

An ADSL Router Using FreeBSD

Brad Marshall <brad.marshall@member.sage-au.org.au>

January 15, 2004

1 Introduction

This article will cover setting up FreeBSD as an router for a internal network that has an internet connection via ADSL. This example is for a connection to Internode with a Billion 711CE ADSL router in bridging mode.

It will also cover basic firewalling to protect the internal network, as well as how to forward ports to an internal machine so services can be made available over the Internet.

2 ADSL Modem

To configure a Billion 711CE in bridging mode is fairly simple. First log into the web interface of the modem, then go to Configuration — WAN. Tick the box to enable bridging, and change the encapsulation to 1483 Bridging IP LLC. Remove all PPP configuration details, then submit and save the settings. When the modem reboots, it should be in a state ready for use by our FreeBSD box.

The next step is to connect the modem to the FreeBSD router. Find a crossover cable and connect this between the two devices. At this point we are now ready to configure the FreeBSD box to be our router.

3 Installing and Upgrading FreeBSD

After doing a normal install of FreeBSD, it is important to upgrade it to get all security fixes and the latest versions of the ports. This article assumes you have installed 5.2-RELEASE and want to track this release for security fixes.

First it is important to have installed the ports and source - if you didn't do so during install, run `/stand/sysinstall` and go to Configure — Distributions and select Ports and Src, then install. After doing this install `cvsup` from the ports system, by running the following command:

```
# cd /usr/ports
# pkg_add -r cvsup
```

Now edit `/usr/share/examples/cvsup/standard-supfile` to set the default release to 5.2-RELEASE.

```
*default release=cvs tag=RELENG_5_2
```

Next upgrade the packages and source by the following:

```
# cvsup -h cvsup.au.freebsd.org -g -L2 \  
    /usr/share/examples/cvsup/standard-supfile  
# cvsup -h cvsup.au.freebsd.org -g -L2 \  
    /usr/share/examples/cvsup/ports-supfile
```

After this, go to `/sys/i386/conf` and copy the GENERIC kernel config to one based on your hostname - ie, if your hostname is eagle, copy it to EAGLE. Then change directory to `/usr/src` and run the following:

```
# make buildkernel KERNCONF=KERNELNAME  
# make buildworld  
# make installkernel
```

At this point you reboot to the new kernel, then finish installing the rest of the newly compiled system, as follows:

```
# make installworld  
# mergemaster
```

This will merge any configuration you have made in the old system with the new one.

4 Configuring PPPoE

Setting up the ethernet card so it can be used by PPPoE is done by adding the following to `/etc/rc.conf`.

```
# Add any other interfaces you have here  
network_interfaces="r10 lo0"  
# Don't have to have an IP for r10, just enable it  
ifconfig_r10="media 100baseT/UTP up"
```

Actually configuring PPPoE to connect to the internet is fairly easy. Add the following to `/etc/ppp/ppp.conf`:

```
default:  
set device PPPoE:r10:provider  
set speed sync  
set mru 1492  
set mtu 1492  
set ctsrts off
```

```

enable lqr
add default HISADDR
set timeout 0 set redial 0 0

# Network Address Translation (NAT)
nat enable yes
nat log yes
nat same_ports yes
nat unregistered_only yes
enable dns

provider:
set authname username@isp.example.com
set authkey password

```

This configuration assumes you are using device rl0 as the one connected to the ADSL modem. Simply set the username and password to the one your ISP provided you with.

To set the connection to dial on boot and redial if it dies add the following to /etc/rc.conf:

```

ppp_enable="YES"
ppp_profile="provider"
ppp_mode="ddial"

```

To test that it works run ppp and tell it to dial manually, as follows:

```

# ppp
Working in interactive mode
Using interface: tun0
ppp ON hostname> dial provider
Ppp ON hostname>
PPp ON hostname>
PPP ON hostname>

```

As each p changes to P it indicates progress of the negotiation - the first is LCP completing, the next authentication, the final one is an IP being assigned. From this point you should have a fully working internet connection which you can test by pinging a close IP.

5 Firewall

Setting up a firewall to protect your internal network is important and simply done with FreeBSD. First you need to load the ipfw kernel module, and ensure that it is loaded on boot. To load it manually, just do:

```
# kldload ipfw
```

This means you can start setting up your firewall without having to reboot the system. To ensure it is loaded the next time you reboot, put the following in `/boot/loader.conf`:

```
ipfw_load=yes
```

Once this is done, configure `/etc/rc.conf` to tell it what firewall script to load. This example will cover using `rc.firewall` with the simple firewall type.

```
firewall_enable="YES"  
firewall_script="/etc/rc.firewall"  
firewall_type="simple"
```

Next edit `/etc/rc.firewall` and find the section for the simple firewall type and edit the definitions on the top for the outside and inside network. For the outside network, set the interface to `tun0`. Then review the script to check if there is anything else you wish to change - it defaults to not allowing anything to initiate connections to the router.

To forward ports to an internal host and make it accessible to the outside world, add the following to `/etc/ppp/ppp.conf` in the default section, near the rest of the nat settings.

```
nat port tcp 192.168.1.1:1234 2345
```

This example makes port 1234 on IP 192.168.1.1 available on external port 2345.

6 Conclusion

This article has covered updating and configuring FreeBSD to work as a router for an ADSL connection in a basic manner, and should have given you a fully working internet connection. It also shows how to protect the network from unwanted connections, and how to make ports from the internal network available to the outside world.