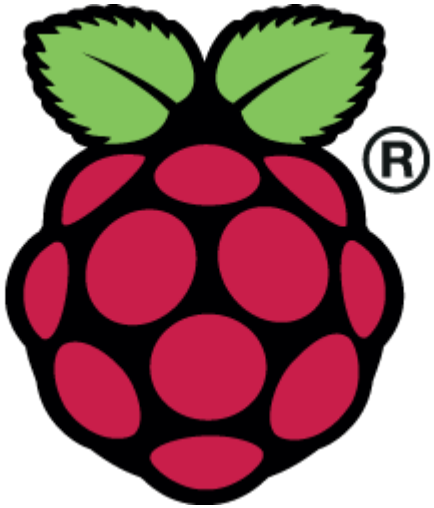
How to use your Raspberry Pi as a wireless access point

TOPICS: Router Wi-Fi Wireless Access Point



POSTED BY: STEPHEN LOVELY NOVEMBER 30, 2017

The Raspberry Pi can do a lot, especially now that the new Raspberry Pi comes with wireless capabilities already on board. It can take the place of a ton of different (and more expensive) devices – including a router! If you turn your Raspberry Pi into a wireless access point, you can make it act as a router. It's not the most powerful thing in the world, but it does work, and the project is a lot of fun.

How to use your Raspberry Pi as a wireless access point

We're going to get into the command line a bit here, but this project isn't really all that difficult. All we're really doing is using Raspbian and installing a couple packages that give the Pi the ability to do router-like things like assign IP addresses to devices that connect to it.

RECENT POSTS

Step 1: Install and update Raspbian

Check out our complete guide to installing Raspbian for the details on this one. Then plug everything in and hop into the terminal and check for updates and ugrades:

```
sudo apt-get update
sudo apt-get upgrade
```

If you get an upgrade, It's a good idea to reboot with **sudo reboot**.

Step 2: Install hostapd and dnsmasq

These are the two programs we're going to use to make your Raspberry Pi into a wireless access point. To get them, just type these lines into the terminal:

```
sudo apt-get install hostapd
sudo apt-get install dnsmasq
```

Both times, you'll have to hit y to continue. hostapd is the package that lets us create a wireless hotspot using a Raspberry Pi, and dnsmasq is an easy-to-use DHCP and DNS server.

We're going to edit the programs' configuration files in a moment, so let's turn the programs off
before we start tinkering:

How to use your Raspberry Pi as a wireless access point
How to use your Raspberry Pi to monitor broadband speed
How to run Raspberry Pi Desktop on Windows or macOS
How to back up your Raspberry Pi
How to create a Raspberry Pi thermometer using the Sense HAT

TOPICS

Chrome Chromium DOSBox Eltechs ExaGear FileZilla FTP games Kodi LEMP Mac OS X media server MS-DOS MySQL NAS Netflix nginx NOOBS OpenELEC OpenMediaVault OpenVPN operating systems

```
sudo systemctl stop hostapd
sudo systemctl stop dnsmasq
```

Step 3: Configure a static IP for the wlan0 interface

For our purposes here, I'm assuming that we're using the standard home network IP addresses, like 192.168.###.###. Given that assumption, let's assign the IP address **192.168.0.10** to the wlan0
interface by editing the dhcpcd configuration file. Start editing with this command:

```
sudo nano /etc/dhcpcd.conf
```

Now that you're in the file, add the following lines at the end:

```
interface wlan0
static ip_address=192.168.0.10/24
denyinterfaces eth0
denyinterfaces wlan0
```

(The last two lines are needed in order to make our bridge work –- but more on that in
Step 8.)

OSMC PHP Plex port forwarding PuTTY Raspbian RealVNC remote desktop access RetroPie router Sense HAT SFTP SSH VNC VNC Connect VNC Server VNC Viewer VPN VyprVPN Wi-Fi Widevine Windows wireless access point

After that, press **Ctrl+X**, then **Y**, then **Enter** to save the file and exit the editor.

Step 4: Configure the DHCP server (dnsmasq)

We're going to use dnsmasq as our DHCP server. The idea of a DHCP server is to
dynamically distribute network configuration parameters, such as IP addresses, for
interfaces and services.

dnsmasq's default configuration file contains a lot of unnecessary information, so
it's easier for us to start from scratch. Let's rename the default configuration file and
write a new one:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
sudo nano /etc/dnsmasq.conf
```

You'll be editing a new file now, and with the old one renamed, this is the config file that dnsmasq will use. Type these lines into your new configuration file:

```
interface=wlan0
  dhcp-range=192.168.0.11,192.168.0.30,255.255.255.0,24h
```

The lines we added mean that we're going to provide IP addresses between 192.168.0.11 and 192.168.0.30 for the wlan0 interface.

Step 5: Configure the access point host software (hostapd)

Another config file! This time, we're messing with the hostapd config file. Open 'er up:

```
sudo nano /etc/hostapd/hostapd.conf
```

This should create a brand new file. Type in this:

```
interface=wlan0
bridge=br0
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
ssid=NETWORK
wpa_passphrase=PASSWORD
```

Note that where I have “NETWORK” and “PASSWORD,” you should come up with your own names. This is how you’ll join the Pi’s network from other devices.

We still have to show the system the location of the configuration file:

```
sudo nano /etc/default/hostapd
```

In this file, track down the line that says #DAEMON_CONF=”” – delete that # and put the path to our config file in the quotes, so that it looks like this:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

The # keeps the line from being read as code, so you’re basically bringing this line to life here while giving it the right path to our config file.

Step 6: Set up traffic forwarding

The idea here is that when you connect to your Pi, it will forward the traffic over your Ethernet cable. So we’re going to have wlan0 forward via Ethernet cable to your modem. This involves editing yet another config file:

```
sudo nano /etc/sysctl.conf
```

Now find this line:

```
#net.ipv4.ip_forward=1
```

...and delete the "#" – leaving the rest, so it just reads:

```
net.ipv4.ip_forward=1
```

Step 7: Add a new iptables rule

Next, we're going to add IP masquerading for outbound traffic on eth0 using iptables:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

...and save the new iptables rule:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

To load the rule on boot, we need to edit the file /etc/rc.local and add the following
line just above the line exit 0:

```
iptables-restore < /etc/iptables.ipv4.nat
```

Step 8: Enable internet connection

Now the Raspberry Pi is acting as an access point to which other devices can connect. However, those devices can't use the Pi to access the internet just yet. To make the possible, we need to build a bridge that will pass all traffic between the wlan0 and eth0 interfaces.

To build the bridge, let's install one more package:

```
sudo apt-get install bridge-utils
```

We're ready to add a new bridge (called br0):

```
sudo brctl addbr br0
```

Next, we'll connect the eth0 interface to our bridge:

```
sudo brctl addif br0 eth0
```

Finally, let's edit the interfaces file:

```
sudo nano /etc/network/interfaces
```

...and add the following lines at the end of the file:

```
auto br0
iface br0 inet manual
bridge_ports eth0 wlan0
```

Step 9: Reboot

Now that we're ready, let's reboot with **sudo reboot.**

Now your Pi should be working as a wireless access point. Try it out by hopping on another device and looking for the network name you used back in step 5.

This piece has been updated for Raspbian Stretch and is based on the instructions created by GitHub user SurferTim.

Previous post

20 COMMENTS ON "HOW TO USE YOUR RASPBERRY PI AS A WIRELESS ACCESS POINT"

flasterofreality | July 9, 2017 at 3:51 pm | Reply

This is one of the best tutorials I have seen for the Pi3 with using onboard Wi-Fi! Thanks for including the /etc/rc.local config for iptables that often gets overlooked.

Evan | July 20, 2017 at 11:58 am | Reply

Thanks for this, it works better than several other tutorials I've tried.

But, what I'm really looking for is not a hotspot with its own IP range, but an extension of my existing Ethernet network with DHCP done on the main router. So that all devices can see each other. Do you have any ideas about how to accomplish that? There is a guide on the Raspberry Pi website, which uses bridge, but it doesn't work.

Anders J | December 10, 2017 at 3:45 am | Reply

Try to remove all configurations of dnsmasq and dhcp. Uninstall dnsmasq.
Then remove the firewall settings, as it starts masquerading, that is, make the RPi and other LAN on different IP ranges with NAT. You don't want that.

Set the RPi to get IP address from dhcp or static IP from the other networks network address. I would try dhcp first.

Good luck!

The rest should be enough to make it work after reboot.

Riley20019 | November 25, 2017 at 8:10 pm | Reply

Can anyone help me, I have it all working but when i try to connect it says that there is no internet connection and it will not fully connect it will load forever. Do I have to port-forward the Raspberry Pi on my router or is it something i have to fix in the code?

Anders J | December 10, 2017 at 3:39 am | Reply

What is the network address of the network you connect to eth0? If it is 192.168.0.0/24 (that is the IP address prefix is 192.168.0.) you should change the prefixes on your configuration on the RPi to something else like 192.168.42 or

something. The number 42 could be anything between 1 and 255.

So, we need to see what router you uses.

Try the command ‘ip route’ on the RPi.

Corey | December 3, 2017 at 12:19 pm | Reply

So I followed SurferTims tutorial and pretty much ended up doing what you’ve got here except with different IPs — do you know how to add a captive portal to this AP so it points to an Apache web server on the Raspberry Pi?

Phil | December 3, 2017 at 11:17 pm | Reply

Really good, worked well, with VNC its a simple cut and paste

Deki | December 5, 2017 at 4:56 pm | Reply

Hello and thank you for the tutorial. It’s the only one working since updating to Stretch… 🙂

Now that I've replicated the tutorial, I'd like to implement a slightly different version of it. I'd like to use the Raspberry Pi 3 as an access point AND as a client simultaneously. This is for a little project I'm working on (building a weather station). This should be possible with the Wi-Fi chipset built into the Raspi. There is no need for the clients that access the Raspi AP to have internet connectivity, so there's no need for forwarding traffic (only the Raspi needs internet connectivity). I'm rather new to networking on Linux, so I don't bring much experience to the table. Unfortunately, there is no other way than this to realize the project. Can this be done by simply altering some of the above steps? What would be the right way of doing so? If not, could anyone give me some pointers to it?

I have found some tutorials for doing so, but all of them work with /etc/network/interfaces. Unfortunately, editing anything in this file breaks access to wireless connectivity… ("no wireless interface found")

Hopefully someone can help me. Thanks in advance!

Kind regards,
Deki

Viktor | December 6, 2017 at 10:04 am | Reply

Could you please say why it is losing connection in 30 sec after start? (I can't see SSID in Wi-Fi list and no connection between the RPi and my phone.)

Viktor | December 8, 2017 at 11:13 pm | Reply

Unfortunately I did exactly as described (twice) and I can’t connect to the Raspberry Pi.

Viktor | December 9, 2017 at 1:18 am | Reply

I tried use it on another Raspberry Pi – the same problem.

Anders J | December 10, 2017 at 3:34 am | Reply

Great tutorial. But some things.

Don’t use 192.168.0.0/24 or 192.168.1.0/24, as they are usually used in default configuration on home routers. Use a random number x in range 2-255 in 192.168.x.0/24 for your network. Then the risk of trouble with routing will be less. Even if you have MASQUERADE on.

The firewall rules should also be set up, as NAT is not a secure firewall.

I recommend using ufw there, with a new enough version also do support being a router.

Ro | December 10, 2017 at 4:41 pm | Reply

Thanks for this guide!

All other solutions I tried sometimes Wi-Fi is not available any more and i have to reboot the RPi.

Is there a way for this setup to check if it's up and running and (automatic) take steps needed to get it up and running again?

MeInterwebs | December 12, 2017 at 12:50 am | Reply

It would be helpful to see what the resulting ifconfig and route output looks like with some explanation, so that we could confirm that we did everything right and could make adjustments for our own specific situation with a deeper understanding. Thanks.

Jouni | December 17, 2017 at 2:46 pm | Reply

Million thanks to you! I've been trying to setup wireless Pi AP for two days now, and this tutorial finally did it! What other tutorials around the web didn't have was the bridge part, and they left you with access point sure but without any connection to the net.

Agustin Lobo | December 22, 2017 at 5:31 pm | Reply

Is it possible to use this approach to use an Android device as a remote desktop of the RPi (using VNC) when there is no network (i.e., in the field)? Thanks.

Eid Elagha | December 23, 2017 at 8:01 am | Reply

when I used the command:
sudo brctl addif br0 eth0

It shown an error: Can’t add wlan0 to bridge br0: Operation not supported

Please help to fix this error!

Agustin Lobo | December 27, 2017 at 12:48 pm | Reply

Thanks for the guide.

I think that, in case you are not interested on Step 8 (i.e., you just want to access your Raspberry Pi through SSH or VNC even in the absence of Wi-Fi network), you also have to avoid the two lines with

denyinterfaces in /etc/dhcpcd.conf (Step 3). At least this is how it works for me. Please confirm.

David Whiteley | December 28, 2017 at 2:28 pm | Reply

Many thanks for this AP, I've been working on it for about two months.

My application is remote temperature monitoring of bee hives.

My set-up is AP–Pi-Zero-W–USB/Ethernet–Home Router(Wi-Fi disabled)–Debian PC along with Wi-Fi devices Win7Lap Top and Galaxy/Android
AP static IP 192.168.168.10, range …11 to …30.

The Android/Firefox goes online through the AP just fine.

Android/Fing and Win7/Control Panel list Gateway as 198.168.0.1, i.e., the home router, not the Pi AP at 192.168.168.10.

Win7/Firefox not happy.

On Android/Termius/BotSync/etc attempts to reach the Pi at 192.168.168.10 are thwarted.

What might I try to log on to the Pi? From Android for Terminal/rsync, etc.

Kirk | December 29, 2017 at 7:10 pm | Reply

Thanks for the how-to on this. I'm looking to turn my Pi into a VPN, and got to this tutorial from the VPN post.

After going through these instructions, I'm not seeing a broadcast of my Pi's network. When I reboot, the Wi-Fi connects automatically to the last hotspot, but I've tried turning that off as well with no luck.

Am I missing anything? Thanks for any insight.

Leave a comment

Your email address will not be published.

Comment

Name *

Email *

Website

Post Comment

PRIVACY POLICY