How to setup a multi-homed linux box

Grab the latest version of Slackware from www.linuxiso.org and install it on a machine with two network cards. Make sure to do a FULL install so that you are not missing any software. Its not terribly large and it's a real pain when you have to stop and install software that you need in order to complete this process. So for sanities sake, just do a full install. Also with the full installation you don't have to worry about attending the installation!

During the installation it will have you configure your network, make sure that you do this, because you will be downloading software directly from the net and installing it. I would also make sure you have an ftp available on another box that you can pull down software from. This is the way I do things, if you have a better way, good for you. Make sure that you have a DNS server setup and that you CAN ping to the Internet via hostname(Ctrl+X kills a ping command). If you are good here, just head back to your comfortable office computer and use putty to secure shell into the newly working linux server.

Kernel configuration

This has to be done for every version up to 9.1, which is the current one. Just follow the commands, if you need an explanation, read the man pages. "man topic" at the command line.

```
# cp /boot/config /usr/src/linux/config2
# su –
# mrproper
# make menuconfig
```

Ok, in menuconfig, arrow down to the bottom where it says "load configuration from file", select that then type the path to config2 (/usr/src/linux/config2). Once that configuration is loaded in, just arrow back up to the networking options section, enter it and enable "Advanced Router" then esc out till you are asked to save the configuration. Just click "yes" to save and you are done with menuconfig, unless you want to enable other things. Back to the command line.

```
# make dep
# make clean
# make modules
# make modules_install (this will take the better part of an hour because the original configuration has SOO much in it)
# mv /boot/vmlinuz /boot/vmlinuz.old
# cat arch/i386/boot/zImage > /boot/vmlinuz
# mv /boot/System.map /boot/System.map.old
# cp System.map /boot/System.map
```

Ok, the kernel compilation is complete and the files you needed to change out are changed out. Now we just need to wrap this up.

pico /etc/lilo.conf

Just add this line below the one that looks just like it, I suggest copying the line above and changing vmlinuz to vmlinuz.old and label to LinuxOld (see below).

```
image = /boot/vmlinuz.old
        root = /dev/hda1 (this may be different on your system)
        label = LinuxOld
        read-only
```

**Make sure that your timeout in the lilo.conf is set higher than 0 or you wont be able to select your previous kernel and fix your system if something goes wrong.
**The timeout is the time in seconds your machine waits before starting up the default kernel, so if you don't want to wait 1200 seconds, just change it to like 10.

Just save it and do this
lilo
It should say added, now you just reboot. If all is well and it should be, your machine should reboot with the ability to source route.
#Reboot
or
#Shutdown –r now

Multi-homing

Find and download IPRoute2. Make sure to download the one that is compiled for Slackware. If I am not mistaken the slackware ftp has it under the extras section.

installpkg /yourdownloaddirectory/iproute2.tgz

-It should be installed. As you probably noticed the name of the file I used is different from what I downloaded. Just do: mv iproute-version.blah.blah.tgz iproute2.tgz to get the file name I have. MV command actually works as rename too.

-Now that it's installed, lets make sure that we are good to go!

ip rule list (this should return a few lines of information, don't care what it says as long as it doesn't say that RNETLINK dumped, if it says that, then you didn't follow my kernel configuration correctly.)
ip route (should return something like you see when you do the route command.)

-Ok, if all is well, move on to the script (easy part cause I am giving that to you).
-Here is the script, modify it based on the IP address you want to use(next page):

Make sure to change the IP address, default gateways and networks to match your network setup.
----Begin copy after this line----

```
#!/bin/sh

IF1="eth0"                      #first network card
IF2="eth1"                      #second network card
IP1="192.168.1.2"               #ip address of first network card
IP2="192.168.10.2"              #ip address of second network card
P1="192.168.1.1"                #ip address of default router for the first NIC
P2="192.168.10.1"               #ip address of the default router for the second NIC
P1_NET="192.168.1.0/24"         #network ID of the first network card
P2_NET="192.168.10.0/24"        #network ID of the second network card

echo "1       T1" >> /etc/iproute2/rt_tables
echo "2             T2" >> /etc/iproute2/rt_tables

echo "1" > /proc/sys/net/ipv4/route/gc_elasticity
echo "1" > /proc/sys/net/ipv4/route/gc_interval
echo "1" > /proc/sys/net/ipv4/route/gc_timeout

ip route flush cache
ip route add $P1_NET dev $IF1 src $IP1 table T1
ip route add default via $P1 table T1
ip route add $P2_NET dev $IF2 src $IP2 table T2
ip route add default via $P2 table T2
ip route add $P1_NET dev $IF1 src $IP1
ip route add $P2_NET dev $IF2 src $IP2
ip route add default via $P1
ip rule add from $IP1 table T1
ip rule add from $IP2 table T2
ip route add $P1_NET dev $IF1 src $IP1
ip route add $P2_NET dev $IF2 src $IP2
ip route add default via $P1
ip rule add from $IP1 table T1
ip rule add from $IP2 table T2
ip route del default
ip route add default scope global nexthop via $P1 dev $IF1 weight 1 \
   nexthop via $P2 dev $IF2 weight 1
ip route flush cache
```

----End Cut before this line----
-Ok, just cut in paste this into a file called routes and do a “chmod 755 routes” at the command line to make it executable.

-Run it like this "./routes" You may get some errors, but you can ignore them if you are able to test and connect to both interfaces from the outside world. I usually test by opening port 22 on the firewall and doing and SSH into the box. A great ssh program is called "putty", just do a search on the net to find it if its not there.

-If the script works just move it to the root directory like so "#mv routes /routes" and then go to the rc.d directory "# cd /etc/rc.d" then do "# pico rc.local" add "/routes" at the end of the file, then save it.

-Now that we are multi-homed we can move on to email, which was the whole reason to perform multi-homing.

Useful programs:
Webmin – Used as web admin console for your linux server, not hard to setup and easy to run!
Putty – Very useful ssh/telnet tool for windows.

Websites:
www.webmin.com
www.apache.org