

Dynamic Routing with Quagga

1. Install 'net/quagga' from ports:

```
# cd /usr/ports/net/quagga
# make
# make install
```

2. Copy the sample configuration files into place

```
# cd /usr/local/etc/quagga
# cp /usr/local/share/examples/quagga/zebra.conf.sample zebra.conf
# cp /usr/local/share/examples/quagga/ospfd.conf.sample ospfd.conf
```

3. Have a look at these files; they are very simple. "zebra" is the overall manager of the IP forwarding table; "ospfd" is the daemon which speaks the OSPF protocol.
4. On the router machine, edit /usr/local/etc/quagga/ospfd.conf and add an OSPF section listing the network(s) on which you wish to exchange OSPF information

```
router ospf
  redistribute connected
  network 202.144.139.192/26 area 0
```

5. Configure your interfaces as normal

```
# ifconfig eth0 x.x.x.x netmask y.y.y.y
# ifconfig eth1 x.x.x.x netmask y.y.y.y
```

6. Edit /etc/rc.conf to enable the daemons we want:

```
quagga_enable="YES"
quagga_daemons="zebra ospfd"
```

7. Start the selected daemons:

```
# /etc/rc.d/quagga start
```

8. Check your forwarding table (netstat -rn)
9. You can manage the router using telnet: the interface is just like a Cisco router! This lets you make configuration changes in real time.

```
# telnet 127.0.0.1 2601      -- to manage zebra (password "zebra")
Try: show interface
    enable                  -- to get superuser mode (password "zebra")
    show run
    show ip route
    exit
```

```
# telnet 127.0.0.1 2604      -- to manage ospfd
Try: show ip ospf neighbours
    show ip ospf route
    exit
```

Hint: use [TAB] for command completion, and '?' to get a list of options

10. On the client machine, you can just point default route at the router. Or if you wish to play with zebra, then install it as above. You will need a 'network' statement on both machines to exchange information.

If you were running this in production, remember to change the default passwords!