Dynamic Host Configuration Protocol (DHCP)

COSC301 Laboratory Manual

> **Note**
>
> In this lab, we do not consider IPv6 at all, DHCP is related only to IPv4. We could have a look at DHCPv6, but that is a more advanced issue that is not currently well supported at present in default configurations.

The DHCP server that ships with most Linux distributions is the standard ISC (Internet Software Consortium) DHCPd software. You can get more information from www.isc.org [http://www.isc.org].

In order to set up DHCP on our server today, you will need to **apt-get install isc-dhcp-server**. After you install it, log messages will tell you that it failed to start. This is expected, as there is no useful default configuration that can be supplied, so the server fails to start because of the missing information. However, it has been installed correctly on Server1 after using the above **apt-get** command, so don't worry.

> **Note**
>
> In this lab, and most others, we have been loading all our services onto your Server1. However, in general there is no operational requirement for the DHCP service to be hosted on the same server as the DNS service, and on larger network it would be preferable to have them housed on different servers.

1. DNS Alterations

During this lab, we shall be offering a pool of addresses. These addresses should be listed in DNS. There may be other things you need to check too. You need to make the following maintenance to your DNS server:

1. There is no well-known-alias for DHCP services, so you don't need to worry about adding anything like `dhcp.domain` to our DNS.

2. Ensure you have forward and reverse DNS mappings for Client1, with the address 192.168.1.11. You do not need to concern yourself with IPv6 for this lab. Client1 will be given a static IPv4 address from the DHCP server.

3. Add in mappings for 32 hosts, starting from 192.168.1.32 (this range is exactly the same as the CIDR prefix 192.168.1.32/27). We will use these for our pool of dynamic addresses. You should employ some suitable editor wizardry, as outlined below. You may like to refer to the video about this task on the website.

   When you have a lot of workstations, you generally have some sort of numerical scheme, and so can easily write a small script to generate lots of A or PTR records. Here's an example which you would write into the DNS server configuration file. We normally store it in the file commented out, then copy, uncomment and execute to generate the output.

```
for (( i=32; i<32+32; i++ ))
do
  This version is suitable for a forward zone.
  printf "ip%03d\tA\t192.168.1.%d\n" $i $i
  This version is suitable for a reverse zone.
  # printf "%d\tPTR\tip%03d.localdomain.\n" $i $i
done
```

In **vim**, you can select this text (using the `V` command), then use `!bash` followed by **RETURN** to execute and replace the text using the **bash** interpreter. You will end up with the set of entries below.

```
ip032   A       192.168.1.32
ip033   A       192.168.1.33
ip034   A       192.168.1.34
…
```

Hit **u** to undo the changes, make a copy of the script text, and comment out one copy, so you can reuse it later. Execute the uncommented version so you get the output you want. You will also need to follow a similar procedure for the reverse zone also. Don't ignore the reverse zone, its correctness is relied on for many servers and services.

Note

If you want to do a similar thing in Emacs, use `C-u M-|` on a selection (region). Without the `C-u`, the output will be shown in a seperate buffer, rather than replacing the region in the current buffer.

BIND $GENERATE statement (for reference only, no action required)

Actually, generated data is a common requirement, so the BIND implementation has a shorthand for this in the master zone file. That would allow us to use one line in each zone file to replace each script that we used.

```
$GENERATE 32-63 ip${3,d}  A  192.168.1.$       Forward entries
$GENERATE 32-63 $  PTR  ip${3,d}.localdomain.  Reverse entries
```

We're not going to use it here, but if you want to know more, consult the BIND 9 Administrator Reference Manual [http://www.bind9.net/arm97.pdf] for more information.

4. Make the changes take effect by using **rndc reload**, and then check the logs to make sure that it loaded without complaint.

5. Test that you can resolve forwards and backwards the mappings you just modified or added. For the dynamic addresses, you need only test the border cases.

2. DHCP Server Configuration

You have already installed the DHCP server package earlier in this lab, so all that is left to do is to configure it.

Now we shall revise the configuration file `/etc/dhcp/dhcpd.conf`, a template of which is shown below. You will need to edit this as indicated.

```
ddns-update-style none;
option domain-name "localdomain";
option domain-name-servers ns1.localdomain;
default-lease-time 3600; 1 hour
max-lease-time 3600;

authoritative;          Send DHCPNAK when nodes request an incorrect address.

This is for the "outside" interface; we don't do anything with it.
subnet 10.0.2.0 netmask 255.255.255.0 {}

subnet 192.168.1.0 netmask 255.255.255.0 {
   option routers 192.168.1.1;
   range 192.168.1.start 192.168.1.end;     Change to suit pool addresses
}

This is an example of a static mapping.
host client1 {
   hardware ethernet 00:05:1c:05:38:ae;     Change
   fixed-address client1.localdomain;
}
```

ddns-update-style ...
When we issue a lease, we can update the data in the effected DNS zones by telling the DNS server the new values. This is called Dynamic DNS and is used mostly by Windows in a domain environment. We won't deal with it here, although it could be more useful perhaps with DHCPv6 for IPv6 addresses.

option domain-name ..., **option domain-name-servers ...**
These are the DNS-related configuration details we give to the DHCP client machines. They do not affect how Server1 interacts with the DNS system (although commonly the same DNS server will be used).

The DNS details are often common to a network, rather than a subnet, and so we have specified the DNS domain name and nameservers as globals. If needed, they can be overridden in the *subnet* stanza. Note that we have specified them by name, not address; the DHCP server resolves these names at the startup of the DHCP server.

Important

If you specify anything using DNS names, rather than an IP address, if the lookup fails, the DHCP server will *silently* ignore the entry! If this bothers you, specify IP addresses instead.

On the other hand, specifying names instead of addresses could be easier to maintain.

default-lease-time ..., **max-lease-time ...**
A lease time is how long a client can have an address before they have to renew the lease. A shorter lease time allows for updates to be made faster, at the cost of more traffic. Clients can ask for a particular lease time, or we give them a default; the requested lease time can be restricted to a particular maximum and minimum value. Suggested defaults vary widely; it depends on your network. You can change these settings for different subnets etc. Clients configured with a static address might have a longer lease time compared to those configured via a dynamic pool.

authoritative
The single most common misconfiguration is to *not* specify the "authoritative" statement. This means that your server contains the knowledge to tell clients when they are requesting something that is invalid. This is important in portable devices. Consider a laptop that requests a lease from a network at home, then later on requests to use the same address on the network at work. By instructing the DHCP server that it has the authoritative configuration for the network, it will respond with a DHCPNAK, which tells the client it must reject its current lease and request a new one.

subnet 10.0.2.0 netmask 255.255.255.0 {}
When the server starts, it will look at the available interfaces; if there is not a `subnet` clause for each subnet the host is connected to, it will refuse to start. So we have to tell the server to not do anything on our "outside" subnet.

Protecting ourselves from being a DHCP server on a network in which we are not meant to be is a very important (particularly on a multi-homed host, such as our server which is in-between two networks). We can also edit the command-line arguments of the DHCP server to specify which interfaces the DHCP server should use. To do this, edit `/etc/default/isc-dhcp-server`, changing the `INTERFACES` statement to be `inside`.

subnet 192.168.1.0 netmask 255.255.255.0 { ... }
This is the "inside" subnet, the "localdomain" zone, which has a network address of 192.168.1.0/255.255.255.0 (commonly expressed as 192.168.1.0/24).

It must correspond to the configuration details of one of the host's network interfaces.

option routers ...
You can specify various options that that take effect inside the subnet. Notice that the globals options also are in effect. There are *many* different options that you can support, although they all depend on client support. Have a look at dhcp-options(5) for further ideas.

range ...
Any requests for a DHCP lease that is not fulfilled by a matching `host` statement are given a dynamic address from this range. This is where client2 will get its address from. You will need to adjust this to make it agree with the 32 entries you added to DNS.

host client1 { ... }
This starts a scope where we provide a static mapping between MAC address and IP address (which can also be provided as a DNS name, which is then looked up before giving it to the client).

The `fixed-address` argument could be an IP address, but we prefer to use DNS names, because that way all the mappings between ethernet MAC addresses and DNS names is contained in the DHCP configuration, and all the mappings between DNS names and IP addresses are in the DNS. It just makes things easier to deal with, and less error prone. But you do need to take care that the address maps to a valid address when the server is started, because it fails silently. You would then see the client getting an address provided by the dynamic pool configured with the `range` statement.

The argument to the host statement, which in this example is `client1` is the Client Identifier. If you don't use a `hardware` statement, the client would have to provide a DHCP Client Identifier in order to be given the address specified by the `fixed-address` statement. Some ISPs, most notably some cable internet providers overseas, require their users to provide such a Client Identifier.

To find the MAC address of a machine, you can use the **ifconfig eth0** command on the client, or by **ping**ing the machine then checking your **arp** table. Most physical ethernet cards have it written on the card as well. Laptops often have it written on a sticker on the bottom of the unit.

The server program **dhcpd** will be started at boot-time, because you have installed the server and by doing so its startup script was activated at installation. If you want to change the options that the *startup script* uses to start the DHCP server, just edit `/etc/default/isc-dhcp-server`. For example, if you have multiple interfaces, it's useful to specify the interfaces on which it should listen in `/etc/default/isc-dhcp-server`.

Now restart the service (**sudo service isc-dhcp-server restart**) to affect the changes you have made.

Exercise

As an exercise check your logs (they will appear in `/var/log/syslog`) and make sure there is no error in the startup messages logged by the server.

3. Client Configuration

Note

In this section, you will need to make changes to *both* Client1 and Client2. Ensure both clients are connected to the Internal Network.

We shall assume that Client1 is like a workstation, and doesn't move around like a laptop might. Therefore, we shall *not* use Gnome Network Manager, and instead we shall manage Client1's network configuration in `/etc/network/interfaces`.

Change Client1's eth0 stanza in the `interfaces` file to the following, leaving any other interface (ie. the loopback interface) unmodified:

```
auto eth0
iface eth0 inet dhcp
Delete the address, netmask and gateway lines
```

Restart Client1. There are other things we could have done instead of restarting, but restarting can be faster in this case, and gets Network Manager out of the way. It should not be visible after it has started (because there are no interfaces available for it to manage).

There is no really special configuration for the clients, and generally no extra software to install, as a DHCP client is generally installed by default on most systems. Configuration can be as simple a single checkbox.

Exercise

As an exercise check that Client1 has an IP address of 192.168.1.11. Restart Client2, and check it has an address in the range 192.168.1.32-63.

4. Self-assessment

1. DNS must be well-configured.

2. The DHCP server process must start correctly.

3. Both clients must get their correct addresses: client1 gets bound to its MAC address, and the IP address 192.168.1.11; client2 gets a dynamic address.

4. What are the risks to a network when using DHCP for configuration of network details? What can be done about those risks while keeping the benefits of DHCP.

Hopefully, this will have been a short lab. You can use whatever time is left for catching up on previous labs. You should spend some time reading about layer-2 Ethernet security; you should find the *Cisco SAFE Layer 2 Security In-depth [http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/sfblu_wp.pdf]* whitepaper informative reading.