Networking Infrastructure – Overview Sheet

DHCP Server

DHCP is a method a networked computer can broadcast a request for an IP address over the local network and having a centralised system for handing how random or pre-configured IP addresses to machines in a given range. The server will respond with not only an IP address, but also gateway and dns server information.

Installing a DHCP service in ubuntu is easy.

```
sudo apt-get install dhcp3-server
```

To set up the service we need to edit /etc/dhcp3/dhcpd.conf to reflect the kind of network we want to have configured by dhcp. First we specify some universal options which apply to all computers that this service will respond to:

```
option routers 192.168.1.1;
option domain-name-servers 192.168.1.1;

default-lease-time 86400;
max-lease-time 604800;

authoritative;
```

These options set the default network route (to the internet), a list of domain name servers, some timers which tell the clients how often they should request a new ip-address and a boolean option which specifies that this server should be considered authoritative (the main dhcp server with all the configurations for the entire network) it basically means this server will always have the correct configuration.

Next we can set up a small C class network for any client requests:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
        option domain-name "network.local";
        ddns-rev-domainname "network.local";
        option domain-name-servers 192.168.1.1;
        ddns-domainname "network.local";
        range 192.168.1.20 192.168.1.50;
        option subnet-mask 255.255.255.0;
        option routers 192.168.1.1;
    }
```

This section sets a number of catch all options, some of which aren't required as they're specified globally as the same value but are noted here for completeness. The main sections are the parts where we specify the ip range and subnet mask.

Finally we're going to add a specific dhcp ip address for a machine with a specific MAC address:

```
host clientA.local {
        hardware ethernet 00:30:6c:76:be:8b;
        fixed-address 192.168.1.2;
    }
```

This part of the configuration can contain the same kinds of options as the catch all above, but is specific to machines with that specific interface. This does mean that plugging the computer into the network with a second ip ethernet interface will result in an automatic ip address as it will then have a different interface MAC address.

To finish up simply save the configuration and then restart the dhcp server:

```
sudo /etc/init.d/dhcp3-server restart
```

DNS Server

A DNS server is a simple service which resolves a name into an IP address. Each name is separated out into a zone (domain name section) and the configurations have also been separated out into zones to make it easy to manage and update each one separately.

Over time the DNS system got overloaded with other configurations and different re-purposing. This has made DNS configuration far more complex than it needs to be and it's best if most of these extraneous configurations not be used beyond simple forward and backwards resolution and MX mail records.

Installation of the ubuntu dns server (bind9) is simple as always:

```
sudo apt-get install bind9
```

The first thing we need to do is recognise that our bind server will only contain records for our local networks and not for all the servers on the internet. There for we need to configure bind to forward domain resolutions to an internet based domain name server so internet domains continue to work. We do this in the /etc/bind/named.conf.options file:

```
options {
        directory "/var/cache/bind";
        forwarders {
                12.6.1.12;
        };
        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
};
```

Your main configuration is in /etc/bind/named.conf, but you won't be editing this file because it points to a special configuration for local networks called /etc/bind/named.conf.local, instead we're going to edit this config. Both of these files configure various zones and tell bind where to find their zone databases. For instance in this example I will create a single domain zone for our network called “network.local” the second zone configuration is just the reverse lookup of the network.lookup zones:

```
zone "network.local" IN {
    type master;
    file "/etc/bind/zones/network.local.db";
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/rev.1.168.192.in-addr.arpa";
};
```

You will see that there is our network.local zone pointing to /etc/bind/zones/network.local.db, all your zone files should probably go into the /etc/bind/zones/ directory. We also specify that these zone records are master (similar in principle to the authoritative option in DHCP). And now we have to move onto creating each of those zone files we just configured. You'll notice that the IP address in the reverse lookup is backwards, this is because this is the most efficient way to lookup unindexed IP address records.

In our /etc/bind/zones/network.local.db file we want to first specify some global options, you'll notice that each of the domains ends with a single full stop. This is because all DNS records end this way, we just never see the final full stop as most systems fill this in for us when looking up domains.

```
$TTL 3600
network.local. IN SOA dns.network.local. mail.network.local. (
    2009080403 ; serial (unique id)
    14400      ; refresh in seconds (8h )
    900        ; retry in seconds (15m )
    1209600    ; expire in seconds (2w )
    3600       ; minimum expire in seconds (1h )
  )
```

You **must** specify the DNS and email servers for this SOA record and each of the settings shown. The serial id should change every time there is a new version of the zone file, that way the server will know to distribute it. So for simplicity it's simply set to the date you modified it, plus a number indicating how many times in that day you modified it. But it can really be anything.

We also add a name server record (NS) to specify where the authoritative records for this domain are held, for the local computer it's likely to be the same dns server as mentioned in the SOA record above. We also specify the mail servers for this local domain (not required).

```
setc.local. IN NS reflection.setc.local.
setc.local. IN MX 10 mail.setc.local.
```

Next add all the records which link the local network IP addresses to a name on the network. Each of these records are simple A records which link a name under 'network.local.' to a local IP address, you'll see we also specify the mail and dns domains which are used above in the SOA record:

```
router    IN A 192.168.1.1
mail      IN A 192.168.1.2
dns       IN A 192.168.1.5
desktopa  IN A 192.168.1.21
desktopb  IN A 192.168.1.22
desktopc  IN A 192.168.1.23
```

Save this and now onto the zone file for the reverse lookup which is "/etc/bind/zones/rev.1.168.192.in-addr.arpa", again we start with a header configuration and follow with records:

```
$TTL 3600
@ IN SOA dns.network.local. mail.network.local. (
    2009080403 ; serial
    14400      ; refresh = 8h
    900        ; retry = 15m
    1209600    ; expire = 2w
    3600       ; minimum = 1h
  )
```

And follow with records, don't forget the training full stop:

```
; define the authoritative name server
          IN NS dns.network.local.
; our hosts, in numeric order
1         IN PTR router.network.local.
2         IN PTR mail.network.local.
5         IN PTR dns.network.local.
21        IN PTR desktopa.network.local.
22        IN PTR desktopb.network.local.
23        IN PTR desktopc.network.local.
```

Save this zone file and then restart the DNS bind service using init.d:

```
sudo /etc/init.d/bind9 restart
```

You can test it works by attempting to do ping and lookup requests from the command line to some of your configured domains. So long as your DHCP server is configured to point as your DNS server for resolution.