

# Failover with ISC DHCP

Paul Heinlein

## Introduction

Small- and medium-sized networks often have a single DHCP server, which can become a single point of failure for a large number of hosts on the network. When the DHCP server goes off-line, DHCP client hosts lose their addresses and ability to communicate with the rest of the network. Since most desktop computers, and even some servers, get their networking configuration via DHCP, such an outage can result in a lot of downtime.

If the network has a Unix infrastructure, there's a good chance that it's using the Internet Systems Consortium (ISC) DHCP server, which is widely available on Linux and BSD systems.

Starting with version 3.0, the ISC DHCP server offered failover capabilities that allow network administrators to offer a more robust DHCP service. A failover setup requires a little care, but it's fairly straightforward to implement.

## A Simple Starting Point

Before getting to the failover setup, let's establish a simple baseline DHCP configuration with no frills.

```
#
# /etc/dhcpd.conf for simple network
#

authoritative;
ddns-update-style none;

subnet 192.168.200.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.200.255;
    option routers 192.168.200.1;
    option domain-name-servers 192.168.200.1;
    pool {
        max-lease-time 1800; # 30 minutes
        range 192.168.200.100 192.168.200.254;
    }
}
```

With this configuration, our server will act as the authoritative DHCP server on the 192.168.200.0 subnet, handing out addresses from 192.168.200.100 to 192.168.200.254 to any host that asks for one.

## The Problem

Our configuration will work fine until the DHCP server goes off-line. The cause of its demise might be a hardware failure, a power outage, or even an OS upgrade; it doesn't matter. Once it's gone, all DHCP client hosts will lose their network configurations within 30 minutes (our maximum lease time).

We could just bring another DHCP server online in its place, but the information about leases will be lost, possibly forcing clients to acquire new addresses. In that situation, clients would have to break any existing network connections. In some cases, local X sessions would also break. (If you're bored sometime, try changing the hostname of your machine when running a live X desktop. The recovery process can be amusing.)

## Failover with ISC DHCP

Paul Heinlein

Alternatively, we could plan for a downtime by increasing lease times from 30 minutes to the better part of a day. That would reduce—but not completely remove—the risk of any given client having its lease expire while the server is off-line, but any newly arriving client won't get an address.

### Configuring the Primary

Once you identify the machine that will act as the secondary DHCP server (or the new primary, if you're going to demote the old server), you'll want to make sure the clocks on the two machines are in sync. Timestamps are very important to dhcpd! After that, it's time to configure it. Using the simple configuration above, we'll add the bits necessary to upgrade it to serve as the primary in a failover situation:

- The failover peer section that identifies the primary and secondary servers; in the example below, it's called "dhcp-failover," but it can be any string that works for you. The example identifies the two DHCP servers by address, but you can use DNS names as well. In the past couple years, TCP ports 647 (primary) and 847 (peer) have emerged as the standard bindings for DHCP failover. It's worth noting that as recently as 2005, the dhcpd.conf(5) man page used ports 519 and 520 in its failover example, but 647 and 847 look like good choices as of 2008.
- The dhcpd.conf(5) man page says that the primary port and the peer port may be the same number. That's the configuration I deploy, using the port 647 for both the primary and the peer.
- The pool sections for which the failover pair is active; in our example, we have only one pool section, so we add a reference to our failover peer set.

```
#
# /etc/dhcpd.conf for primary DHCP server
#

authoritative;
ddns-update-style none;

failover peer "dhcp-failover" {
    primary; # declare this to be the primary server
    address 192.168.200.2;
    port 647;
    peer address 192.168.200.3;
    peer port 647;
    max-response-delay 30;
    max-unacked-updates 10;
    load balance max seconds 3;
    mclt 1800;
    split 128;
}

subnet 192.168.200.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.200.255;
    option routers 192.168.200.1;
    option domain-name-servers 192.168.200.1;
```

# Failover with ISC DHCP

Paul Heinlein

```
pool {
    failover peer "dhcp-failover";
    max-lease-time 1800; # 30 minutes
    range 192.168.200.100 192.168.200.254;
}
}
```

## Configuring the Secondary

For our simple network, the configuration for the secondary is quite similar to that of the primary. The only significant differences are in the failover peer definition: there's a secondary declaration, the mclt and split declarations are omitted, and the local and peer addresses are switched.

```
#
# /etc/dhcpd.conf for secondary DHCP server
#

authoritative;
ddns-update-style none;

failover peer "dhcp-failover" {
    secondary; # declare this to be the secondary server
    address 192.168.200.3;
    port 647;
    peer address 192.168.200.2;
    peer port 647;
    max-response-delay 30;
    max-unacked-updates 10;
    load balance max seconds 3;
}

subnet 192.168.200.0 netmask 255.255.255.0 {
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.200.255;
    option routers 192.168.200.1;
    option domain-name-servers 192.168.200.1;
    pool {
        failover peer "dhcp-failover";
        max-lease-time 1800; # 30 minutes
        range 192.168.200.100 192.168.200.254;
    }
}
```

The folks at ISC note that the DHCP failover protocol is still under development, which makes it sort of a moving target. As a result, they strongly suggest that the primary and secondary servers both be running the same version of dhcpd.

## SELinux Notes

As noted, running dhcpd in failover mode involves opening a TCP port for communication with the peer server. The SELinux policy distributed with CentOS 4 and 5 allows dhcpd to send packets over ports 647 and 847, but you'll need to tweak the policy if you want to use different ports.

## Failover with ISC DHCP

Paul Heinlein

The instructions below apply specifically to CentOS 4 (and, by extension, to Red Hat Enterprise Linux 4), though I suspect that they would also work on Fedora Core 3 and 4.

1. Install the selinux-policy-targeted-sources rpm, if it's not already on your system.
2. Open a command prompt in the /etc/selinux/targeted/src/policy directory.
3. Create or edit domains/misc/local.te using your editor of choice. Add a single line:

```
allow dhcpd_t port_t:tcp_socket name_bind;
```

4. Run **make reload** to install your modified policy.

### What the Logs Will Show

Once both servers are configured and working, the system logs will show when one goes offline. Here's what shows up when the primary goes down:

```
Nov 6 19:50:51 secondary dhcpd: failover peer dhcp-failover: I move  
from normal to communications-interrupted
```

When the primary comes back, the log will say (among other things)

```
Nov 6 19:51:37 secondary dhcpd: failover peer dhcp-failover: I move  
from communications-interrupted to normal
```

The other main difference in the logs will be the presence of pool reports. In failover mode, dhcpd will try to ensure that the primary and secondary servers each have a similar number of free dynamic leases for each pool declared in the configuration file. As the servers work to keep that balance, they'll occasionally log their status.

```
Nov 6 20:27:09 secondary dhcpd: pool 98e82b8 192.168.200.0/24  
total 155 free 38 backup 37 lts 0
```

In this case, 75 of the 155 of the addresses we declared eligible for dynamic assignment are still available. The primary holds 38 in reserve, the secondary 37. As long as the values for free and backup differ by no more than one, things are good. Should they vary by two or more (with a resulting non-zero lts), the pool addresses will be juggled until balance is restored.

Now, the single point of failure is gone. So go hog wild: install those security patches on your DHCP server that you'd put off because you didn't want to lose leases!