

KanREN DHCP Seminar November 7-9, 2005

Configuring Failover

Partner with another student to build a failover pair. First, decide on a name for your server cluster. It can be anything—I'll call mine "classroom-1-6". Also decide which server will be the primary and which will be the secondary. The duties of the two servers are the same; they just need to be identified separately to perform their startup duties correctly.

In order to provide coverage for each other, both servers need to see DISCOVER messages (broadcasts) from all subnets that they serve. This means that additional relays need to be set up to pass traffic from each yellow and green subnet to the partner server. Once you've paired up, make sure the presenter knows your team selections so the relaying can be set up in the classroom router.

Each server only needs to have a failover configuration for its own role. However, I prefer to put *both* servers' failover configurations onto *each*. Since we automate the synchronization of the configuration files between our servers, I build a *dhcpd.conf-failover-primary* and *dhcpd.conf-failover-secondary* on each server, and then symlink the appropriate file to *dhcpd.conf-failover*, which is the file I include from the master configuration file.

So on each server, create the file *dhcpd.conf-failover-primary* defining your primary server (and changing the server and partner IP addresses to be appropriate for the primary and secondary servers in your partnership):

```
failover peer "classroom-1-6" {
    primary;

    # my server settings
    address 192.168.10.253;
    port 519;

    # my partner's server settings
    peer address 192.168.60.253;
    peer port 520;

    # fraction of addresses for load-sharing (0-255)
    split 128;

    # how long leases to grant when partner has failed
    # should normally be 3600 (1 hour), but shorter for demonstration
    mclt 300;
}
```

Next, create the file *dhcpd.conf-failover-secondary* defining your secondary server:

```
failover peer "classroom-1-6" {
    secondary;

    # my server settings
    address 192.168.60.253;
    port 520;

    # my partner's server settings
    peer address 192.168.10.253;
    peer port 519;

    # split and mclt inherited from primary server
}
```

Then on whichever server is the primary, symlink (alias) the primary failover file as *dhcpd.conf-failover*; and on whichever server is the secondary, symlink the secondary failover file as *dhcpd.conf-failover*:

```
lab1:/var/dhcp # ln -s dhcpd.conf-failover-primary dhcpd.conf-failover
```

OR

```
lab1:/var/dhcp # ln -s dhcpd.conf-failover-secondary dhcpd.conf-failover
```

Finally, include the failover symlink (alias) into your master configuration file:

```
# Include failover configuration
include "dhcpd.conf-failover";
```

Then test your configuration, and restart both servers to bring them up in failover mode. Watch the log files as the servers go through different phases of bringing up the failover protocol.

Next, define pools within each subnet to contain the dynamic address ranges. Change each subnet to look something like this:

```
subnet 192.168.10.0 netmask 255.255.255.0 {
    option routers 192.168.10.254;

    pool {
        deny dynamic bootp clients;
        failover peer "classroom-1-6";

        range 192.168.10.32 192.168.10.63;
    }
}
```

After the servers have reached NORMAL state, make sure that each of your PCs can get an IP address on all four of your subnets.

Then shut down either server (`kill `cat dhcpcd.pid``) and watch the failover messages in the remaining server's log file. Test that you can still get an IP address on each of your four subnets. Once you're sure your failover is working correctly, restart the halted server, watching the log messages on both servers as they resynchronize.

If you have time, shut down the other server and test again.

If you *still* have time available, ask the presenter for a failover debugging exercise.