

DHCP on a Multi-Segment Network

Dru Lavigne

So far in this series about DHCP I have demonstrated how to configure DHCP clients and a DHCP server for a single segment network. In today's article I'd like to finish the series by explaining how to use DHCP in a multi-segment network.

While I happen to be concentrating on the ISC software on a FreeBSD system, DHCP is a standard protocol: regardless of your particular mix of operating systems and the software you use to provide DHCP, the logic behind configuring DHCP remains the same.

What needs to be considered when using DHCP in a network that contains more than one segment? I'll discuss the following:

1. the addressing and subnetting scheme
2. dealing with broadcasts over multiple segments
3. configuring any intervening firewalls or router access lists

IP Addressing

Before you can successfully configure a network for DHCP, you need to know the physical and logical layout of the network. If you are fortunate, this information has already been recorded, is kept up-to-date, and you can actually find the necessary documentation. If so, immediately track down the responsible administrator and buy her or him lunch.

If you're not so fortunate, grab a pen and notepad and start walking through the network. Make note of every hub or switch and how many devices are plugged into each. Work your way toward the server closet and record the number of routers or LAN router interfaces. Find the locations of any DNS servers, WINS servers, and any other servers that may require static addresses. When you're finished, sketch out your results.

Next, determine which IP addressing scheme, if any, is currently in use on the network and add it to your sketch. If you are responsible for creating the addressing scheme, you will most likely be using one of the private range addresses:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

Here is an example of a small office with four network segments:

```
network ID:      192.168.10.0
subnet mask:    255.255.255.224
```

```
front office:
subnet ID          192.168.10.32      available addresses:
broadcast ID      192.168.10.63      192.168.10.33 - 61
default gateway   192.168.10.62
6 workstations
```

```
server closet:
subnet ID          192.168.10.64      available addresses:
192.168.10.68 - 93
```

Revised May 5, 2003

Page 1 of 7

DHCP on a Multi-Segment Network

Dru Lavigne

```
broadcast ID      192.168.10.95
default gateway   192.168.10.94
DNS server        192.168.10.65
WINS server       192.168.10.66
file server       192.168.10.67
```

```
lab1:                                     available addresses:
subnet ID      192.168.10.96                 192.168.10.98 - 125
broadcast ID   192.168.10.127
default gateway 192.168.10.126
WINS server    192.168.10.97
25 workstations
```

```
lab2:                                     available addresses:
subnet ID      192.168.10.128                192.168.10.129 - 157
broadcast ID   192.168.10.159
default gateway 192.168.10.158
15 workstations
```

It's important to record the subnet ID and broadcast ID of each network segment, as those two addresses are unavailable for use as host IDs. Each segment will have a unique default gateway address which must be a valid host ID for that segment.

Now let's see how the sketch of a network translates into a DHCP server configuration file. Remember that you will need a subnet declaration for each network segment. For the example above, I would need four subnet declarations or something like this:

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

#global options
option domain-name "smallcompany.com";
option domain-name-servers 192.168.10.65;
option netbios-name-servers 192.168.10.66, 192.168.10.97;
option netbios-node-type 2;
default-lease-time 86400;
max-lease-time 86400;
authoritative;
ddns-update-style none;

#front office
subnet 192.168.10.32 netmask 255.255.255.224 {
    range 192.168.10.33 192.168.10.61;
    option routers 192.168.10.62;
}

#server closet
subnet 192.168.10.64 netmask 255.255.255.224 {
    range 192.168.10.68 192.168.10.93;
    option routers 192.168.10.94;
}
```

DHCP on a Multi-Segment Network

Dru Lavigne

```
#lab1
subnet 192.168.10.96 netmask 255.255.255.224 {
    range 192.168.10.98 192.168.10.125;
    option routers 192.168.10.126;
}

#lab2
subnet 192.168.10.128 netmask 255.255.255.224 {
    range 192.168.10.129 192.168.10.157;
    option routers 192.168.10.158;
}
```

See how straightforward the subnet declarations are once you know the layout of your network? You may have noticed that I've included two additional options in the global options section. The option `netbios-name-servers` refers to WINS, so it includes the IP addresses of the two WINS servers. It is followed by the option `netbios-node-type` which I have set to 2. There are four possible node types:

Value	Type	Description
1	b-node	uses broadcasts instead of a WINS server
2	p-node	only uses a WINS server
4	m-node	tries a broadcast first, then a WINS server
8	h-node	tries a WINS server first, then a broadcast

The node type tells a computer running a Microsoft OS how to deal with NetBIOS name resolution. This type of name resolution is required whenever a computer needs to access a resource on a Microsoft network. In networking land, broadcasts are considered to be a bad thing and are discouraged when there are alternative ways to get the job done. The alternative in a Microsoft network is to use a WINS server. Microsoft has more information about node types and netbios name resolution.

The two WINS server options don't have to be global options. For example, if only lab1 contains Microsoft operating systems, I could remove those two options from the global section and instead insert them into lab1's subnet declaration:

```
#lab1
subnet 192.168.10.96 netmask 255.255.255.224 {
    range 192.168.10.98 192.168.10.125;
    option routers 192.168.10.126;
    option netbios-name-servers 192.168.10.66, 192.168.10.97;
    option netbios-node-type 2;
}
```

Dealing with Broadcasts

You may be thinking that creating a DHCP server configuration file isn't all that hard. It isn't, but we're not finished yet. We still have to deal with broadcasts and ensure that DHCP clients will receive a lease that is suited to their network segment. Since DHCP uses broadcasts and a multi-segment network contains routers that will drop those broadcasts, you have a few choices on how to deal with dropped DHCP broadcasts. I'll discuss two possible options:

DHCP on a Multi-Segment Network

Dru Lavigne

- place a DHCP server on every segment
- ensure every segment has either a DHCP server or a DHCP relay (but not both)

Either method will allow DHCP to run smoothly. Which one you choose will be a matter of configuration preference for the software which you have available.

If you decide to use option one, add a DHCP server to each segment in your sketch. Assign each one a static IP and ensure those addresses aren't in your pools of available addresses. Install the DHCP server software on each PC and create its configuration file.

In my example network, should I use the same DHCP server configuration file on each DHCP server on each of the four network segments? If I do, the DHCP servers won't know which subnet they are responsible for. For example, I want the DHCP server on the front office segment to just use the subnet declaration for the front office. Remember from the last article that you could use empty subnet declarations? This is where they come into play. I should modify the configuration file for the front office DHCP server so it looks like this:

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

#global options
option domain-name "smallcompany.com";
option domain-name-servers 192.168.10.65;
option netbios-name-servers 192.168.10.66, 192.168.10.97;
option netbios-node-type 2;
default-lease-time 86400;
max-lease-time 86400;
authoritative;
ddns-update-style none;

#front office
subnet 192.168.10.32 netmask 255.255.255.224 {
    range 192.168.10.33 192.168.10.61;
    option routers 192.168.10.62;
}

#server closet
subnet 192.168.10.64 netmask 255.255.255.224 {
}

#lab1
subnet 192.168.10.96 netmask 255.255.255.224 {
}

#lab2
subnet 192.168.10.128 netmask 255.255.255.224 {
}
```

Now this DHCP server has lease information for the front office segment. It is aware that there are three other segments on the network, but it is not responsible for leasing out information to the DHCP clients on those segments.

DHCP on a Multi-Segment Network

Dru Lavigne

The other three DHCP servers would have similar configuration files. The DHCP server on the server closet segment would have lease information for that segment and empty subnet declarations for the remaining three. The same idea would apply to the DHCP server on the lab1 segment and the DHCP server on the lab2 segment.

Using a Relay Agent

For option two, decide which segments will use a DHCP server and which will use a relay agent, and label your sketch accordingly. The DHCP relay agents don't have to use a static address, but they should be reliable machines that will be up whenever a client needs to contact a DHCP server. A relay agent isn't of much use when it is powered down.

If you are using a FreeBSD system for your relay agent, you will still have to build the DHCP server port in order to build the relay. However, instead of editing the DHCP server configuration file, you will instead edit the dhcrelay configuration file. Building and installing the DHCP server created a sample file and an editable copy in `usr/local/etc`:

```
$ cd /usr/local/etc
$ ls | grep dhcrelay

rc.isc-dhcrelay.conf
rc.isc-dhcrelay.conf.sample

$ more rc.isc-dhcrelay.conf

dhcrelay_options=      # command option(s)
dhcrelay_ifaces=      # ethernet interface(s)
dhcrelay_servers=     # dhcpd server(s)
```

You'll note that the DHCP relay configuration file is very short and fairly straightforward once you understand how a DHCP relay agent operates. When a segment contains a relay agent instead of a DHCP server, the relay agent will intercept a client's DHCP broadcast and convert it to a unicast. This means that it readdresses the packet so it is destined for the IP address of a DHCP server. Since routers pass unicast packets, the DHCP server will receive the request and respond with a lease.

There is one caveat: somehow the DHCP server needs to know which network segment the original client broadcast came from so that it can offer a lease that is appropriate for the client. This information is provided by the `dhcrelay_options`.

You should add `-a` to the `dhcrelay_options`. This tells the relay agent to add an option to the DHCP request informing the DHCP server which interface on the relay agent the client request came from. This information is important: it allows the DHCP server to calculate which network segment the client resides on so it can offer it a suitable lease.

Let's see how this translates into my example network. I'll have one DHCP server in the server closet and three DHCP relay agents: one in the front office, one in lab1, and the third in lab2.

DHCP on a Multi-Segment Network

Dru Lavigne

Since there is only one DHCP server, it will be responsible for assigning leases to all four subnets. Accordingly, its configuration file will contain the full lease information and no empty subnet declarations.

Each relay agent will have a configuration file similar to this:

```
dhcrelay_options=-a           # command option(s)
dhcrelay_ifaces=ed0          # ethernet interface(s)
dhcrelay_servers=192.168.10.68 # dhcpd server(s)
```

When configuring your own dhcrelay_ifaces line, use the interface name for that FreeBSD system. When configuring your dhcrelay_servers line, use the DHCP server address for your network. Once you've made your changes, make the script executable:

```
$ chmod +x /usr/local/etc/rc.isc-dhrelay.conf
```

Then copy the sample startup script and see if the agent starts without any error messages:

```
% cp /usr/local/etc/rc.d/isc-dhcrelay.sh.sample
   /usr/local/etc/rc.d/isc-dhcrelay.sh
% /usr/local/etc/rc.d/isc-dhcrelay.sh start
```

```
Internet Software Consortium DHCP Relay Agent V3.0.1rc11
Copyright 1997-2000 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP
Listening on BPF/ed0/00:d0:09:ef:25:38
Sending on   BPF/ed0/00:d0:09:ef:25:38
Sending on   Socket/fallback
```

A sockstat should also show the relay agent is listening for DHCP requests:

```
$ sockstat | grep dhcrelay
root    dhcrelay  1664    4  udp4    *:67          **
root    dhcrelay  1664    3  dgram   syslogd[77]:3
```

The DHCP relay startup script is similar to the DHCP server script in that it supports four options: start, stop, restart, and status. If you change the relay agent's configuration file, use the restart script to stop and start the agent. If you ever need to stop the agent or check its status, use the appropriate option with the script.

Firewalls and DHCP

Once you have your DHCP server(s) and relay agent(s) configured, you'll want to ensure that the DHCP packets aren't being dropped by any intervening firewalls or routers. Depending upon the layout of your network, this may or may not be an issue. Some networks allow internal LAN traffic to flow freely and only inspect packets that are leaving for or entering from the Internet.

DHCP on a Multi-Segment Network

Dru Lavigne

However, if your network does filter internal packets, check the rules on the firewall or the router access list. The README file found in /usr/local/share/doc/isc-dhcp3 explains what you're looking for in your rulebase:

```
"If you are running the DHCP server or client on a computer that's also acting as a firewall, you must be sure to allow DHCP packets through the firewall. In particular, your firewall rules must allow packets from IP address 0.0.0.0 to IP address 255.255.255.255 from UDP port 68 to UDP port 67 through. They must also allow packets from your local firewall's IP address and UDP port 67 through to any address your DHCP server might serve on UDP port 68. Finally, packets from relay agents on port 67 to the DHCP server on port 67, and vice versa, must be permitted.
```

```
We have noticed that on some systems where we are using a packet filter, if you set up a firewall that blocks UDP port 67 and 68 entirely, packets sent through the packet filter will not be blocked. However, unicast packets will be blocked. This can result in strange behavior, particularly on DHCP clients, where the initial packet exchange is broadcast, but renewals are unicast - the client will appear to be unable to renew until it starts broadcasting its renewals, and then suddenly it'll work. The fix is to fix the firewall rules as described above."
```

Obviously, the syntax you use to achieve this will vary greatly depending upon the firewall or router in use.

I hope you have enjoyed the DHCP series and will have the opportunity to configure a network for DHCP. Until next time, happy networking.