

Setting up a Raspberry Pi as a Wireless Access Point

You will need:

- A Raspberry Pi (of course!)
- An SD Card for your Pi – I use 4GB for this but whatever you have should be good
- A USB Wifi Dongle – mine was a ZyXEL Communications Corp. ZyAIR G-202 802.11bg

First install Raspbian from the Raspberry Pi site - <http://www.raspberrypi.org/downloads> – at the time of writing the current version is 'Wheezy'.

Install the image to your SD card as explained here - http://elinux.org/RPi_Easy_SD_Card_Setup

Log in to your Pi – I did mine via SSH but no reason why you can't do it via a keyboard and screen if you have it connected that way. The default login is username 'pi' and password 'raspberrypi'. You'll be wanting to change these later for security. If not logging in via SSH make sure you have a network connection with internet access. If you're using SSH then you'll need to locate your Pi's IP address on your LAN using (on Linux systems, at least) **sudo nmap -sP 192.168.0.0/24** which will list all IP's on your network (if your network is 192.168.1.x then change the network range in the command to 192.168.1.0/24 and so on). You'll get a list along the lines of...

```
Nmap scan report for UNKNOWN (192.168.0.54)
Host is up (0.65s latency).
MAC Address: 00:24:2C:12:62:55 (Hon Hai Precision Ind. Co.)
Nmap scan report for UNKNOWN (192.168.0.55)
Host is up (0.23s latency).
MAC Address: B8:27:EB:9E:CA:4D (Raspberry Pi Foundation)
Nmap scan report for UNKNOWN (192.168.0.57)
Host is up (0.0036s latency).
MAC Address: 00:A0:DE:86:D3:6B (Yamaha)
```

In this case the Pi is at 192.168.0.55 so I'd login with **ssh pi@192.168.0.55**.

Firstly run **sudo raspi-config** if you haven't yet, to setup your Pi (you'll likely want to at least change your memory split to 16) then reboot and set a root password using **sudo passwd** and enter a decent (i.e. not 'password!') password for the root user. From this point on I'll assume you're logged in as root.

I prefer using **aptitude** for package installs and will be using it throughout this article – if you're happier using apt-get then you'll be wanting to alter some of the installation commands to suit. Install aptitude with **apt-get install aptitude**.

Bring your Pi up to date using **aptitude update; aptitude safe-upgrade** – this may take a little while and also depend on your internet connection speed.

You'll need to install a few packages:

```
aptitude install rfkill zd1211-firmware hostapd hostap-utils iw
dnsmasq
```

These are:

- rfkill – Wireless utility
- zd1211-firmware – Software for dealing with zd1211-based Wireless hardware
- hostapd – The hostap wireless access point daemon
- hostap-utils – Tools that go with hostap
- iw – Wireless config utility
- dnsmasq – a DHCP and DNS utility

You may also want to add 'vim' to that list which is a nice nicer console editor than the default 'vi'.

The **zd1211-firmware** package is hardware-specific but is a very common implementation for wifi dongles. Your dongle will also need to support 'AP' mode which you can verify by typing **iw list** and under the 'Supported interface modes' you will hopefully see 'AP' listed along with others like 'managed' and 'monitor'. If not you're out of luck with your particular dongle.

Your wireless interface must be set statically for hostap, edit your /etc/network/interfaces file to look like this (or copy and paste!):

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp

iface wlan0 inet static
address 192.168.1.1
netmask 255.255.255.0
```

Then restart your wireless interface using **ifdown wlan0; ifup wlan0** which should hopefully go smoothly without errors. The address of 192.168.1.1 should *not* be the same as the network connected to eth0, my main LAN is 192.168.0.0/24 and the wireless interface should be a different network. This adds a simple layer of security in case you're configuring an open access point as I was and allows you to firewall one network from the other far more easily.

Now to configure hostap. Edit /etc/hostapd/hostap.conf (it may not already exist but this will create it, anyway) to look like this:

```
interface=wlan0
driver=nl80211
ssid=test
channel=1
```

The settings are pretty obvious, 'driver' being the only exception, just leave it as it is but change the other values as you see fit though these should do for an initial test. One thing to bear in mind is that hostap seems to be very literal about reading its configuration file so make sure you have no trailing spaces on the end of any lines! All being well you should now see 'test' as a wireless network now though a little more work needs to be done to connect to it yet.

The final step is to configure dnsmasq so you can obtain an IP address from your new Pi-Point. Edit your `/etc/dnsmasq.conf` file to look like this:

```
# Never forward plain names (without a dot or domain part)
domain-needed

# Only listen for DHCP on wlan0
interface=wlan0

# create a domain if you want, comment it out otherwise
#domain=Pi-Point.co.uk

# Create a dhcp range on your /24 wlan0 network with 12 hour lease time
dhcp-range=192.168.1.5,192.168.1.254,255.255.255.0,12h

# Send an empty WPAD option. This may be REQUIRED to get windows 7 to behave.
#dhcp-option=252,"\n"
```

Remember to change the `dhcp-range` option for the network IP range you're using. If you're having problems with Windows7 machines then try uncommenting the last line. Restart dnsmasq by typing **`service dnsmasq restart`** for the configuration to take effect. Now you should be able to connect to your new Pi-Point and get a proper IP address from it.

Now your Pi-Point is up and running there's a few things to take care of. You'll probably have noticed that you can't load Google or anything else. That's because although you can connect to the Pi-Point and get an IP address from it, it's not yet routing your requests across your network and out onto the internet. We'll address this next.

The first thing you need to do is to allow forwarding in the Pi-Point, this is achieved by issuing the command **`echo 1 > /proc/sys/net/ipv4/ip_forward`** to turn forwarding on. Now your Pi-Point is forwarding packets you need it to provide NAT between your wifi network and your main network which is attached to the internet. Use the command **`iptables -t nat -A POSTROUTING -j MASQUERADE`** to achieve this.

You should now have a working Pi-Point which will allow user to connect wirelessly and use the internet via your internet connection!

One final thing – to ensure your Pi-Point works from a reboot you'll need to create a run file to turn on forwarding, nat and run hostap at boot time. Create a file /etc/init.d/pipoint with these contents:

```
#!/bin/sh
# Configure Wifi Access Point.
#
### BEGIN INIT INFO
# Provides:          WifiAP
# Required-Start:    $remote_fs $syslog $time
# Required-Stop:     $remote_fs $syslog $time
# Should-Start:      $network $named slapd autofs ypbind nscd nslcd
# Should-Stop:       $network $named slapd autofs ypbind nscd nslcd
# Default-Start:     2
# Default-Stop:
# Short-Description: Wifi Access Point configuration
# Description:       Sets forwarding, starts hostap, enables NAT in iptables
### END INIT INFO

# turn on forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

# enable NAT
iptables -t nat -A POSTROUTING -j MASQUERADE

# start the access point
hostapd -B /etc/hostapd/hostapd.conf

# restart dhcp
service dnsmasq restart
```

Next add the script to the startup sequence using **update-rc.d pipoint start 99 2**. This should ensure your Pi-Point should reboot as a functioning wifi access point.

Further things to try:

- Add wireless WEP/WPA authentication via the hostap configuration
- Strengthen firewalling between the Pi-Point and your LAN with iptables
- Add a Squid proxy to reduce traffic across your LAN
- Add a MAC filter with iptables or dnsmasq
- Don't forget to change your default login password!