Configuring a Linux Gateway

This is a typical situation here at our Centre: we run a workshop where a few computers (laptops and towers) need to connect to the internet, or to another computer acting as a webserver, through a single mobile access point with a given wireless interface and IP address, but none of the computers have a wireless interface – only wired ethernet.

If we have an old laptop with a wired port on one side and a mobile port on the other, we can turn it into a gateway/router and connect everyone. In order to show what is going on behind the scenes, this talk looks in detail at the command-line configuration that is necessary.

The Hardware Arrangement

Figure 1 shows what is needed. Our computers on the left connect *via* a hub to the wired ethernet interface of the gateway computer. The gateway connects by wireless to the access point, which might be the final destination, or which might go off to the internet in general.

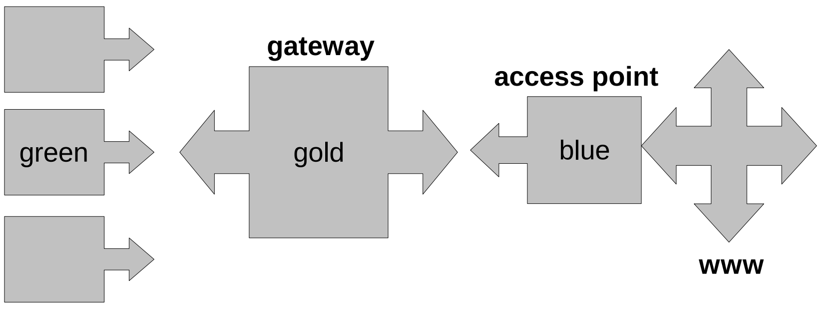

Figure 1: What we have to Work With

Installation of Necessary Software to run NICs

For this demonstration we have three computers (refer to Figure 1).

green on the left runs Debian 4.0 kernel 2.6.18 with only a wired ethernet interface `eth0`.

gold in the middle runs Debian 5.0 kernel 2.6.26 and has a wired ethernet interface `eth0` on one side and a mobile ethernet interface `ath0` on the other.

blue on the right represents our Access Point, and it runs Debian 5.0 kernel 2.6.26 and has a mobile ethernet interface `ath0` only. It also has a webserver to make it look like the `www`; this was installed by executing the command: `$ sudo apt-get install apache2`

Setting up an Ad-Hoc Network

I set up three computers – call them `green`, `blue` and `gold` - that intend to form an ad-hoc network called 'lsgnet'. (Refer to our previous talk on configuring this system, at:

`http://linuxlsga.net/adhoc.pdf`

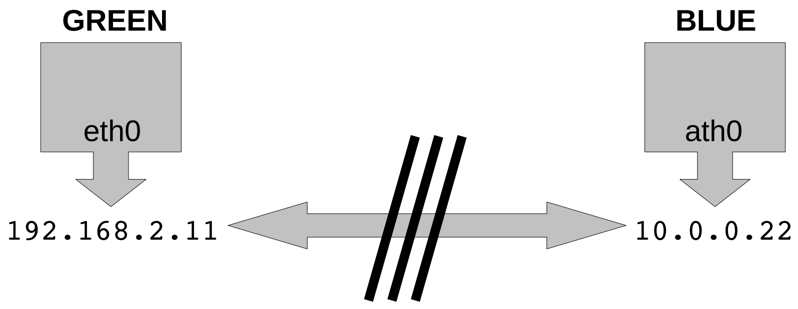

Figure 2: Direct Communication

We want to communicate direct from `green` to `blue` as depicted in Figure 2, but cannot do so, because `green` only has a wired ethernet interface and `blue` only has a mobile (unwired) interface. However we have access to a simple computer that has one wired and one mobile interface, that can act as a router, as shown in Figure 3. Thus we can get from the wired interface on `green` to the wired interface on `gold` (`eth0`), and then, using `gold` as a gateway, get from the mobile interface on `gold` (`ath0`) to the mobile interface on `blue`. If `gold` is configured properly as a gateway router, this configuration is the same as going between them directly as if `gold` did not exist.

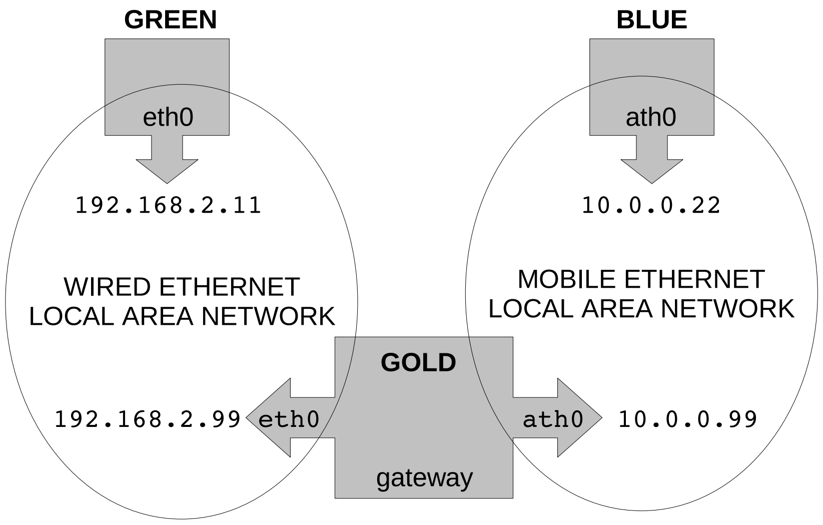

Figure 3: Gateway Configuration

Scripts

Connect the three computers as in Figure 3. Choose or discover your interface names and IP addresses as appropriate, using the command `/sbin/ifconfig -a`.

First: get **gold** and **blue** talking wireless on an ad-hoc basis (*q.v.* LSGA ad-hoc network talk). It is necessary to run all these commands as root; execute `sudo su -` and you should see the root prompt `#`. When this is done, run this to set the IP address of the wireless system on `blue`:

```
# ifconfig ath0 10.0.0.22
```

Second: establish these netfilter rules, in order, *via* the `/sbin/iptables` program on `gold`, as root (note the `#` prompt) following Figure 3 carefully:

1. Clear the filter, nat, mangle and user-defined tables.

   ```
   # iptables -t filter -F
   # iptables -t nat -F
   # iptables -t mangle -F
   # iptables -t filter -X
   ```

2. Anything input on the loopback interface is accepted.

   ```
   # iptables -t filter -A INPUT -i lo -j ACCEPT
   ```

3. Anything input as a new connection originating from anywhere but the `ath0` interface. is accepted.

   ```
   # iptables -t filter -A INPUT -m state --state NEW -i ! ath0 -j ACCEPT
   ```

4. Anything input that is related to traffic that has already been seen or that is traffic for an established connection is accepted.

   ```
   # iptables -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
   ```

5. Only forward packets coming in from `ath0` and going out to `eth0` if related to existing connections that went out through `ath0` some time before.

   ```
   # iptables -t filter -A FORWARD -i ath0 -o eth0 -m state \
                       --state ESTABLISHED,RELATED -j ACCEPT
   ```

6. Forward all packets without examination coming in from `eth0` and going out to `ath0`.

   ```
   # iptables -t filter -A FORWARD -i eth0 -o ath0 -j ACCEPT
   ```

7. Re-write (forge) the source (return) IP address in packets going out to `ath0` so that the reply can get routed back to the `ath0` interface.

   ```
   # iptables -t nat -A POSTROUTING -o ath0 -j SNAT --to 10.0.0.99
   ```

8. Remember to enable packet forwarding in the kernel.

   ```
   # echo "1" > /proc/sys/net/ipv4/ip_forward
   ```

Third: following Figure 3, establish the IP addresses and the default gateway for the interfaces on `gold` thus:

```
# ifconfig eth0 192.168.2.99
# ifconfig ath0 10.0.0.99
# route add default gw 10.0.0.22
```

Finally: go to `green` and run this to set the IP address of the wired system:

```
# ifconfig eth0 192.168.2.11
# route add default gw 192.168.2.99
```

The gateway should now be operational.

Testing the Gateway

You may now go to green and try these TCP/IP commands.
Ping blue like this:

```
$ ping -c 3 10.0.0.22
PING 10.0.0.22 (10.0.0.22) 56(84) bytes of data.
64 bytes from 10.0.0.22: icmp_seq=1 ttl=63 time=1.16 ms
64 bytes from 10.0.0.22: icmp_seq=2 ttl=63 time=2.26 ms
64 bytes from 10.0.0.22: icmp_seq=3 ttl=63 time=3.81 ms
--- 10.0.0.22 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 1.162/2.415/3.818/1.089 ms
```

Trace the route from green to blue like this:

```
$ traceroute -n 10.0.0.22
traceroute to 10.0.0.22 (10.0.0.22), 30 hops max, 40 byte packets
 1  192.168.2.99  1.431 ms  0.156 ms  1.287 ms
 2  10.0.0.22  1.691 ms  2.124 ms  1.075 ms
```

View web pages (stored on blue during the apache2 install) like this:

```
$ iceweasel http://10.0.0.22/index.html
It works!
```

If user has an account on blue with a known password, (this will not generally be the case with a real Access Point, only with our demonstration) log in to blue like this:

```
user@green:~$ ssh user@10.0.0.22
user@10.0.0.22's password:
Linux blue 2.6.26-1-686 #1 SMP Mon Dec 15 18:15:07 UTC 2008 i686
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Mon Mar 23 16:33:15 2009 from red.local
user@blue:~$
```

Now that you have seen how some simple routing and filtering commands are implemented, this might help in debugging otherwise inexplicable failures.

You may adapt this to your own situation, go on to try out the various GUI solutions to the same general problem, or examine how more complex networks are administered.

I should point out that if the blue computer is in fact an access point for the internet, dynamic IP addresses are handed out and the POSTROUTING rule needs to be amended slightly as shown:

```
# iptables -t nat -A POSTROUTING -o ath0 -j MASQUERADE
```

Useful References

"Linux Firewalls (Second Edition)" Robert L. Ziegler (ISBN 0-7357-1099-6; New Riders 2002).