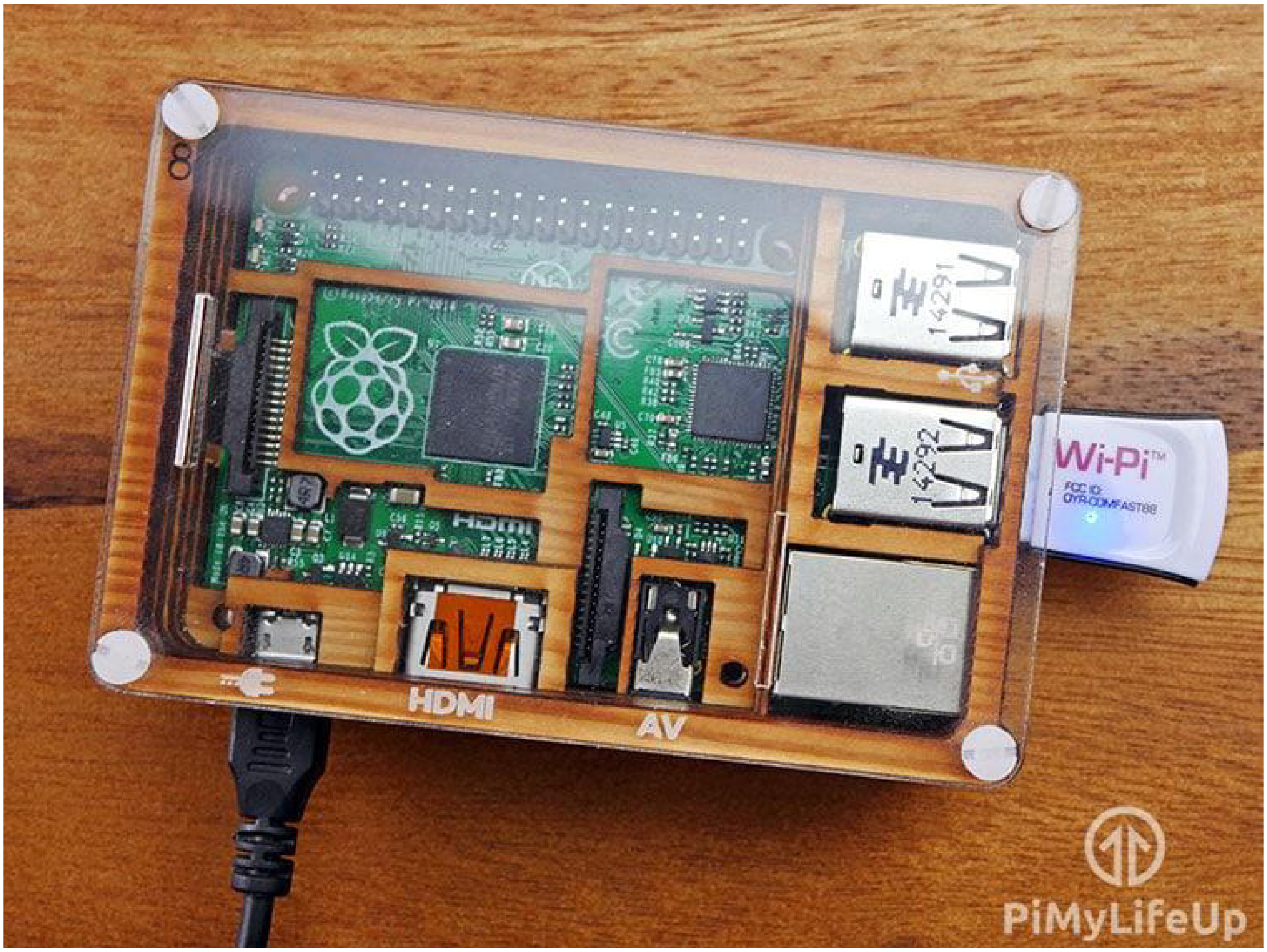
PiMyLifeUp

Easy Raspberry Pi WiFi Bridge

by Gus May 8, 2017 Updated Jul 29, 2018 Beginner, Network



A Raspberry Pi WiFi bridge is one of the best ways of providing internet access to a device that only supports an Ethernet connection.

In this tutorial, we will show you how to setup a WiFi bridge, and how to setup dnsmasq correctly to allow any device to connect through your Pi without issues.

You will need to keep in mind that you will not see speeds as good as what you would with a direct connection to your router. As there is some overhead with the connection having to run through the Raspberry Pi.

Remember to do this tutorial you will need either a WiFi dongle or a Raspberry Pi 3 with the inbuilt WiFi module.

This tutorial can be combined with our basic VPN access point, you can find the tutorial on how to set that up directly after this tutorial. Basically, it will show how to setup a OpenVPN client and redirect all traffic through that client.

Please note: This tutorial will require some slight changes if you decide to go down the VPN route, we will explain the necessary changes needed at the end of this tutorial.

Equipment List

Below are all the bits and pieces that I used for this Raspberry Pi WiFI bridge tutorial, you will need A Wireless internet connection to be able to complete this tutorial.

Recommended:

Raspberry Pi 2 or 3

Micro SD Card

Wifi dongle (The Pi 3 has WiFi inbuilt)

Ethernet Connection

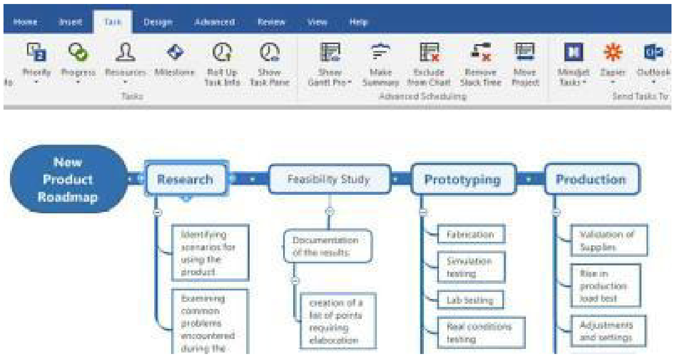
Optional:

Raspberry Pi Case

Setting up the WiFi Bridge

To setup the Raspberry Pi Wifi bridge we will be utilizing the dnsmasq package, this package handles most of the grunt work for this tutorial.

Dnsmasq is a package that acts as both a local DHCP server and a local DNS server. We utilize this package so that we can assign IP addresses and process DNS requests through the Raspberry Pi itself and act like a router. One of the bonuses to utilizing *dnsmasq* is that it is very easy to configure while being somewhat lightweight in comparison to the *isc-dhcp-server* and *bind9* packages.



Remember for this tutorial you will need to have an active WiFi router to connect to and an ethernet device you intend on bridging the Wi-Fi connection to.

1. Before we get started with installing and setting up our packages, we will first run an update on the Raspberry Pi by entering the following two commands into the terminal.

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

2. With that done we can now install the one and only package we will be utilizing, run the following command to install dnsmasq.

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

3. Before we get too far ahead of ourselves, we should setup the wlan0 connection that we plan on using. If you have already setup your wireless connection then you can skip ahead to step 5.

Otherwise open up the wpa_supplicant file by running the following command:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

4. Within this file add the following, making sure you replace the ssid with the name of the network you want to connect to and replace the psk value with the password for that network.

```
network={
        ssid="networkname"
        psk="networkpassword"
}
```

5. With the wireless network now setup to correctly connect we can proceed with setting up our *eth0 interface*. This will basically force it to use a static IP address, not setting this up can cause several issues.

To do this we need to modify the dhcpcd.conf file by running the following command:

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

⚠ **Important Note:** If you're on Raspbian stretch then wlan0 and eth0 may need to be changed. Use the `ifconfig` command to see the new names, they're likely quite long and will contain the MAC address.

Make sure you update these for all the commands in this tutorial.

6. Within this file we need to add the following lines, make sure you replace eth0 with the correct interface of your ethernet. Raspbian Stretch uses what is known as predictable names, so they may differ from this tutorial. To find out what your interfaces are you can utilize the `ifconfig` command.

```
interface wlan0
static ip_address=192.168.220.1/24
static routers=192.168.220.0
```

Now we can save and quit out of the file by pressing *Ctrl+X* then pressing *Y* and then *Enter*.

7. With our changes made to dhcpcd configuration we should now restart the service by running the following command:

```
sudo service dhcpcd restart
```

8. Before we get started with modifying dnsmasq's configuration we will first make a backup of the original configuration by running the following command.

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

9. With the original configuration now backed up and moved out of the way we can now move on and create our new configuration file by typing the command below into the terminal.

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

10. Now that we have our new file created we want to add the lines below, these lines basically tell the dnsmasq package how to handle DNS and DHCP traffic.

```
interface=eth0       # Use interface eth0
listen-address=192.168.220.1   # Specify the address to listen on
bind-interfaces      # Bind to the interface
server=8.8.8.8       # Use Google DNS
domain-needed        # Don't forward short names
bogus-priv           # Drop the non-routed address spaces.
dhcp-range=192.168.220.50,192.168.220.150,12h # IP range and lease time
```

Now we can save and quit out of the file by pressing *Ctrl+X* then pressing *Y* and then *Enter*.

11. We now need to configure the Raspberry Pi's firewall so that it will forward all traffic from our eth0 connection over to our wlan0 connection. Before we do this we must first enable ipv4p IP Forwarding through the sysctl.conf configuration file, so let's begin editing it with the following command:

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

12. Within this file you need to find the following line, and remove the # from the beginning of it.

Find:

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

Replace with:

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

Now we can save and quit out of the file by pressing *Ctrl+X* then pressing *Y* and then *Enter*.

13. Now since we don’t want to have to wait until the next reboot before the configuration is loaded in, we can run the following command to enable it immediately.

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

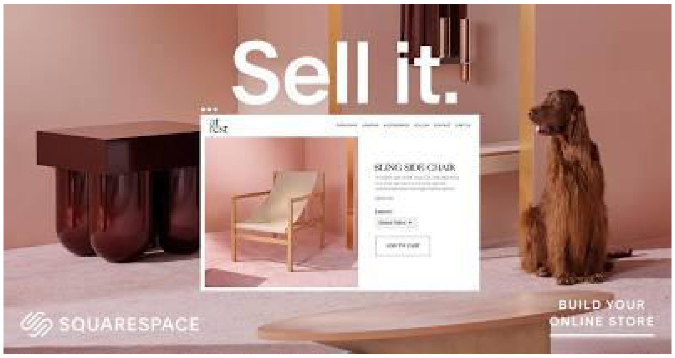
14. Now that IPv4 Forwarding is enabled we can reconfigure our firewall so that traffic is forwarded from our eth0 interface over to our wlan0 connection. Basically this means that anyone connecting to the ethernet will be able to utilize our wlan0 internet connection.

Run the following commands to add our new rules to the iptable:

```
sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
sudo iptables -A FORWARD -i wlan0 -o eth0 -m state --state RELATED,ESTABLISHED -j
sudo iptables -A FORWARD -i eth0 -o wlan0 -j ACCEPT
```

⚠ **Note:** If you get errors when entering the above lines simply reboot the Pi using `sudo reboot`.

15. Of course iptables are flushed on every boot of the Raspberry Pi so we will need to save our new rules somewhere so they are loaded back in on every boot.



To save our new set of rules run the following command:

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

16. Now with our new rules safely saved somewhere we need to make this file be loaded back in on every reboot. The most simple way to handle this is to modify the rc.local file.

Run the following command to begin editing the file:

```
sudo nano /etc/rc.local
```

17. Now we are in this file, we need to add the line below. Make sure this line appears above *exit 0*. This line basically reads the settings out of our iptables.ipv4.nat file and loads them into the iptables.

Find:

```
exit 0
```

Add ABOVE:

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

Now we can save and quit out of the file by pressing *Ctrl+X* then pressing *Y* and then *Enter*.

18. Finally all we need to do is start our dnsmasq service. To do this, all you need to do is run the following command:

```
sudo service dnsmasq start
```

19. Now you should finally have a fully operational Raspberry Pi WiFi Bridge, you can ensure this is working by plugging any device into its Ethernet port, the bridge should provide an internet connection to the device you plugged it into.

To ensure everything will run smoothly it's best to try rebooting now. This will ensure that everything will successfully re-enable when the Raspberry Pi is started back up. Run the following command to reboot the Raspberry Pi:

```
sudo reboot
```

Setting up the Raspberry Pi WiFi Bridge with a VPN

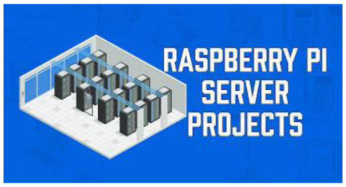
This tutorial is fully compatible with the basic VPN router tutorial. However there is one small change you will have to make in step 13, rather than using the commands showcased there, run the commands below.

Basically the main change you will see here is that instead of redirecting the traffic from wlan0 through the tunnel we will be redirecting the traffic from our eth0 connection to the tunnel.

```
sudo iptables -t nat -A POSTROUTING -o tun0 -j MASQUERADE
sudo iptables -A FORWARD -i tun0 -o eth0 -m state --state RELATED,ESTABLISHED -j A
sudo iptables -A FORWARD -i eth0 -o tun0 -j ACCEPT
```

The rest of the VPN Access Point tutorial can be done without any other changes.

Hopefully by now you should have a fully operational Raspberry Pi WiFi Bridge. If you come across any issues or have some feedback related to this tutorial, then please don't hesitate to leave a comment below.



14+ Raspberry Pi Server Projects

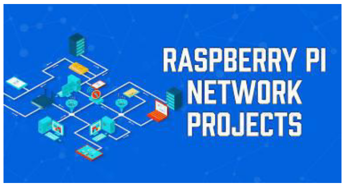
pimylifeup.com

Build a Raspberry Pi Webcam Server in Minutes

pimylifeup.com

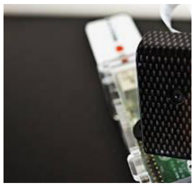
How to setup a Raspberry Pi TOR Access Point

pimylifeup.com



7+ Raspberry Pi Network Tutorials

pimylifeup.com



Build a Raspb
Security Cam
Network

pimylifeup.com

Subscribe to get our FREE Crash Course to the Raspberry Pi.

email

Raspberry Pi Wireless Access Point

Simple Raspberry Pi WiFi Extender

How to setup a Raspberry Pi TOR Access Point

Setting up a Raspberry Pi Captive Portal

Raspberry Pi VPN Access Point: Setup a Basic VPN Router

Raspberry Pi Steam Link

27 Comments

Ed Manning III on May 9, 2017 at 10:20 am

I got up to step 13 and threw an error

modprobe: ERROR: ../libkmod/libkmod.c:557 kmod_search_moddep() could not open moddep file '/lib/modules/4.4.21-v7+/modules.dep.bin'
iptables v1.4.21: can't initialize iptables table `nat': Table does not exist (do you need to insmod?)
Perhaps iptables or your kernel needs to be upgraded.

I rebooted and it lost it's IP

Adrian on June 11, 2017 at 11:41 pm

Same result..

Gus on June 26, 2017 at 1:32 am

I'm looking into this issue, rebooting will fix the issue but you shouldn't lose your IP.

Luigi Pezzullo on January 29, 2018 at 4:07 am

Ho una smart TV con connessione wifi e LAN.
Essendo posizionata lontana dal router la utilizzo solitamente con il wifi.
Utilizzando il suo Pi Wifi Bridge avrei una velocità di connessione migliore o peggiore del wifi?
Grazie
——
I have a smart TV with wifi and LAN.
Being positioned far from the router, I usually use it with wifi.
Using your Pi Wifi Bridge would I have a connection speed better or worse than wifi?
Thank you

Gus on January 30, 2018 at 2:24 pm

Hey Luigi,

The Raspberry Pi hasn't got the fastest WiFi chip or ethernet chip, plus the overhead in it having to run through the Raspberry Pi.
I personally reckon you would be getting a worse connection then connecting straight to your WiFi.

Cheers,
Gus

David GIRAULT on May 9, 2017 at 4:41 pm

Step 6 to 9 configure eth0 instead of wlan0 interface. DHCP/DNS services should run on wlan0 interface, which requires static IP.

David GIRAULT on May 9, 2017 at 4:44 pm

Sorry. I read too fast. It's a bridge for Ethernet device. You're right.

Josh on June 15, 2017 at 6:27 am

Got this when starting dnsmasq.

```
Job for dnsmasq.service failed. See 'systemctl status
dnsmasq.service' and 'journalctl -xn' for details.
```

Josh on June 15, 2017 at 6:34 am

and journalctl shows no journals

Gus on June 26, 2017 at 1:33 am

Not sure what's happened here, did you receive any errors earlier on in the tutorial ?

lingam mohan krishnasen on June 27, 2017 at 11:55 am

actually I am also getting the same error, please give the pic of the ipv4 file, also dns config
following is errors

```
Jun 27 01:53:34 raspberrypi dnsmasq[1255]: dnsmasq: failed to
create listening socket for 192.168.1.1: Cannot assign
requested address
Jun 27 01:53:34 raspberrypi systemd[1]: dnsmasq.service:
control process exited, code=exited status=2
Jun 27 01:53:34 raspberrypi systemd[1]: Failed to start
dnsmasq - A lightweight DHCP and caching DNS server.
Jun 27 01:53:34 raspberrypi systemd[1]: Unit dnsmasq.service
entered failed state.
```

Gus on July 4, 2017 at 3:15 pm

Make sure you **have not** defined eth0 twice in the interfaces file. This will cause an error like this to appear.

Hugh on July 7, 2017 at 11:10 am

Hello. Is there any way to bridge the raspberry pi wifi bridge to my home network? I have the raspberry pi wifi bridge connected to a switch and would like to talk to the other devices from my 192.168.1.x network to the pi's 192.168.220.x network.

pictsidhe on January 17, 2018 at 6:38 am

This tutorial doesn't make the pi a dhcp server. If you want to use it on 192.168.1.*, you'll need to make it's fixed ip 192.168.1.*, preferably not in the range handed out by your dhcp server or things get interesting.

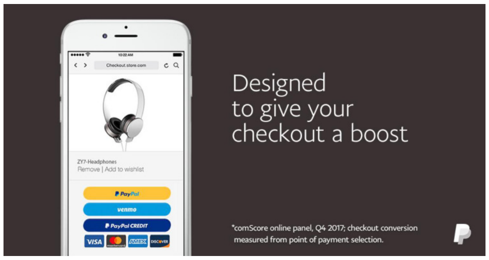
Tim on July 7, 2017 at 12:15 pm

How much of a hit did your connection speed take when using this setup? I have a 92Mbps down connection normally on my pc wired directly to my router, but when testing with the Rpi3 I'm only getting 4Mbps down. And my wireless router is just in the next room. Any ideas why that might be? I'm not sure what to troubleshoot.

Marcus on July 8, 2017 at 1:04 pm

Should step 13 read "-i wlan0" instead of "-i wlan"?

Also, important to note that this setup is a wifi client NAT router, not technically a bridge. It is not possible for clients on the wifi side of the raspberry Pi to directly connect to clients connected to the Pi's eth0.


Designed
to give your
checkout a boost
*comScore online panel, Q4 2017; checkout conversion measured from point of payment selection.

I can confirm the NAT config you've detailed does work on a Pi2 w/ Jessie though, so thanks for the writeup!

Gus on July 8, 2017 at 1:13 pm

Hi Marcus,

Yep you are correct, it should be wlan0, thanks for pointing that out.

Andrew Nasson on July 11, 2017 at 1:39 pm

So I looked into this project and it looks amazing but before I dive in I'm wondering if it is compatible with the original Pi + a wifi dongle. Thanks!

Red Baron on July 19, 2017 at 3:12 pm

As long as the dongle is compatible with the Pi – yes.

Red Baron on July 19, 2017 at 3:18 pm

Great tutorial. It works like a charm. The only thing that is missing is a description of what to do on the machine that you want to connect to the wifi through the Pi. ☺ On Windows you should leave it at “Obtain IP address” (the IPv4 properties of the connection once you connect both devices through Ethernet). On Linux you should have your connection set at “Automatic (DHCP) for you IPv4 settings.

Steve on August 3, 2017 at 11:20 pm

Same question as Hugh. I want to be able to connect a BluRay with an ethernet connection to the Rpi then via WiFi to my router which has my NAS drive plugged in to one of its ports… Hopefully someone can help.

Not bothered about connecting my Blu Ray to the internet – just want it to be able to play stuff on my NAS drive which has DLNA on it.

I can confirm the above tutorial works with the original 256MB RPi B.

Matt on August 19, 2017 at 6:34 am

How exactly does one go about changing the text in step 11? I have tried both the write out function and the cut text function and have not gotten any results.

Ron on August 20, 2017 at 4:43 am

I can not get this to work on a Pi3 with built in wifi. wlan0 will not “see” any ssid’s (three wifi routers running). Tried twice with fresh Jessie 2017-06-21 image –> update –> upgrade.
ifconfig -a does show the wlan0. eth0 does show static ip.

Ron on August 20, 2017 at 4:59 am

Update:
The “sudo iwlist wlan0 scan” does see the two closest wifi ssid’s.

Michael on September 25, 2017 at 1:59 pm

Is there a way to use hotel wifi that uses a splash screen as the wifi source?

Paulinho on October 7, 2017 at 11:09 am

Hi
By doing using the Raspberry Pi as a Wi-Fi Bridge will it still be free to other purposes or will only work as a WiFi bridge and nothing else?

Andrew Schott on January 7, 2018 at 2:49 pm

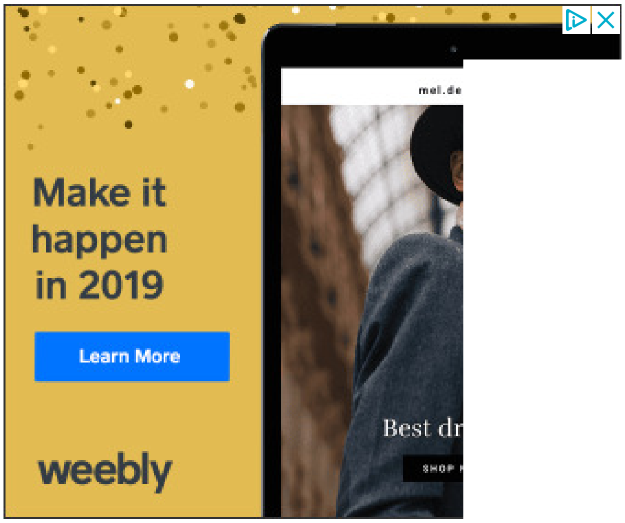
Yeah you can do other things with it

Sorry comments are closed! For help, feedback and more, head over to the forums.

Follow us on social

Search for Tutorials

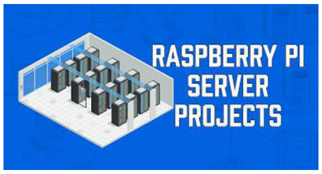
Search



Bitdefender ANTIVIRUS PLUS
Bitdefender INTERNET SECURITY
Bitdefender TOTAL SECURITY

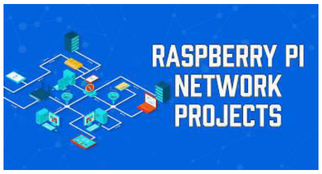
Build a Raspberry Pi Webcam Server in Minutes

pimylifeup.com



14+ Raspberry Pi Server Projects

pimylifeup.com

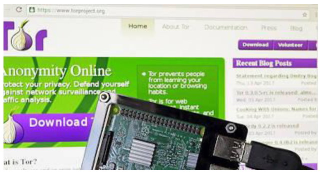


7+ Raspberry Pi Network Tutorials

pimylifeup.com

Build a R Pi Securi Network

pimylifeup.com



How to se Raspberr Access P

pimylifeup.com