



Bluetooth Networking to the Internet

by [Muhammad Al-Ismail](#) on May 26, 2004

In an effort to exercise new Linux capabilities, I have been messing with the BlueZ open source **implementation of bluetooth for the past 2 months. I have managed to get a working NAT network** over bluetooth connecting my laptop and desktop to the internet. Now, I can surf the internet from my laptop with no wires being attached. This article describes what is needed in order to get this done.

The following is assumed:

1. A recent kernel with the [BlueZ](#) package installed; including SDP, PAN, BLUEZ-UTILS and BLUEZ-LIBS. Instructions could be found [here](#) by Marcel Holtmann. Run the following command to check that you have bluetooth running:

```
$ /usr/sbin/hciconfig -a
```

It should say something like UP RUNNING. Please note down the BD Address which is a 6 bytes address in the following format:

```
00:10:EC:71:F9:D6
```

The response I get from the above command is

```
hci0:  Type: USB
      BD Address: 00:10:EC:71:F9:D6 ACL MTU: 192:8  SCO MTU: 64:8
      UP RUNNING PSCAN ISCAN AUTH ENCRYPT
      RX bytes:512035 acl:3921 sco:0 events:17349 errors:0
      TX bytes:5177521 acl:27340 sco:0 commands:25 errors:0
      Features: 0xff 0xff 0x0f 0x00
      Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
      Link policy: HOLD SNIFF PARK
      Link mode: SLAVE ACCEPT
      Name: 'Grand Tux'
      Class: 0x120104
      Service Classes: Networking, Object Transfer
      Device Class: Computer, Desktop workstation
      HCI Ver: 1.1 (0x1) HCI Rev: 0x110 LMP Ver: 1.1 (0x1) LMP Subver: 0x110
      Manufacturer: Cambridge Silicon Radio (10)
```

Note that the name of your Bluetooth device is controlled by

```
/etc/bluetooth/hcid.conf
```

change the line

```
name "%h-%d";
```

to

```
name "Grand Tux";
```

You also have to setup a PIN access code to your system. For example, if you would like to set it to 1234 then do the following

```
$ echo "1234" > /etc/bluetooth/pin  
$ chmod 600 /etc/bluetooth/pin
```

You will need root access to apply the above changes. Moreover, restart the bluetooth service to re-confirm name and pin changes

```
$ /etc/init.d/bluetooth restart
```

2. **A working bridge utility**. Refer to the respective [HOWTO](#) for instructions. Fedora is shipped with that, so you most probably have it installed. Do the following to check that you have the bridge utility:

```
$ /usr/sbin/brctl
```

if not then get it from your fedora installation CDs/DVD.

3. iptables has to be installed and loaded. Do the following the check that it is installed.

```
$ /sbin/iptables -V  
$ /sbin/lsmmod | grep ip_tables
```

You have to create a bnep0 configuration file on the server side. Here is what to do in /etc/sysconfig/network-scripts/ifcfg-bnep0

```
DEVICE=bnep0  
ONBOOT=no  
BOOTPROTO=DHCP
```

On the Server Side Do:

```
$ /usr/sbin/brctl addbr pan0  
$ /sbin/ifconfig pan0 10.0.0.1  
$ /usr/sbin/brctl setfd pan0 0  
$ /usr/sbin/brctl stp pan0 disable  
$ /sbin/modprobe bnep  
$ pand -s -M --role=NAP
```

On the Client Side Do:

```
$ /sbin/modprobe bnep  
$ pand -c 00:10:EC:71:F9:D6  
$ /sbin/ifconfig bnep0 10.0.0.2 netmask 255.255.255.0  
$ /sbin/route add default gw 10.0.0.1
```

On the Server Side Do:

```
$ /usr/sbin/brctl addif pan0 bnep0  
$ /sbin/ifconfig bnep0 0.0.0.0
```

enable IP forwarding on the Server Side

```
$ echo "1" > /proc/sys/net/ipv4/ip_forward  
$ /sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  
$ /sbin/iptables -A FORWARD -i pan0 -j ACCEPT  
$ /sbin/iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

For Internet Sharing on Server Side Do:

```
$ /sbin/iptables -t nat -A POSTROUTING -j MASQUERADE
```

Save your iptables setting so you don't have to re-do them again

```
$ /sbin/iptables-save
```

In addition, I also allow DHCP and DNS on the Server for total enjoyment. I have given my server and client both ethernet and bluetooth names. Bluetooth speed is sufficient for surfing the net but not for copying huge files for example. Here is a listing of what I got in my `/etc/hosts` file:

```
192.168.0.1    server.home.net server    localhost.localdomain  localhost
192.168.0.2    laptop.home.net laptop
10.0.0.1      btserver.home.net btserver
10.0.0.2      btclient.home.net btserver
```

Add the lines below to your `/etc/named.conf`. Note that you have to change `ip_esp_dns1` and `ip_esp_dns2` to your service provider DNS IP addresses. I use a dial-up connection to go online and this way, allows me to do DNS caching on my server side. Hence, it lowers DNS traffic from my side to my ISP.

```
forwarders { ip_esp_dns1; ip_esp_dns2; 192.168.0.1 };
allow-query { 192.168.0.0/24; 127.0.0.1/32; 10.0.0.2; 10.0.0.1 };
```

However, for DHCP, I have only allowed it for ethernet. This is what I have in my `/etc/dhcpd.conf`

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option routers 192.168.0.1;
option domain-name-servers 192.168.0.1;
option domain-name "home.net";
ddns-update-style ad-hoc;
option netbios-name-servers 192.168.0.1;
option netbios-dd-server 192.168.0.1;
option netbios-node-type 8;
option netbios-scope "";

subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.2 192.168.0.254;
}
```

You could do the same thing for your bluetooth network by creating another subnet. Check the output below to see how different it is when I ping over ethernet and bluetooth from the client side

```
$ ping -c 2 btserver
PING btserver.home.net (10.0.0.1) 56(84) bytes of data.
64 bytes from btserver.home.net (10.0.0.1): icmp_seq=0 ttl=64 time=36.3 ms
64 bytes from btserver.home.net (10.0.0.1): icmp_seq=1 ttl=64 time=23.9 ms

--- btserver.home.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 23.962/30.170/36.378/6.208 ms, pipe 2

$ ping -c 2 server
PING server.home.net (192.168.0.1) 56(84) bytes of data.
64 bytes from server.home.net (192.168.0.1): icmp_seq=0 ttl=64 time=0.236 ms
64 bytes from server.home.net (192.168.0.1): icmp_seq=1 ttl=64 time=0.269 ms

--- server.home.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.236/0.252/0.269/0.022 ms, pipe 2
```

Conclusion

I like to thank all the people whom without their various hints and tips, this article would have not been possible. Now, disconnect your ethernet cables. Lay on bed and enjoy blue internet.

References

- Bluez Mailing List
<http://bluez.sourceforge.net/mlist.html>
- HOWTO Setup Common PAN Scenarios with BlueZ
<http://bluez.sourceforge.net/contrib/HOWTO-PAN>
- HOWTO Linux iptables
<http://www.linuxguruz.org/iptables/howto/iptables-HOWTO.html>

Copyright © 2003-2006 FedoraNEWS.ORG. All rights reserved.