redhat.

Chapter 18. Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP) is network protocol for automatically assigning TCP/IP information to client machines. Each DHCP client connects to the centrally-located DHCP server which returns that client's network configuration including IP address, gateway, and DNS servers.

18.1. Why Use DHCP?

DHCP is useful for fast delivery of client network configuration. When configuring the client system, the administrator can choose DHCP and not have to enter an IP address, netmask, gateway, or DNS servers. The client retrieves this information from the DHCP server. DHCP is also useful if an administrator wants to change the IP addresses of a large number of systems. Instead of reconfiguring all the systems, he can just edit one DHCP configuration file on the server for the new set of IP address. If the DNS servers for an organization changes, the changes are made on the DHCP server, not on the DHCP clients. Once the network is restarted on the clients (or the clients are rebooted), the changes will take effect.

Furthermore, if a laptop or any type of mobile computer is configured for DHCP, it can be moved from office to office without being reconfigured as long as each office has a DHCP server that allows it to connect to the network.

18.2. Configuring a DHCP Server

You can configure a DHCP server using the configuration file `/etc/dhcpd.conf`.

DHCP also uses the file `/var/lib/dhcp/dhcpd.leases` to store the client lease database. Refer to Section 18.2.2 *Lease Database* for more information.

18.2.1. Configuration File

The first step in configuring a DHCP server is to create the configuration file that stores the network information for the clients. Global options can be declared for all clients, or options can be declared for each client system.

The configuration file can contain any extra tabs or blank lines for easier formatting. The keywords are case-insensitive, and lines beginning with a hash mark (#) are considered comments.

Two DNS update schemes are currently implemented — the ad-hoc DNS update mode and the interim DHCP-DNS interaction draft update mode. If and when these two are accepted as part of the IETF standards process, there will be a third mode — the standard DNS update method. The DHCP server must be configured to use one of the two current schemes. Version 3.0b2pl11 and previous version used the ad-hoc mode; however, it has been depreciated. If you want to keep the same behavior, add the following line to the top of the configuration file:

```
ddns-update-style ad-hoc;
```

To use the recommended mode, add the following line to the top of the configuration file:

```
ddns-update-style interim;
```

Read the `dhcpd.conf` man page for details about the different modes.

There are two types of statements in the configuration file:

- Parameters — state how to perform a task, whether to perform a task, or what network configuration options to send to the client.
- Declarations — describe the topology of the network, describe the clients, provide addresses for the clients, or apply a group of parameters to a group of declarations.

Some parameters must start with the `option` keyword and are referred to as options. Options configure DHCP options; whereas, parameters configure values that are not optional or control how the DHCP server behaves.

Parameters (including options) declared before a section enclosed in curly brackets ({ }) are considered global parameters. Global parameters apply to all the sections below it.

Important

> If you change the configuration file, the changes will not take effect until you restart the DHCP daemon with the command `service dhcpd restart`.

In Example 18-1, the `routers`, `subnet-mask`, `domain-name`, `domain-name-servers`, and `time-offset` options are used for any `host` statements declared below it.

As shown in Example 18-1, you can declare a `subnet`. You must include a `subnet` declaration for every subnet in your network. If you do not, the DHCP server will fail to start.

In this example, there are global options for every DHCP client in the subnet and a `range` declared. Clients are assigned an IP address within the `range`.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
        option routers                  192.168.1.254;
        option subnet-mask              255.255.255.0;

        option domain-name              "example.com";
        option domain-name-servers       192.168.1.1;

        option time-offset              -18000;     # Eastern Standard Time

 range 192.168.1.10 192.168.1.100;
}
```

Example 18-1. Subnet Declaration

All subnets that share the same physical network should be declared within a `shared-network` declaration as shown in Example 18-2. Parameters within the `shared-network` but outside the enclosed `subnet` declarations are considered global parameters. The name of the `shared-network` should be a descriptive title for the network such as test-lab to describe all the subnets in a test lab environment.

```
shared-network name {
    option domain-name              "test.redhat.com";
    option domain-name-servers      ns1.redhat.com, ns2.redhat.com;
    option routers                  192.168.1.254;
    more parameters for EXAMPLE shared-network
    subnet 192.168.1.0 netmask 255.255.255.0 {
        parameters for subnet
        range 192.168.1.1 192.168.1.31;
    }
    subnet 192.168.1.32 netmask 255.255.255.0 {
        parameters for subnet
        range 192.168.1.33 192.168.1.63;
    }
}
```

Example 18-2. Shared-network Declaration

As demonstrated in Example 18-3, the `group` declaration can be used to apply global parameters to a group of declarations. You can group shared networks, subnets, hosts, or other groups.

```
group {
   option routers                  192.168.1.254;
   option subnet-mask              255.255.255.0;

   option domain-name              "example.com";
   option domain-name-servers       192.168.1.1;

   option time-offset              -18000;     # Eastern Standard Time

   host apex {
      option host-name "apex.example.com";
      hardware ethernet 00:A0:78:8E:9E:AA;
      fixed-address 192.168.1.4;
   }

   host raleigh {
      option host-name "raleigh.example.com";
      hardware ethernet 00:A1:DD:74:C3:F2;
      fixed-address 192.168.1.6;
   }
}
```

Example 18-3. Group Declaration

To configure a DHCP server that leases a dynamic IP address to a system within a subnet, modify Example 18-4 with your values. It declares a default lease time, maximum lease time, and network configuration values for the clients. This example assigns IP addresses in the `range` 192.168.1.10 and 192.168.1.100 to client systems.

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "example.com";

subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.10 192.168.1.100;
}
```

Example 18-4. Range Parameter

To assign an IP address to a client based on the MAC address of the network interface card, use the `hardware ethernet` parameter within a `host` declaration. As demonstrated in Example

18-5, the `host apex` declaration specifies that the network interface card with the MAC address 00:A0:78:8E:9E:AA always receives the IP address 192.168.1.4.

Notice that you can also use the optional parameter `host-name` to assign a host name to the client.

```
host apex {
   option host-name "apex.example.com";
   hardware ethernet 00:A0:78:8E:9E:AA;
   fixed-address 192.168.1.4;
}
```

Example 18-5. Static IP Address using DHCP

Tip

> You can use the sample configuration file in Red Hat Linux 9 as a starting point and then add your own custom configuration options to it. Copy it to its proper location with the command
>
> ```
> cp /usr/share/doc/dhcp-<version-number>/dhcpd.conf.sample /etc/dhcpd.conf
> ```
>
> (where <*version-number*> is the DHCP version you are using).

For a complete list of option statements and what they do, refer to the dhcp-options man page.

18.2.2. Lease Database

On the DHCP server, the file `/var/lib/dhcp/dhcpd.leases` stores the DHCP client lease database. This file should not be modified by hand. DHCP lease information for each recently assigned IP address is automatically stored in the lease database. The information includes the length of the lease, to whom the IP address has been assigned, the start and end dates for the lease, and the MAC address of the network interface card that was used to retrieve the lease.

All times in the lease database are in Greenwich Mean Time (GMT), not local time.

The lease database is recreated from time to time so that it is not too large. First, all known leases are saved in a temporary lease database. The `dhcpd.leases` file is renamed `dhcpd.leases~`, and the temporary lease database is written to `dhcpd.leases`.

The DHCP daemon could be killed or the system could crash after the lease database has been renamed to the backup file but before the new file has been written. If this happens, there is no `dhcpd.leases` file that is required to start the service. Do not create a new lease file if this occurs. If you do, all the old leases will be lost and cause many problems. The correct solution is to rename the `dhcpd.leases~` backup file to `dhcpd.leases` and then start the daemon.

18.2.3. Starting and Stopping the Server

Important

> Before you start the DHCP server for the first time, it will fail unless there is an existing `dhcpd.leases` file. Use the command `touch /var/lib/dhcp/dhcpd.leases` to create the file if it does not exist.

To start the DHCP service, use the command `/sbin/service dhcpd start`. To stop the DHCP server, use the command `/sbin/service dhcpd stop`. If you want the daemon to start automatically at boot time, see Chapter 14 *Controlling Access to Services* for information on how to manage services.

If you have more than one network interface attached to the system, but you only want the DHCP server to start on one of the interface, you can configure the DHCP server to start only on that device. In `/etc/sysconfig/dhcpd`, add the name of the interface to the list of `DHCPDARGS`:

```
# Command line options here
DHCPDARGS=eth0
```

This is useful if you have a firewall machine with two network cards. One network card can be configured as a DHCP client to retrieve an IP address to the Internet. The other network card can be used as a DHCP server for the internal network behind the firewall. Specifying only the network card connected to the internal network makes the system more secure because users can not connect to the daemon via the Internet.

Other command line options that can be specified in `/etc/sysconfig/dhcpd` include:

- `-p` *<portnum>* — Specify the udp port number on which dhcpd should listen. The default is port 67. The DHCP server transmits responses to the DHCP clients at a port number one greater than the udp port specified. For example, if you accept the default of port 67, the server listens on port 67 for requests and responses to the client on port 68. If you specify a port here and use the DHCP relay agent, you must specify the same port on which the DHCP relay agent should listen. See Section 18.2.4 *DHCP Relay Agent* for details.
- `-f` — Run the daemon as a foreground process. This is mostly used for debugging.
- `-d` — Log the DHCP server daemon to the standard error descriptor. This is mostly used for debugging. If this is not specified, the log is written to `/var/log/messages`.
- `-cf` *filename* — Specify the location of the configuration file. The default location is `/etc/dhcpd.conf`.
- `-lf` *filename* — Specify the location of the lease database file. If a lease database file already exists, it is very important that the same file be used every time the DHCP server is started. It is strongly recommended that this option only be used for debugging purposes on non-production machines. The default location is `/var/lib/dhcp/dhcpd.leases`.
- `-q` — Do not print the entire copyright message when starting the daemon.

18.2.4. DHCP Relay Agent

The DHCP Relay Agent (`dhcrelay`) allows you to relay DHCP and BOOTP requests from a subnet with no DHCP server on it to one or more DHCP servers on other subnets.

When a DHCP client requests information, the DHCP Relay Agent forwards the request to the list of DHCP servers specified when the DHCP Relay Agent is started. When a DHCP server returns a reply, the reply is broadcast or unicast on the network that sent the original request.

The DHCP Relay Agent listens for DHCP requests on all interfaces unless the interfaces are specified in `/etc/sysconfig/dhcrelay` with the `INTERFACES` directive.

To start the DHCP Relay Agent, use the command `service dhcrelay start`.

18.3. Configuring a DHCP Client

The first step for configuring a DHCP client is to make sure the kernel recognizes the network interface card. Most cards are recognized during the installation process, and the system is configured to use the correct kernel module for the card. If you install a card after installation, **Kudzu** [1] should recognize it and prompt you to configure the corresponding kernel module for it. Be sure to check the Red Hat Linux Hardware Compatibility List available at http://hardware.redhat.com/hcl/. If the network card is not configured by the installation program or **Kudzu** and you know which kernel module to load for it, refer to Chapter 31 *Kernel Modules* for details on loading kernel modules.

To configure a DHCP client manually, you need to modify the `/etc/sysconfig/network` file to enable networking and the configuration file for each network device in the `/etc/sysconfig/network-scripts` directory. In this directory, each device should have a configuration file named `ifcfg-eth0` where `eth0` is the network device name.

The `/etc/sysconfig/network` file should contain the following line:

```
NETWORKING=yes
```

You might have more information in this file, but the `NETWORKING` variable must be set to `yes` if you want networking to start at boot time.

The `/etc/sysconfig/network-scripts/ifcfg-eth0` file should contain the following lines:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

You need a configuration file for each device that you want to configure to use DHCP.

If you prefer a graphical interface for configuring a DHCP client, refer to Chapter 12 *Network Configuration* for details on using **Network Administration Tool** to configure a network interface to use DHCP.

18.4. Additional Resources

For configuration options not covered here, please refer to the following resources.

18.4.1. Installed Documentation

- `dhcpd` man page — describes how the DHCP daemon works
- `dhcpd.conf` man page — explains how to configure the DHCP configuration file; includes some examples
- `dhcpd.leases` man page — explains how to configure the DHCP leases file; includes some examples
- `dhcp-options` man page — explains the syntax for declaring DHCP options in `dhcpd.conf`; includes some examples
- `dhcrelay` man page — explains the DHCP Relay Agent and its configuration options.

[1] **Kudzu** is a hardware probing tool run at system boot time to determine what hardware has been added or removed from the system.