This page describes an unusual use case: I have an old Fonera 2100 A/B/C that provides a FON hotspot, which pre-dates FON's tie-up with British Telecom (in the United Kingdom). This entitles me to free access to BTWiFi in exchange for sharing my own connection. I also have a shiny new R7000, and I'd like to run the FON hotspot on my primary router and kill off the old Fonera to save power (or re-purpose it). Obviously, we can't register the R7000 with FON, so we're reliant on using the credentials of our previously registered Fonera.

### Contents

- 1 Configure the FON heartbeat
- 2 Install chillispot
- 3 Create /jffs/chilli.conf
- 4 Create wireless interface
- 5 Start chillispot
- 6 Verify that DHCP works
- 7 Set up NAT
- 8 Secure the local network
- 9 Run chillispot on startup
- 10 Reboot-proof test
- 11 Speed restrictions

### Configure the FON heartbeat

All routers running official FON firmware send a "heartbeat" signal back to fon every few hours once they're connected to the internet, mainly to update the router's online status on maps.fon.com and let FON know you're part of their community. To do this, you need to implement the FON Heartbeat in DD-WRT by downloading the heartbeat code originally created by Freddy. NOTE: The heartbeat uses port 1937 to send information so ensure that this port is not blocked by a firewall on your router. If you're unsure, it's probably open anyway. Ensure Cron is enabled on the Admin > Management web page and then type (or paste) the following in via SSH to download the files.

```
cd /jffs
wget http://www.inaudible.co.uk/perm/fonkey
wget http://www.inaudible.co.uk/perm/nolog/thinclient
chmod 755 /jffs/thinclient
ls
```

After the ls, ensure that you see both the fonkey and thinclient files listed, the latter being a light green colour. Now modify the thinclient script, hard-coding the MAC addresses you want to use under WLMAC and ETMAC. Now start the heartbeat script by typing the following code into telnet:

```
/jffs/thinclient cron
```

After a minute or two wait, the following line should be printed out, but with your own Wireless and LAN MAC addresses listed.
Ignore the final warning: the heartbeat should now be up and running and appear active on maps.fon.com.
Now go on and add a cron job to run the heartbeat every half an hour, i.e.

```
24,54 * * * * root /jffs/thinclient cron > /dev/null 2>&1 &
```

and you should hopefully find yourself with a working FON heartbeat. So far as FON is concerned, our R7000 is indistinguishable from a Fonera and if the heartbeat is up, we're online.

### Install chillispot

```
opkg install chillispot
```

This will return some errors regarding kernel modules; ignore them.

### Create /jffs/chilli.conf

```
radiusserver1 radius01.fon.com
radiusserver2 radius02.fon.com
radiussecret garrafon
macauth
radiusnasid 00-18-84-XX-XX-XX
uamsecret garrafon
dhcpif wl0.1
uamanydns
uamallowed maps.fon.com, c20.statcounter.com, crl.comodoca.com
uamallowed 208.67.222.222
uamallowed 208.67.220.220
uamserver https://www.fon.com/login/gateway
net 192.168.182.0/24
dynip 192.168.182.0/24
dns1 208.67.222.222
dns2 208.67.220.220
```

### Create wireless interface

Via the DD-WRT GUI, create an additional VAP wl0.1 and set it to unbridged, AP isolation, net isolation, no wireless security. Give it an SSID, e.g. FON. Set an IP address that is NOT in 192.168.182.x

### Start chillispot

```
chilli --conf /jffs/chilli.conf
```

Configure the FON heartbeat
Resolve any minor config errors that may prevent chillispot from starting, there are always some...

**Verify that DHCP works**

You should now be able to connect a client to the FON SSID and get an IP address in 192.168.182.x. You should be able to ping 192.168.182.1, but not 192.168.1.1 or anything in your local network. (if you can, something is probably wrong with the setup of your virtual AP.)

**Set up NAT**

Normally, a Fonera runs as a secondary device behind a primary NAT router, so traffic is routed between chillispot's "tun0" and the LAN interface. Here, we are trying to run it on our primary NAT router, so we need to tell DD-WRT that any traffic from tun0 (chillispot) going out of vlan2 (WAN) is to be masqueraded to our public IP address. Something like

```bash
iptables -t nat -A POSTROUTING -s 192.168.182.0/24 -o vlan2 -j MASQUERADE
```

should do the trick, and chillispot clients should now be duly redirected to the FON login page, able to log in, and able to browse the Web once logged in.

**Secure the local network**

Our chillispot clients can also access everything in the local network, which is almost certainly not what we want, so let's set up some firewalling. We also exempt clients in 192.168.182.0/24 from being sent through privoxy.

```bash
iptables -t nat -A POSTROUTING -s 192.168.182.0/24 -o vlan2 -j MASQUERADE
iptables -I INPUT -s 192.168.182.0/24 -d 192.168.1.1 -j logdrop
iptables -I FORWARD -i tun0 -o vlan2 -m state --state NEW -j ACCEPT
iptables -I FORWARD -i br0 -o tun0 -j logdrop
iptables -A FORWARD -i tun0 -j DROP;
iptables -t nat -I PREROUTING -s 192.168.182.0/24 -p tcp --dport 80 -j RETURN
iptables -t nat -I PREROUTING -i tun0 -d 192.168.182.1/32 -j ACCEPT
```

At this point, you should be able to connect to anything in the wider Internet (subject to FON login), but not to anything in the local network. Add the relevant commands to your firewall script to make them persistent.

**Run chillispot on startup**

Add

```bash
/opt/usr/sbin/chilli --conf /jffs/chilli.conf
```

to your startup scripts.
Reboot-proof test

Reboot the router and verify that chilli is started automatically, that proper firewall rules are created, and generally that a reboot of the router leaves everything working as expected.

Speed restrictions

During initial testing I was able to achieve ~35Mbit down and ~5Mbit up from a laptop connected to the FON hotspot, which is more bandwidth than I want to share. From the QoS page, add a restriction by netmask 192.168.182.0/24 and set the WAN bandwidth as desired. I've set this to a fairly generous 30Mbit down and 3Mbit up, which is 25% of the rated speed of my cable connection.