

[Get started](#)

Turn a RaspBerryPi 3 into a WiFi router-hotspot



Edo Scalafiotti

[Follow](#)

Jun 17, 2016 · 5 min read



Following this tutorial you'll be able to configure your RaspBerryPi model 3 to accept incoming connections using its **built-it WiFi antenna** and to forward the requests to the Ethernet port.

Requirements

- 1x RaspBerryPi model 3 (earlier versions will not work unless you have a WiFi dongle. In that case, just follow this [Adafruit tutorial](#))
- 1x Ethernet cable
- Ethernet access

Context

My current ISP provided me with an old, locked WiFi router that is only capable to broadcast 20mb/s, which is not great for my nominal 100mb/s connection.

Furthermore, it messes up my home devices assigning them to different subnets (and sometimes, it mixes local devices with ipv4 and ipv6). Using a Chromecast or a simple Plex server is therefore becoming a bit of an hassle and I thought for a moment that I

[Get started](#)

useful WiFi router? Please note that this project is basically a proof of concept: the on-board WiFi antenna hasn't a great range and the top-speed goes barely over 30mb/s. However, seen that this method works, I'll try to run it with a powerful WiFi USB dongle with AP support to see if the PI can be converted to a more serious 300mb/s router.

Also, this tutorial is a slight modification from this Adafruit tutorial that is meant for RaspberryPi < v3 and requires a WiFi dongle.

Let's get on with it, shall we?

Initial checks

Once logged in to your Pi (via Terminal, on another computer), check that everything is setup:

```
ifconfig -a
```

```
# You should see something like this:
#eth0      Link encap:Ethernet  HWaddr b8:27:eb:77:8a:b2
#          inet addr:192.168.2.3  Bcast:192.168.2.255
Mask:255.255.255.0
#          inet6 addr: fe80::395:c29a:6e8:3756/64 Scope:Link
#          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
#          RX packets:1257 errors:0 dropped:0 overruns:0 frame:0
#          TX packets:932 errors:0 dropped:0 overruns:0 carrier:0
#          collisions:0 txqueuelen:1000
#          RX bytes:417879 (408.0 KiB)  TX bytes:165808 (161.9 KiB)
#
#lo        Link encap:Local Loopback
#          inet addr:127.0.0.1  Mask:255.0.0.0
#          inet6 addr: ::1/128 Scope:Host
#          UP LOOPBACK RUNNING  MTU:65536  Metric:1
#          RX packets:325 errors:0 dropped:0 overruns:0 frame:0
#          TX packets:325 errors:0 dropped:0 overruns:0 carrier:0
#          collisions:0 txqueuelen:1
#          RX bytes:26584 (25.9 KiB)  TX bytes:26584 (25.9 KiB)
#
#wlan0     Link encap:Ethernet  HWaddr b8:27:eb:22:df:e7
#          inet addr:192.168.42.1  Bcast:192.168.42.255
Mask:255.255.255.0
#          inet6 addr: fe80::6b78:7810:9062:550a/64 Scope:Link
#          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
#          RX packets:24737 errors:0 dropped:25 overruns:0 frame:0
#          TX packets:841 errors:0 dropped:0 overruns:0 carrier:0
```



Generally, **eth0** is the ethernet connection and **wlan0** the WiFi connection. If both are there, we're good to go.

Software setup

We'll install and configure 2 programs:

- hostapd
- isc-dhcp-server

```
sudo apt-get update
sudo apt-get install hostapd isc-dhcp-server
```

DHCP server

```
# Be wise and always make a backup of the default config
sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.default
# Edit the default config file
sudo nano /etc/dhcp/dhcpd.conf

# Comment the following lines...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

# ...to read:
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;

# ...and un-comment this line
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# ... to read:
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
```

[Get started](#)

```
option broadcast-address 192.168.42.255;
option routers 192.168.42.1;
default-lease-time 600;
max-lease-time 7200;
option domain-name "local";
option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

Now, with this configuration we are assigning the subnet **192.168.42.10–50** (40 devices in total) and we are configuring our WiFi local IP address to be **192.168.42.1**. While we're at it, we're assigning Google's public DNS: **8.8.8.8, 8.8.4.4**.

Next, let's specify on what interface should the DHCP server service DHCP requests (**wlan0** in this case):

```
sudo nano /etc/default/isc-dhcp-server

# Edit this line:
INTERFACES=""

# ...to read
INTERFACES="wlan0"
```

Let's setup wlan0 for static IP:

```
# First, shut it down...
sudo ifdown wlan0

# ...keep it safe and make a backup file:
sudo cp /etc/network/interfaces /etc/network/interfaces.backup

# ...edit the network interfaces file:
sudo nano /etc/network/interfaces

# ... edit accordingly to read:
source-directory /etc/network/interfaces.d

auto lo
```

[Get started](#)

```
allow-hotplug wlan0

iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0
    post-up iw dev $IFACE set power_save off

# ...close the file and assign a static IP now
sudo ifconfig wlan0 192.168.42.1
```

Hostapd

```
# Create a file and edit it:
sudo nano /etc/hostapd/hostapd.conf

# ...modify ssid with a name of your choice and wpa_passphrase to a
WiFi authen
interface=wlan0
ssid=WiPi
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=xyz
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Some tutorials requires you to set a driver ID. If you need to do that, in order to check what is your current driver ID, run:

```
basename $( readlink /sys/class/net/wlan0/device/driver )
```

...but even though my driver ID reads **brcmfmac_sdio**, if I put it into the hostapd.conf file I still got an error, but commenting out does the trick.

[Get started](#)

```
# Create a backup file
sudo cp /etc/sysctl.conf /etc/sysctl.conf.backup

# ...edit the config file
sudo nano /etc/sysctl.conf

# ...un-comment or add to the bottom:
net.ipv4.ip_forward=1

# ...and activate it immediately:
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"

# ...modify the iptables to create a network translation between eth0
and the wifi port wlan0
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT

# ...make this happen on reboot by running
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"

# ...and editing again
sudo nano /etc/network/interfaces

# ...appending at the end:
up iptables-restore < /etc/iptables.ipv4.nat

# Our /etc/network/interfaces file will now look like this:
source-directory /etc/network/interfaces.d

auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0

iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0

source-directory /etc/network/interfaces.d
```

Let's test our access point by running:

[Get started](#)

```
# Your hotspot is up and running: try to connect to it from a
computer or a smartphone. When you do so, you should also see some
log activity on your terminal. If you're satisfied, stop it with
CTRL+D
```

```
# Let's clean everything:
sudo service hostapd start
sudo service isc-dhcp-server start
```

```
# ...and make sure that we're up and running:
sudo service hostapd status
sudo service isc-dhcp-server status
```

```
# ...let's configure our daemons to start at boot time:
sudo update-rc.d hostapd enable
sudo update-rc.d isc-dhcp-server enable
```

```
# ...reboot the pi.
sudo reboot
```

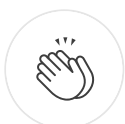
You should now be able to see your pi WiFi, connect to it and access internet to it. As a quick comparison, streaming 4k videos will consume about 10% of the pi CPU so... use it accordingly.

As a bonus, if you want to check what's happening on your WiFi hotspot, check the log file:

```
tail -f /var/log/syslog
```

If you really feel *naughty*, (or if you want to prevent naughty behaviour from your connected devices...) let me know in the comments below if you're interested in a **tutorial on how to configure a honeypot with your pi.**

Raspberry Pi DIY



455 claps



[Get started](#)

Edo Scalafiotti

Hacker in Chief. Fascinated by nature - and bees. Also, dark chocolate.

[Follow](#)[See responses \(40\)](#)

More From Medium

Related reads

[Get started](#)

Running Debian Linux on Android device natively



The Quantified VC

Sep 12, 2016 · 17 min read



249



Related reads

How to make a DIY home alarm system with a raspberry pi and a webcam



Alex Nikolaidis

Sep 15, 2013 · 7 min read



555



Related reads

[Get started](#)

Kali Linux 2018.2 on your Pocket with the GPD 7 mini-laptop.



Tomas C. in HackerNoon.com

Jul 25, 2018 · 4 min read



252

