

Configuring a DHCP Server

Dru Lavigne

Unlike the built-in dhclient, your FreeBSD system does not come with DHCP server software. This is because you only need to configure a DHCP server if you want to lease out IP configuration for your own network.

However, there are two ports that allow you to create your own DHCP server. The first is known as WIDE, or Widely Integrated Distributed Environment. As the name suggests, it has been optimized for very large networks, so I won't cover it in this series. The second is from the ISC, or Internet Software Consortium, and can be found here.

The dhclient that comes with FreeBSD is also from the ISC. The site has some good information regarding DHCP, including a FAQ and some introductory tutorials.

Before building the DHCP server port, ensure that the bpf device is built into your kernel. If you are using the default kernel, do a search through the default kernel configuration file:

```
$ grep bpf /usr/src/sys/i386/conf/GENERIC
# The `bpf' device enables the Berkeley Packet Filter.
device          bpf                # Berkeley packet filter
```

If you're not using the default kernel, substitute the name of your custom kernel configuration file for "GENERIC." If you don't get anything back when you grep for bpf, add that line to your kernel configuration file and rebuild your kernel.

Once you have the bpf device, build the DHCP server port:

```
# cd /usr/ports/net/isc-dhcp3
# make install clean
```

This build will install several files. Let's take a quick overview. First, you'll get four executables:

- /usr/local/sbin/dhclient will be the latest edition of the DHCP client; the DHCP client that came with your FreeBSD system is located in /sbin/dhclient. On a side note, system binaries (applications) are always located in a directory called/sbin. If the binary came installed with the operating system, it will be in the/sbin located on the root (/sbin). If the binary was installed as a port or a package, it will be located in /usr/local/sbin.
- /usr/local/sbin/dhcpd is the actual DHCP server application. Like most servers, or daemons, it ends in "d".
- /usr/local/sbin/dhcrelay is the bootp relay agent. You may remember from the last article that you will only need this application if you have multiple network segments and don't want to create a DHCP server on every segment.
- /usr/local/bin/omshell is the OMAPI command shell. This application allows you to make changes to the DHCP server while it is running. You don't have to stop and restart DHCP in order for the changes to take effect.

Configuring a DHCP Server

Dru Lavigne

You'll also get two sample startup scripts and a sample configuration script to get started with your own configurations:

- `/usr/local/etc/rc.d/isc-dhcpd.sh.sample`
Sample startup script for the DHCP server.
- `/usr/local/etc/rc.d/isc-dhcrelay.sh.sample`
Sample startup script for the bootp relay agent.
- `/usr/local/etc/dhclient.conf.sample`
Sample configuration script for the DHCP server.

To aid in your configuration, the following manpages are installed:

- `man dhcpd`
- `man dhcrelay`
- `man dhcpd.leases`
- `man dhcpd.conf`

And finally, a documents directory:

`/usr/local/share/doc/isc-dhcp3/`

The Configuration File

Let's start by taking a look at the configuration file for the DHCP server. You should leave the sample as is, and copy it over to the file that you will edit:

```
$ cp /usr/local/etc/dhcpd.conf.sample /usr/local/etc/dhcpd.conf
```

Let's go through each line of this file to make sure you understand all of the options; then we'll customize it for a sample network.

```
$ more /usr/local/etc/dhcpd.conf
```

```
# dhcpd.conf
# Sample configuration file for ISC dhcpd
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;
```

Each bit of information a DHCP server leases to a client is known as an "option." Some options are considered to be "global," meaning that every DHCP client in the network will receive that option as part of their lease. Some options are considered to be "local" to a specific subnet. For example, the option for the IP address of the default gateway will always be "local," as a default gateway must live on the same subnet as the client. However, the two above options are considered to be "global," as every computer in your network will share the same domain name and will use the same DNS servers.

Configuring a DHCP Server

Dru Lavigne

```
default-lease-time 600;  
max-lease-time 7200;
```

Some DHCP client software requests a lease time. If the client doesn't, the server will assign the lease with the default-lease-time value. If the client does, the server will honor the request, but only up to the max-lease-time value. Both values are in seconds.

```
# If this DHCP server is the official DHCP server for the local  
# network, the authoritative directive should be uncommented.  
#authoritative;
```

This line should be uncommented, as it allows your DHCP server to send a DHCPNACK to misconfigured clients. An example of a misconfigured client would be a computer that was physically moved to another subnet without releasing its old lease.

```
# ad-hoc DNS update scheme - set to "none" to disable dynamic DNS updates.  
ddns-update-style ad-hoc;
```

The ddns-update-style parameter has three possible values. ad hoc has been deprecated and shouldn't be used. interim allows your DHCP server to update a DNS server whenever it hands out a lease. This way, your DNS server will know which IP addresses are associated with which computers in your network. In order for this to work, your DNS server must support DDNS (Dynamic DNS). If your DNS server doesn't support DDNS, or you don't want to take advantage of dynamic DNS, you should change this value to none.

```
# Use this to send dhcp log messages to a different log file (you also  
# have to hack syslog.conf to complete the redirection).  
#log-facility local7;
```

How you handle this option will affect where the DHCP server will send its logging information. local7 refers to a locally defined log file. Until you define that log file, the DHCP server will write all of its events to the system log file, or /var/log/messages.

```
# No service will be given on this subnet, but declaring it helps the  
# DHCP server to understand the network topology.
```

```
subnet 10.152.187.0 netmask 255.255.255.0 {  
}
```

Now we get to the meat of this file, the "subnet declarations." A DHCP server needs to know which network or subnet IDs your network contains. Additionally, for each network or subnet, it needs to know which "pool" of addresses it is allowed to lease out to the devices on that segment of the

Configuring a DHCP Server

Dru Lavigne

network. It is helpful to sketch out your network ahead of time, so you know which addresses are available for DHCP clients and which addresses are unavailable because they are already statically assigned. I'll walk through such a sketch with you in the next article, when I demonstrate a more complex network configuration.

In the meantime, it is important to "declare" each segment of your network, even if a segment does not contain any DHCP clients. This is the case in the above declaration for the subnet ID 10.152.187.0. Notice that the declaration includes the mask that matches the network ID, and is then followed by a pair of curly braces ({}). Let's compare this declaration to the next subnet declaration:

```
# This is a very basic subnet declaration.

subnet 10.254.239.0 netmask 255.255.255.224 {
    range 10.254.239.10 10.254.239.20;
    option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
}
```

This declaration is for the subnet 10.254.239.0. Within the curly braces is the "range" of IP addresses available to be leased. If you're familiar with classful subnet masking, you know that every IP address in your network must share the portion of the IP address that is masked by 255. In this example, there are three 255s in the mask, so every IP address in this network must start with the same three numbers: 10.254.239. The mask also contains a 224 in the last octet, which leaves a range of 30 possible valid addresses for each subnet represented by that octet. In this example, the DHCP server has been instructed to give out 11 of those possible valid addresses: 10 to 20.

The DHCP server has also been instructed to lease out two default gateway addresses. The closing curly brace indicates the end of the information to be leased out to each client.

The default configuration file continues on with several more examples of subnet declarations. I won't rehash them here; you'll notice as you read through them on your own that the examples vary in which options are to be leased to clients on each declared subnet.

A Sample Network

Now that we've had a chance to look through the default configuration file, let's configure a DHCP server for a simple network scenario. This sample network includes the following:

- A network ID of 192.168.10.0 255.255.255.0
- A domain name of mynetwork.com
- 10 DHCP clients on one network segment
- One default gateway with the address 192.168.10.1
- One DHCP server with the address 192.168.10.2
- Two DNS servers: 192.168.10.3 and 192.168.10.4

Note that the default gateway, DHCP server, and two DNS servers each have their own statically assigned address. It is important that the DHCP server is configured not to assign any of those addresses to the DHCP clients.

Configuring a DHCP Server

Dru Lavigne

I'll now create the following file:

```
# vi /usr/local/etc/dhcpd.conf
#my dhcp server configuration file
#first, the global options

option domain-name "mynetwork.com";
option domain-name-servers 192.168.10.3, 192.168.10.4;

default-lease-time 86400;
max-lease-time 86400;

authoritative;
ddns-update-style none;

#next, my one and only subnet

subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.5 192.168.10.20;
    option routers 192.168.10.1;
}
```

You'll note that I changed the lease time to 86400 seconds, or 24 hours. I kept the default logging facility and disabled DDNS. I also defined a range of addresses: 5-20. This bypasses the statically assigned addresses (1-4) and leaves room for another five computers, should this network segment ever experience growth. When you make your own configuration file, remember to place a ; at the end of each statement and to enclose your subnet declaration between opening and closing curly braces.

Now, let's see if the configuration file works. First, I'll start the daemon and watch for any error messages:

```
# dhcpd
Internet Software Consortium DHCP Server V3.0.1rc11
Copyright 1995-2003 Internet Software Consortium.
All rights reserved.
For info, please visit http://http://www.isc.org/products/DHCP
Wrote 0 leases to leases file.

Listening on BPF/de0/00:80:c8:3a:b8:46/192.168.10.0/24
Sending on   BPF/de0/00:80:c8:3a:b8:46/192.168.10.0/24
Sending on   Socket/fallback/fallback-net
```

While I'm at it, I should also rename the sample startup script and check its permissions; this way, the DHCP server will restart, should I ever reboot:

```
# mv /usr/local/etc/rc.d/isc-dhcpd.sh.sample /usr/local/etc/rc.d/isc-dhcpd.sh

# ls -l /usr/local/etc/rc.d/isc-dhcpd.sh
-r-xr-xr-x 1 root wheel 1662 Apr 13 10:32 /usr/local/etc/rc.d/isc-dhcpd.sh
```

Configuring a DHCP Server

Dru Lavigne

Good. The script is executable, so it's ready to do its thing. You can also run this script manually if you give it one of the following options: start, stop, restart, or status. For example:

```
# /usr/local/etc/rc.d/isc-dhcpd.sh status
root  1830  0.0  0.5  1784 1392  ??  Is  5:00PM  0:00.00  dhcpd
```

The restart option is very handy if you make a change to your configuration file. DHCP is one service that won't change its configuration if you simply send it a "signal one." Instead, you have to actually determine the PID of the process, send a signal 15 to terminate the process, then restart the process. Running the above script with restart will do all of that for you.

Okay, let's see if the DHCP server is actually handing out leases. I'll boot one of the machines on the network which has already been pre-configured as a DHCP client. Once it finishes booting, I'll check its lease file:

```
# more /var/db/client.leases

lease {
  interface "ed0";
  fixed-address 192.168.10.20;
  option subnet-mask 255.255.255.0;
  option routers 192.168.10.1;
  option dhcp-lease-time 86400;
  option dhcp-message-type 5;
  option domain-name-servers 192.168.10.3,192.168.10.4;
  option dhcp-server-identifier 192.168.10.1;
  option domain-name "mynetwork.com";
  renew 1 2003/4/21 08:50:05;
  rebind 1 2003/4/21 18:38:59;
  expire 1 2003/4/21 21:38:59;
}
```

Excellent. It looks like this DHCP client successfully received a lease from the server. I'll also take a look at the leases file on the DHCP server to see which addresses it has leased out:

```
# more /var/db/dhcpd.leases

# All times in this file are in UTC (GMT), not your local timezone.  This is
# not a bug, so please don't ask about it.  There is no portable way to
# store leases in the local timezone, so please don't request this as a
# feature.  If this is inconvenient or confusing to you, we sincerely
# apologize.  Seriously, though - don't ask.
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-V3.0.1rc11

lease 192.168.10.20 {
  starts 0 2003/04/20 21:49:28;
  ends 1 2003/04/21 21:49:28;
  binding state active;
  next binding state free;
```

Configuring a DHCP Server

Dru Lavigne

```
hardware ethernet 00:50:ba:de:36:33;
}
```

Changing the Logging File

The last configuration I would like to demonstrate today is changing the default logging file. First, I'll change the logging line in `/usr/local/etc/dhcpd.conf` so that it looks like this:

```
# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;
```

Next, I'll create an empty log file called `dhcpd.log`:

```
# touch /var/log/dhcpd.log
```

Then, I'll create an entry for this logfile in `/etc/syslog.conf` by adding this line:

```
local7.*                                /var/log/dhcpd.log
```

Let's take a look at that entry for a moment. By default, you're given eight logging "facilities" to use for local applications; these are called local0 to local7. You can use whichever local facility you wish, as long as it isn't being used by another application. I've decided to use local7, which is why I also referred to it by that name in the DHCP server configuration file.

Once you've chosen a facility, you follow it by a period and a logging level. I've chosen the logging level of *, which will log all events, regardless of their level. I then gave the location of the log file to which to write events.

Once I've saved the changes to `/etc/syslog.conf`, I need to send syslogd a signal one so it is aware of the changes:

```
# killall -1 syslogd
```

I also need to make the DHCP server aware of the change. Remember, a signal one won't do it, so I'll use the restart option to the startup script:

```
# /usr/local/etc/rc.d/isc-dhcpd.sh restart
```

Finally, I'll see if it worked:

```
# more /var/log/dhcpd.log
Apr 20 19:32:22 fubar dhcpd: Internet Software Consortium DHCP Server V3.0.1rc11
Apr 20 19:32:22 fubar dhcpd: Copyright 1995-2003 Internet Software Consortium.
Apr 20 19:32:22 fubar dhcpd: All rights reserved.
Apr 20 19:32:22 fubar dhcpd: For info, please
    visit http://http://www.isc.org/products/DHCP
Apr 20 19:32:22 fubar dhcpd: Wrote 1 leases to leases file.
<snip>
```