve, and either a client lookup key (string, binary, or MAC address), or its MAC address proper (you cannot use both the lookup key and the MAC address). Ensure that the reservation is part of an existing range in the Ranges area of the page. The lookup key or MAC address is required.
- In the CLI, list the lease reservations for the scope (**scope name listReservations**). The output shows a value for MAC address or lookup-key for each reservation.
- Step 2** Click **Add Reservation**. Note that if there are more than ten reservations, they appear on a separate List/Add DHCP Reservations for Scope page, accessible by clicking **List Reservations**.
- In the CLI, use **scope name addReservation ipaddr {macaddr | lookupkey}**. If you use the *lookupkey*, you can add the **-mac**, **-blob**, or **-string** option to the command.
- Step 3** Click **Modify Scope**. (Be sure to do this or your changes will not go into effect.)
- In the CLI, save (**save**), then send the reservation (**lease ipaddr send-reservation**), which does not require a reload.
- Step 4** If you want to make an existing lease reserved:
- In the Web UI, on the Edit DHCP Scope page, click the View icon () under the Lease column for the scope that has leases.
 - Click the name of the lease on the List DHCP Leases for Scope page.
 - On the Manage DHCP Lease page, if the address is not leased (in available state), enter the lookup key or MAC address for the reservation.
 - Click **Make Reservation**. On the List DHCP Leases for Scope page, the lease will now show as being reserved.
 - Click **Modify Scope**. (Be sure to do this or your changes will not go into effect.)
 - To remove the reservation, repeat the steps, then click **Remove Reservation** on the Manage DHCP Lease page, and then modify the scope. The lease no longer shows as being reserved.
- Step 5** If you set staged edit mode for the scope, reload the server.

**Note**

If you want to override use of an incoming DHCP client-id that is based on the device's MAC address, either set the *override-client-id* attribute for the client or client-class (an expression value that overrides the incoming client-id in the request packet), or enable the *use-client-id-for-reservations* attribute for the client's DHCP policy (disabled by default). (See the [“Extending Reservations to Non-MAC Addresses” section on page 21-8.](#))

Extending Reservations to Non-MAC Addresses

Customers might need to create lease reservations based on something other than the MAC address from the incoming client packet. There is often a need to allow any DHCP client device attached to a port on a switch always to get the same IP address, regardless of the MAC address. This approach is sometimes used when you need to replace factory floor devices with identical devices (with different MAC addresses), but still maintain the same DHCP IP address.

Overriding Client IDs

In Network Registrar 6.2, you can set an expression in a client-class *override-client-id* attribute (or *v6-override-client-id* attribute for IPv6) that extracts the switch's MAC address-and-port from the relay-agent-info option (82), and creates a client identity from it. Then, regardless of the client-id in the incoming packet, the identity used to allocate an IP address is always the same for any device coming in through the same switch port. The expression you use for the attribute varies based on the option 82 format. The expression is calculated when the packet is assigned to the client-class. The *[v6-]override-client-id* value becomes the identity of the client from that point forward.

In the Web UI, the *[v6-]override-client-id* attribute is found on the Add DHCP Client-Class page, or Edit DHCP Client-Class page. In the CLI, the syntax for setting this attribute is:

```
nrcmd> client-class name set [v6-]override-client-id="expression"
```

However, when the *use-client-id-for-reservations* attribute is enabled in a policy, the client-id of that request is turned into a string of the form *nn:nn:nn ... nn:nn*, and that string is used to look up the reservation. Hence, if you want a reservation for something other than a MAC address:

1. Configure the policy to enable the *use-client-id-for-reservations* attribute. The CLI syntax is:

```
nrcmd> policy name enable use-client-id-for-reservations
```

2. Configure a *[v6-]override-client-id* for the client-class to give clients the identity that you want them to have (using the previously mentioned syntax in the CLI).
3. Configure a *client-class-lookup-id* expression for the server, to put every packet into a particular client-class where the *[v6-]override-client-id* expression is configured. The CLI syntax is:

```
nrcmd> dhcp set client-class-lookup-id="expression"
```

The *add-to-environment-dictionary* attribute for a client or client-class also serves to send attribute values to the DHCP extension environment dictionary (see [Chapter 28, “Using Extension Points”](#)), specified as attribute-value pairs. The attributes for the client go to the environment dictionary before those for the client-class. Because of this, if the attributes for both are the same, then the client-class values replace those of the client. This is based on the fact of not knowing whether the existing environment dictionary attributes come from the client (where they would be left alone) or from an earlier operation (where they would be replaced by the client-class values).

Reservation Override Example

-
- Step 1** Create a scope (or IPv6 prefix) for the reservation:
- Enter a subnet address.
 - If you want dynamic reservations, add an address range.
- Step 2** Add the reservation for the scope:
- Include a value for the lookup-key.
 - Specify the lookup-key type as binary.
- Step 3** Create a policy for the purpose, enabling the *use-client-id-reservations* attribute.
- Step 4** Create a client-class for the purpose:
- Specify the policy created in the previous step.
 - Include an expression for the *[v6-]override-client-id* attribute that returns a blob value with the client ID you want based on the contents of the packet.
- Step 5** Get a lease for a client with the MAC address. This client will then get the override ID.
-

Reserving Currently Leased Addresses

It is possible to delete a reservation for one client and reuse the reservation for a second client, even though the first client still has the lease. The following example describes how Network Registrar behaves in this situation. Assume that you have this reservation and lease:

```
nrcmd> scope examplescope1 addReservation 192.168.1.110 1,6,00:d0:ba:d3:bd:3c
nrcmd> save
nrcmd> lease 192.168.1.110 send-reservation
nrcmd> lease 192.168.1.110 activate
nrcmd> save
```

Client 1,6,00:d0:ba:d3:bd:3b does a DHCPDISCOVER and gets an offer for 192.168.96.180. The client then does a DHCPREQUEST and gets an ACK message for the same IP address.

As time passes, client 1,6,00:d0:ba:d3:bd:3b does several DHCPREQUESTs that are renewals, which the server acknowledges. Then, at some time before the client's lease expiration time, you terminate the reservation:

```
nrcmd> lease 192.168.1.110 deactivate
nrcmd> scope examplescope1 removeReservation 192.168.1.110
nrcmd> save
nrcmd> lease 192.168.1.110 delete-reservation
```

You then add a reservation for a different client for that IP address, even though the address is still leased to the first client:

```
nrcmd> scope examplescope1 addReservation 192.168.1.110 1,6,02:01:02:01:02:01
nrcmd> save
nrcmd> lease 192.168.1.110 send-reservation
nrcmd> lease 192.168.1.110 activate
nrcmd> save
```

This action results in an address that is leased to one client, but reserved for another. If the new client (1,6,02:01:02:01:02:01) does a DHCPDISCOVER before the original client (1,6,00:d0:ba:d3:bd:3b) does, the new client does not get 192.168.96.180, but gets a random IP address from the dynamic pool.

When the original client (1,6,00:d0:ba:d3:bd:3b) sends its next DHCPREQUEST/RENEW for the lease on 192.168.96.180, it gets a NAK message. Generally, upon receipt of the not-acknowledged message, the client immediately sends a DHCPDISCOVER. On receiving that DHCPDISCOVER, the server cancels the remaining lease time for 192.168.96.180.

The server then gives client 1,6,00:d0:ba:d3:bd:3b whatever lease is appropriate for it—some reservation other than 192.168.96.180, some dynamic lease (if one is available), or nothing (if no dynamic leases are available). When the new client (1,6,02:01:02:01:02:01) tries to renew the random IP address it received, the server sends it a NAK, because it wants to give it the reserved address. When the new client then does a DHCPDISCOVER, it gets the 192.168.96.180 reserved address.

You could also force availability of a lease, using **lease ipaddr force-available**. However, that does not stop the original client (1,6,00:d0:ba:d3:bd:3b) from using 192.168.96.180. Also, it does not prevent the new client (1,6,02:01:02:01:02:01) from getting 192.168.96.180. In other words, this means that making a reservation for a client is independent of the lease state (and actual lease client) of the IP address for which the reservation is made. Thus, making a reservation for one client does not cause another client to lose that lease right away, although that client receives a NAK response the next time it contacts the DHCP server (which could be seconds or days). Additionally, the client that reserved the IP address does not get it if some other client already has it. Instead, it gets some other IP address until the:

- IP address it is supposed to receive is free.
- Client sends a DHCPREQUEST as a renewal and receives a NAK response.
- Client sends a DHCPDISCOVER.

Unreserving Leases

You can remove lease reservations at any time. However, if the lease is still active, the client continues to use the lease until it expires. If you try to reserve the lease for a different client, you will get a warning.

In the local cluster Web UI, on the Edit DHCP Scope page, in the Reservations area, click the Delete icon (🗑️) next to the reservation you want to remove. This removes the reservation immediately, with no confirmation. Then, click **Modify Scope**.

In the CLI, use **scope name removeReservation {ipaddr | macaddr | lookupkey}**. (Note that omitting the addresses or lookup-key removes all lease reservations.) Removing the reservation requires a server reload in staged scope edit mode. You can also completely delete a reservation by using **lease ipaddr delete-reservation**, which does not require a reload. However, when using this command:

- Ensure that the reservation was already removed from the **nrcmd** internal database.
- If you use failover on the scope in which the reservation resides:
 1. Use **lease ipaddr delete-reservation** on the backup server.
 2. Use **lease ipaddr delete-reservation** on the main server.
 3. Use **scope name removeReservation** on both systems.



Tip


Save the results of this operation to ensure that it is preserved across server reloads, because issuing **lease ipaddr delete-reservation** alone affects only the server's internal memory.

Forcing Lease Availability

You can force a current lease to become available. You should request that the user release the lease, or do so yourself, before forcing its availability. Forcing lease availability does not require a server reload.

**Caution**

After a lease is forced available, the client continues to use it until the client contacts the DHCP server.

-
- Step 1** In the local Web UI, click **DHCP**, then **Scopes** to open the List/Add DHCP Scopes page.
- Step 2** Click the View icon () under the Lease column for the scope that has leases.
- Step 3** Click the name of the lease on the List DHCP Leases for Scope page.
- Step 4** On the Manage DHCP Lease page, click **Force Available**. On the List DHCP Leases for Scope page, the lease will now show an empty value in the Flags column.

In the CLI, use **lease ipaddr force-available**. Use **scope name clearUnavailable** to force all leases in the scope to become available.

Inhibiting Lease Renewals

Normally, the Network Registrar DHCP server retains the association between a client and its leased IP address. The DHCP protocol explicitly recommends this and it is a feature that is usually desirable. However, for some customers, such as ISPs, clients with long-lived lease associations may be undesirable, as these client should change their IP addresses periodically. Network Registrar includes a feature that allows customers to force lease associations to change when DHCP clients attempt to renew their leases or reboot.

A server can never force a client to change its lease, but can compel the client to do so based on a DHCPRENEW or DHCPDISCOVER request. Network Registrar offers configuration options to allow customers to choose which interactions to use to force a client to change its address:

- Inhibiting all lease renewals—While a client is using a leased address, it periodically tries to extend its lease. At each renewal attempt, the server can reject the lease, forcing the client to stop using the address. The client might have active connections that are terminated when the lease terminates, so that renewal inhibition at this point in the DHCP interaction is likely to be user-visible.
- Inhibiting renewals at reboot—When a DHCP client reboots, it might have recorded a valid lease binding that did not expire, or it might not have a valid lease. If it does not have a lease, you can prevent the server from granting the last held lease. If the client has a valid lease, the server rejects it, forcing the client to obtain a new one. In either case, no active connections can use the leased address, so that the inhibition does not have a visible impact.
- Effect on reservations—Reservations take precedence over renewal inhibition. If a client has a lease reservation, it can continue to use the reserved address, whether or not renewal inhibition is configured.
- Effect on client-classes—Client-class testing takes place after renewal inhibition testing. A client may be forced to change addresses by renewal inhibition, then client-class processing might influence which address the server offers to the client.

In the local cluster Web UI, create a policy, on the Edit DHCP Policy page, enable the *inhibit-all-renews* or *inhibit-renews-at-reboot* attribute. Both are disabled by default. Then, click **Modify Policy**.

In the CLI, you can enable or disable lease renewal inhibition for a policy, which you can set system-wide, for a scope, or on a client-by-client basis. The *inhibit-all-renews* attribute causes the server to reject all renewal requests, forcing the client to obtain a new address any time it contacts the DHCP server. The *inhibit-renews-at-reboot* attribute permits clients to renew their leases, but the server forces them to obtain new addresses each time they reboot.

The DHCP server needs to distinguish between a client message that it should reject (such as a renewal request) and one that represents a retransmission. When the server processes a message, it records the time the packet arrived. It also records the time at which it made a lease binding to a client, and the last time it processed a message from the client about that binding. It then compares the packet arrival time with the lease binding time (the start-time-of-state) and processes packets from the client within a certain time interval from the start time of the binding. By default, this time interval is one minute.

Handling Leases Marked as Unavailable

One of the aspects of effective lease maintenance is determining the number of unavailable leases in a scope. This number is sometimes higher than expected. Each unavailable lease is probably an indication of a serious problem. Possible causes and are:

- The DHCP server is configured for a ping before an offer, and the ICMP echo message is returned successfully—This indicates that there is a currently active client using that address, causing the DHCP server to mark it as *unavailable*. To prevent the server from doing so, disable pinging an address before offering it to a client. See the [“Pinging Hosts Before Offering Address” section on page 21-5](#).
- The server receives a DHCPDECLINE message from a client to which it leased what it considered to be a good address—The client does an address resolution (ARP) request for the address on its local LAN segment, and another client responds to it. The client then returns the address to the server with a DHCPDECLINE packet and sends another DHCPDISCOVER packet to get a new address. The server marks as *unavailable* the address that the client returns. To prevent the server from reacting to DHCPDECLINE messages, you can set a scope attribute, *ignore-declines*.
- The server receives “other server” requests from the client—Because all DHCPREQUEST messages that follow DHCPOFFER messages are broadcast, the server can see messages directed to other DHCP servers. A server knows that a message is directed to it by the value of the *server-id* option in the packet. If the Network Registrar server recognizes a message directed at another server, in that its own address does not appear in the *server-id* option, but the IP address leased in the message is one that the server controls, it believes that two servers must be trying to manage the address simultaneously. It then marks the local address as *unavailable*. This does not apply in a DHCP failover configuration. Either the two servers are configured with some or all of the same IP addresses, or (in rare cases) the DHCP client placed a wrong *server-id* option value in the packet.

If you have reason to believe that the client is sending bad *server-id* options (rather than packets actually directed to other servers), Network Registrar has a server attribute you can enable that turns this behavior off, the *ignore-requests-for-other-servers* attribute in the Web UI or CLI.

- Inconsistent lease data—This is extremely rare and occurs only during server startup. Inconsistent lease data happens while configuring a lease, when the server reads the lease data from disk during a refresh of the internal cache. The lease state shows as *leased*, but there is incomplete data to construct a client for that lease. For example, it might not yet have a *client-id* option value. The server considers the data to be inconsistent and marks the address *unavailable*. The **lease address force-available** command should clear up this problem.

Setting Timeouts for Unavailable Leases, Including Upgrades

During the times when leases become unavailable, as described in the [“Handling Leases Marked as Unavailable” section on page 21-12](#), all unavailable leases remain in that state for a configured time only, after which time they again become available. A policy attribute, *unavailable-timeout*, controls this time. The *system_default_policy* policy sets this value to one day by default.

To handle upgrades from previous releases of Network Registrar that do not have this timeout feature, a special upgrade timeout attribute, *upgrade-unavailable-timeout* (which also defaults to one day) is included at the server level.

The *upgrade-unavailable-timeout* value is the timeout given to leases set to unavailable before the Network Registrar 6.0 upgrade. This setting affects the running server only and does not rewrite the database. If the server stays up for one day without reloading, all of the unavailable leases that were present at the last reload will time out. If the server is reloaded in less than a day, the entire process restarts with the next reload. Note that this process only occurs for leases that were set unavailable before the upgrade to Network Registrar 6.0. Leases that become unavailable after the upgrade receive the *unavailable-timeout* value from the policy, as previously described.

If a Network Registrar 6.0 failover server receives an update from a Network Registrar DHCP server running prior to Network Registrar 6.0, the unavailable leases do not have a timeout value. In this case, the Network Registrar 6.0 server uses the *unavailable-timeout* value configured in the scope policy or *system_default_policy* policy as the timeout for the unavailable lease. When the lease times out, the policy causes the lease to transition to available in both failover partners.

Running Address and Lease Reports

You can run these reports on IP addresses and leases that are addressed in the following sections:

- Address Usage—Displays the IP addresses used by a DHCP server.
- Lease History—Provides a history of the leases in a network.
- Lease Utilization—Displays statistics about leases in scopes.
- Current Utilization—Displays the current utilization of the addresses in a subnet.
- Lease Notification—Receive notification if available addresses in a scope fall below a certain level.

Running Address Usage Reports

In the local cluster Web UI, on the Edit DHCP Scope page, in the Leases area, click **List Leases** to open the List DHCP Leases for Scope page (see [Figure 21-1](#)).

To manage a specific lease, click its name on the page. This opens the Manage DHCP Lease page (see [Figure 21-2](#) for a partial view).

On this page, you can force a lease to be available, and you can deactivate a lease:

- To force a lease to become available, click **Force Available**.
- To deactivate an active lease, click **Deactivate**.
- To cancel the page, click **Cancel**.

Each action returns to the List DHCP Leases page.

Figure 21-1 List DHCP Leases for Scope Page (Local)

Home Administration Servers Clusters Routers DHCP DNS Hosts Address Space					
Scopes Scope Templates Prefixes Links Options Policies Clients Client-Classes VPNs Networks Failover DNS LDAP Extensions Traps DHCP Server					
List DHCP Leases for Scope <i>example-scope</i>					
Address	State	MAC Address	Hostname	Flags	Expiration
192.168.50.101	available			failover-updated	
192.168.50.102	available			failover-updated	
192.168.50.103	available			failover-updated	
192.168.50.104	available			failover-updated	
192.168.50.105	available			failover-updated	
192.168.50.106	available			failover-updated	

149416

Figure 21-2 Manage DHCP Lease Page (Local)

Home Administration Servers Clusters Routers DHCP DNS Hosts Address Space		
Scopes Scope Templates Prefixes Links Options Policies Clients Client-Classes VPNs Networks Failover DNS LDAP Extensions Traps DHCP Server		
Manage DHCP Lease <i>192.168.50.101</i>		
Attribute	Value	
Client Identifier	null	
Reservation Lookup Key		
Attribute	Value	Data Type
State <small>(state)</small>	available	32-bit enum
MAC Address <small>(client-mac-addr)</small>	<input type="text"/>	MAC address
Hostname <small>(client-host-name)</small>	<input type="text"/>	string
Flags <small>(flags)</small>	<input type="checkbox"/> reserved <input type="checkbox"/> valid <input type="checkbox"/> deactivated <input type="checkbox"/> initialized <input checked="" type="checkbox"/> failover-updated <input type="checkbox"/> not_in_ranges <input type="checkbox"/> dynamic <input type="checkbox"/> backup	flags
Expiration <small>(expiration)</small>	none	date
<input checked="" type="checkbox"/> Advanced		
Lookup Key	MAC Address	
<input type="text"/>	<input checked="" type="radio"/> MAC address <input type="radio"/> string <input type="radio"/> binary	<input type="text"/>
<input type="button" value="Deactivate"/> <input type="button" value="Make Reservation"/> <input type="button" value="Cancel"/>		

149417

In the CLI, use **report** to display the IP address usage for specified servers. See the *Cisco CNS CLI Reference Guide* for additional options you can set.

**Tip**

If you are not already using **lease-notification** in an automated way, try **lease-notification available=100%** for a concise scope-by-scope summary of the state of the servers.

Running IP Lease Histories

You can extract IP lease history data from a special database so that you can determine past allocation information for a given IP address. You can get a historical view of when a client was issued a lease, for how long, when the client or server released the lease before it expired, and if and when the server renewed the lease and for how long.

Network Registrar provides a client to control querying IP history data. Through this client, you can:

- Get the MAC addresses associated with a given IP address over a given time.
- See the entire IP history database as a comma-separated file.
- View the attributes of the lease history (the lease history detail report)—See the [“Querying IP Lease History Using the Web UI”](#) section on page 21-15.

You must use additional administrative functions to trim the IP history database of records, to keep the size of the database from growing without bounds.

**Note**

When the state of an existing lease changes (for example, when it is configured as a reserved address or it is deactivated), the change does not appear as a lease history change at the regional cluster, unless the *ip-history-detail* attribute is enabled. With detail collection disabled, a lease history change appears only when the lease transitions from leased to not leased or is assigned to another client.

Enabling Lease History Recording at the Local Cluster

You must explicitly enable lease history recording for the local cluster DHCP server. The DHCP server logs IP history recording errors in the usual DHCP log files.

-
- Step 1** In the Web UI, click **DHCP**, then **DHCP Server** to open the Manage DHCP Server page.
- Step 2** Click the **Local DHCP Server** link.
- Step 3** On the Edit DHCP Server page, look for the Lease History attributes:
- *Lease History (ip-history)*—Be sure this is set to enabled.
 - *ip-history-detail*—Enable this to get detailed lease history data.
 - *ip-history-max-age*—Maximum age of the lease history to collect. With lease history enabled, the DHCP server periodically examines the lease history records and deletes any records with lease history bindings older than this age threshold.
- In the CLI, you must explicitly enable recording IP (lease) history for addresses (**dhcp enable ip-history**).
- Step 4** Click **Modify Server** at the bottom of the page.
- Step 5** If you set staged edit mode for the scope, reload the server.
-

Querying IP Lease History Using the Web UI

Once you have leases, you can query for their history. You set up the local cluster containing the DHCP server as part of the regional cluster, and enable polling for the lease history data from the regional cluster (see the [“Enabling Lease History Collection”](#) section on page 5-10).

You can adjust the polling criteria for the cluster in the regional cluster Web UI using the attributes described in the “[Polling Subnet Utilization and Lease History Data](#)” section on page 5-8.

You must also set the selection criteria for querying the lease history data:

Step 1 In the regional Web UI, click **Address Space**, then **Lease History** to open the Query Lease History page. Using this functionality requires DHCP rights locally and address space rights at the regional cluster.

Step 2 You can query lease history based on the following criteria:

- The virtual private network (VPN) for the addresses to be polled for lease data—A VPN choice is available only if at least one was defined or pulled to the regional cluster (see the “[Managing Virtual Private Networks](#)” section on page 5-14). By default, the query is based on no specific VPN unless you choose it from the VPN drop-down list on the page. You can also query based on all VPNs.
- Time range for the query—Choose from one of the following time ranges for the lease history data:
 - last 10 days
 - last 30 days
 - last 60 days
 - last 90 days
 - from/to (limited to 90 days)

If you choose this value, also choose the Start Date and End Date month, day, and year from the drop-down lists. The result depends on the value of the *poll-lease-hist-interval* attribute. If the time range is set back one month, but you set the polling interval to greater than a month, no data will appear.

- Criteria—Choose the criteria on which you want to base the query (per VPN and time range):
 - By IP Address—Enter the IP address in the adjacent field.
 - By MAC Address—Enter the MAC address in the adjacent field.
 - By IP Range—Enter the IP address range in the adjacent field, the start of range in the left field and end of range in the right field.
 - All—Choose all the leases by no particular criteria.
 - Current Lease by IP—Show the current lease for an IP address (entered in the adjacent field).



Note The regional CCM server references the DHCP server to obtain the most recent lease data for the IP address. Therefore, the regional address space must include the matching subnet from the local cluster, and the particular DHCP server must be running.

Step 3 After entering or choosing these values, click **Query Lease History**.



Tip

At the upper-left corner of the List Lease History Records page is either the Log icon (📄) for the Netscape browser that you can click to view a text version of the report, or the Save icon (💾) for the Internet Explorer browser so that you can save the report to a file (.txt by default). The List Lease History page also has a View Detail column with the View icon (🔍) if you set the CCM server with the *ip-history-detail* attribute enabled. Click this icon to open the View Lease History Detail page. This page shows the change set for each history record.

Querying IP Lease History Using the iphist Utility

You can query the IP history database at the local cluster and direct the results to standard output or a file by using the **iphist** utility. You must run this utility on the same machine as the DHCP server, and you must have superuser/root privileges to read and modify the database file. The default location is:

- On Windows—\Program Files\Network Registrar\bin
- On Solaris and Linux—/opt/nwreg2/usrbin

From the command prompt, change to the location and run the utility using the syntax:

```
iphist [options] {ipaddress | all} [start-date | start [end-date | end]]
```

The IP address is a single address or the keyword **all**, the start date is in local time or the keyword **start** for the earliest date in the database, and the end date is in local time or the keyword **end** for the last date in the database. However, the output is in Greenwich Mean Time (GMT) by default, unless you use the **-l** option to specify local time.

The full list of command options appears in [Table 21-1](#).

Table 21-1 iphist Command Options

Option	Description
-N <i>username</i>	Administrator username. If omitted, you are prompted for the username.
-P <i>password</i>	Administrator password. If omitted, you are prompted for the password.
-C <i>cluster[:port]</i>	Destination server and optional SCP port.
-a	Shows the lease attributes, visibility 5 and 3.
-b	Displays the local and backup server failover leases.
-f " <i>format</i> "	Format of the output lines. The default format is: "address,client-mac-addr,binding-start-time,binding-end-time"
-l	Displays output in local time rather than the default GMT.
-m	Displays the local and main server failover leases.
-n <i>vpn</i>	Name or ID of an associated VPN, or the word all (for all VPNs) or global (for addresses without a VPN). If omitted, the query is based on the global VPN, or the current one set by the session set current-vpn command, unless you use the all value with the option.
-o <i>file</i>	Sends output to a file.
-v	Displays the database version.
-V <i>visibility</i>	Sets the visibility level of the output attributes. The visibility is 3 by default.
-z <i>debug-args</i>	Sets the debug output levels.

Dates can use this syntax (quotes are required if space characters are included):

- *month/day/year@hour:min:sec* (for example, 8/28/2001@10:01:15), with the time optional
- *month/day/year hour:min:sec* (for example, "8/28/2001 10:01:15"), with the time optional
- *month day year hour:min:sec* (for example, "Aug 28 2001 10:01:15"), with the time optional
- Keywords **start**, **end**, or **now** (for the current time)

The date filtering is intended to limit the output to leases that were active during that time. This means that they can begin before the specified start date, as long as they do not end before the start date. They can also not begin after the specified end date. For example, invoking the command:

```
# ./iphist -N user -P password all Aug 28 2003 Dec 31 2003
```

for the following leases:

```
Lease 1   Begin   Jan 01 2003   End   Jun 30 2003
Lease 2   Begin   Mar 10 2003   End   Sep 01 2003
Lease 3   Begin   Jun 01 2003   End   Sep 30 2003
Lease 4   Begin   Jan 01 2004   End   Mar 10 2004
```

would return just Lease 2 and Lease 3, because they both end after the specified start date of the query, even though they both begin before that date. The other two are out of range, because they either end before the specified start date or begin after the specified end date of the query.

The values on each line depend on the specific lease object that the DHCP server stores. You can specify the values to include using the **iphist -f format** command. The *format* argument is a quote-enclosed and comma-separated list of lease attribute names that provides the template for the output lines. The default output is *ipaddress,client-mac-addr,binding-start-time, binding-end-time*.

For example:

```
# ./iphist -f "address,client-mac-addr,binding-start-time,binding-end-time" all
```

The output is a sequence of lines terminated with a newline sequence appropriate to the operating system ($\backslash n$ on UNIX or $\r\backslash n$ on Windows). Each line contains data on a single lease record. The format of the lines is generally comma-separated values enclosed in quotes. Within quotes, literal backslash (\backslash) or quote ($"$) characters are escaped with a single backslash (\backslash). New lines in attributes are printed as $\backslash n$.

[Table 21-2](#) lists some of the common lease object attributes you can include in the output. Also, see the **lease** command in the *Cisco CNS Network Registrar CLI Reference Guide*. To get a full list, use **iphist -a**.

Table 21-2 IP History Query Output Attributes

Lease Attribute	Description
address	IP address of the lease.
binding-start-time	Start time of the lease binding.
binding-end-time	End time of the lease binding.
client-binary-client-id	Binary form of the client's MAC address.
client-dns-name	Latest DNS name of the client known by the DHCP server.
client-domain-name	Domain where the client resides.
client-flags	A number of client flags.
client-host-name	Host name the client requested.
client-id	Client ID requested by or synthesized for the client.
client-last-transaction-time	Date and time when the client most recently contacted the server.
client-mac-addr	MAC address that the client presented to the DHCP server.
client-os-type	Operating system of the leased client.
expiration	Date and time when the lease expires.
flags	Either reserved or deactivated.
lease-renewal-time	Minimal time that the client is expected to issue a lease renewal.

Table 21-2 IP History Query Output Attributes (continued)

Lease Attribute	Description
relay-agent-circuit-id	Contents of the <i>circuit-id</i> suboption (1).
relay-agent-option	Contents of the option from the most recent client interaction.
relay-agent-remote-id	Contents of the <i>remote-id</i> suboption (2).
relay-agent-server-id-override	Address in the <i>server-id-override</i> suboption.
relay-agent-subnet-selection	Address in the <i>subnet-selection</i> suboption.
relay-agent-vpn-id	Contents of the <i>vpn-id</i> suboption.
start-time-of-state	Date and time when the lease changed its state.
state	One of available, expired, leased, offered, or unavailable.
vendor-class-id	Vendor class ID requested by the client.
vpn-id	Identifier for the VPN, if any.

Trimming Lease History Data

If you enabled IP history trimming at the regional cluster, the IP history database is automatically trimmed so that you can reclaim disk space. Each history record has an expiration time. Trimming is necessary for the DHCP server itself, as well as the CCM regional server that polls the DHCP server for history data.

The CCM server performs background trimming at the regional cluster, which trims off the lease history data older than a certain age at regular intervals. The trimming interval is set by default to 24 hours, and the age (how far back to go in time before trimming) to 24 weeks. The DHCP server at the local cluster performs daily automatic trimming (at 3:00 A.M. local time), and stores four weeks of data by default.

In the regional cluster Web UI, you must be a central configuration administrator.

-
- Step 1** Click **Servers** to open the Manage Servers page.
- Step 2** Click the **Local CCM Server** link to open the Edit CCM Server page.
- Step 3** Under Lease History Settings, set the following attributes (you can use the **s**, **m**, **h**, **d**, **w**, **m**, or **y** suffix with values you enter):
- *trim-lease-hist-interval*—How often to trim the old lease history data automatically, the default being daily. If set to 0, no automatic lease trimming occurs, which is not recommended due to the increasing disk space used. The bounded values are 0 to one year.
 - *trim-lease-hist-age*—Provided that the *trim-lease-hist-interval* is not set to 0, how far back in time to trim the old lease history data automatically, the default being 24 weeks. The bounded values are one day to one year.
- Step 4** To force immediate trimming, at the bottom of the page find the Trim/Compact Inputs section (compacting is available only for subnet utilization data). Set the Trim/Compact age to a desired value. This age is how far in time to go back to trim the lease history data. There are no bounds to this value. However, if you set a very small value (such as 1m), it trims or compacts very recent data, which can be undesirable. In fact, if you set it to zero, you lose all of the collected data. Setting the value too high (such as 10y) may end up not trimming or compacting any data.
- Step 5** If you are trimming immediately, click **Trim All Lease History**.
-

You can adjust the trimming that the DHCP server itself performs by setting the *ip-history-max-age* attribute. If *ip-history* is enabled, the DHCP server accumulates database records over time as lease bindings change. This parameter establishes a limit on the age of the history records kept in the database. The server periodically examines the lease history records, establishes an age threshold based on this parameter, and deletes any records that represent bindings that ended before the threshold. The default value is four weeks.

Running Lease Utilization Reports

Using the Web UI, you can view the current utilization for address blocks, subnets, and scopes from pages in the Address Space function.

In the CLI, **report** serves the same function and displays the same output.

For both user interfaces, see the “[Generating Subnet Utilization History Reports](#)” section on page 8-13.

Receiving Lease Notification

The CLI provides the feature of sending notifications if the number of available addresses equals or falls below a certain threshold. The **lease-notification** command specifies, through an *available* attribute, when the notification should occur if the number of available leases reaches or falls below a certain threshold. You can e-mail the report to a user. Although you can use the command interactively, its primary use is in an automated procedure such as a UNIX **cron** task or Windows Scheduled Task.

The following example sets up lease notification for *examplescope* for when its free addresses fall to 10%. It sends the report to recipients *billy*, *joe*, and *jane*, on a specific Windows mail host:

```
nrcmd> lease-notification available=10% scopes=examplescope recipients=billy,joe,jane
mail-host=mailhost
```

The output consists of an explanatory header, a table containing a row for each scope in which the number of free addresses is equal to or less than the threshold, and possible warnings related to the scopes and clusters requested.

Network Registrar uses the default cluster and the *.nrconfig* file by default, unless you specify otherwise. For the command syntax, see the **lease-notification** command in the *Network Registrar CLI Reference*.

Running Lease Notification Automatically in Solaris and Linux

You can run **lease-notification** periodically by means of the **cron(1)** command by supplying **crontab(1)** with the command to run. This example, specified to **crontab**, runs **lease-notification** at 00:15 and 12:15 (15 minutes after midnight and noon), Monday through Friday (note that this encompasses a single command line):

```
15 0,12 * * 1-5 . .profile; /opt/nwreg2/usrbin/nrcmd lease-notification available=10\%
config=/home/jsmith/.nrconfig addresses=192.32.1.0-192.32.128.0
recipients=jsmith,jdoe@example.com >/dev/null 2>&1
```

You can perform **crontab** editing by running the UNIX **crontab -e** command. Set your **EDITOR** environment variable before running the command, unless you want to use **ed(1)**. See the **crontab(1)** man page for additional details.

Note that you must supply the CLI command’s full path on the **crontab** command line. You can determine the full path in your environment with the UNIX **which nrcmd** command.

Also, when you run the **lease-notification** command by means of **crontab**, the **nrcmd** command ignores the user environment variables **CNR_CLUSTER**, **CNR_NAME**, and **CNR_PASSWORD**. Because other viewers can view the command being run, do not provide the password through the **-P** option on the command line, for security reasons.

Supply the cluster name, user, and password information for the cluster you want the **nrcmd** command to run from in a **.profile** or other file in the home directory of the user running **crontab -e**. For example:

```
CNR_CLUSTER=host1
export CNR_CLUSTER
CNR_NAME=admin1
export CNR_NAME
CNR_PASSWORD=passwd1
export CNR_PASSWORD
```

The **.profile** specification in the **crontab** entry explicitly reads the file. The first dot (.) is the shell command that reads the file and you must follow it with white space. For notification on a different cluster, or clusters, than the one on which **nrcmd** is running, specify this information:

- Clusters to check in a config file (see the “[Specifying Configuration Files for Lease Notification](#)” section on page 21-21).
- Fully specified path as in sample **crontab** entry at the beginning of this section.

You can prevent others from examining or changing the contents of the **.profile** and the configuration file that you create by changing its permissions with the **chmod go-rwx config-file** UNIX command.

Running Lease Notification Automatically in Windows

Use the Scheduled Tasks service available in Windows Explorer under My Computer to schedule the **lease-notification** command. If you do not find a Scheduled Tasks folder under My Computer, you need to add this optional component from Microsoft Internet Explorer 4.0 or later, or use some third-party task scheduler. You can also use the **at** command to schedule the **nrcmd lease-notification** command. Put multiple entries in the **at** queue, one for each time of day at which you want to run the job.

Specifying Configuration Files for Lease Notification

If you omit a configuration file, **lease-notification** looks for a default **.nrconfig** file in your current directory, then in your home directory, and finally in the **CNR_INSTALL_PATH/conf** directory. Network Registrar uses the first file it encounters. Each line of the file must either begin with the character **#** (comment), a section header enclosed in square brackets, or a parameter/value pair or its continuation. Network Registrar strips leading white space from each line and ignores blank lines.

Querying Leases

Network Registrar can work together with Cisco routers to provide enhanced provisioning capabilities. Part of the implementation of the Cisco uBR access concentrator’s relay agent is to capture and glean information from DHCP lease requests and responses. It does this so that it can:

- Associate subscriber cable modem and client MAC addresses with server-assigned IP addresses.
- Verify source IP addresses in upstream datagrams.
- Encrypt unicast downstream traffic through the DOCSIS Baseline Privacy protocol.
- Avoid broadcasting downstream Address Resolution Protocol (ARP) requests, which can burden the the uBR as well as the subscriber hosts, and which can be compromised by malicious clients.

The uBR device does not capture all DHCP state information through gleaning. The uBR device cannot glean from unicast messages (particularly renewals and releases) because capturing them requires special processing that would degrade the uBR device's forwarding performance. Also, this data does not persist across uBR reboots or replacements. Therefore, the only reliable source of DHCP state information for the uBR device is the DHCP server itself.

For this reason the DHCP server supports the DHCPLEASEQUERY message, which is similar to a DHCPINFORM message. The DHCPLEASEQUERY message's number in the *dhcp-message-type* option (53) is 13. The uBR device's relay agent sends a DHCPLEASEQUERY request to the DHCP server. The request always yields either a DHCPACK or DHCPNAK response from the server. A DHCP server that does not support this type of message is likely to drop the DHCPLEASEQUERY packet.

Lease Query Requests

In a DHCPLEASEQUERY request, the gateway IP address (*giaddr*) field must have the IP address of the requesting relay agent. The *giaddr* field is independent of the client address (*ciaddr*) searched and is simply the return address for any responses from the server. The relay agent can send one of these types of DHCPLEASEQUERY requests:

- By IP address—The request packet includes an IP address in the client address (*ciaddr*) field. The DHCP server returns data for the most recent client to use that address. A packet that includes a *ciaddr* value must be a request by IP address, despite the values in the MAC address fields (*htype*, *hlen*, and *chaddr*) or the *client-id* option. This is generally the most efficient and should be the most widely used query method.
- By MAC address—The request packet includes a MAC address in the hardware type (*htype*), address length (*hlen*), and client hardware address (*chaddr*) fields, and no value in the client address (*ciaddr*) field. The DHCP server returns all the IP addresses and most recent lease data for this MAC address in the *associated-ip* option of the DHCPLEASEQUERY packet. See the “Lease Query Responses” section.
- By *dhcp-client-identifier* option (61)—The request packet includes a *dhcp-client-identifier* option value. The DHCP server returns a DHCPACK packet containing the IP address data for the most recently accessed client. If the request does not also include a MAC address, the server returns all addresses and their data for the requested *dhcp-client-identifier*. If the request includes the MAC address, the server matches the *dhcp-client-identifier* and MAC address with the client data for the IP address and returns that data in the client address (*ciaddr*) field or *associated-ip* option (161).

Lease Query Responses

The *parameter-request-list* option (55) in a DHCPACK response from the server should contain data pertinent to the requestor, such as *dhcp-lease-time* (51) and *relay-agent-info* (82) option values:

- The *dhcp-lease-time* (51) value is for the remaining time until the lease expires.
- The *relay-agent-info* (82) value is from the most recent DHCPDISCOVER or DHCPREQUEST message from the client to the server.

Network Registrar provides these options for responding to DHCPLEASEQUERY messages:

- *cisco-client-last-transaction* (163)—Contains the most recent time a server accessed the client.
- *cisco-client-requested-host-name* (162)—Contains the host name that the client requested in the *host-name* option (12) or *client-fqdn* option (81).
- *cisco-associated-ip* (161)—Contains data on all the IP addresses associated with the client.

Lease Query and Reservations

The DHCP server includes lease reservation data in DHCPLEASEQUERY responses. In previous releases of Network Registrar, the DHCPLEASEQUERY responses were possible only for leased or previously leased clients for which the MAC address was stored. Cisco uBR relay agents, for example, discard DHCPLEASEQUERY datagrams not having a MAC address and lease time (*dhcp-lease-time* option).

Network Registrar returns a default lease time of one year (31536000 seconds) for reserved leases in a DHCPLEASEQUERY response. If the address is actually leased, Network Registrar returns its remaining lease time.

