(https://switchedtolinux.com/)

Wireless Internet Passed to Ethernet with Raspberry Pi

December 28th, 2022 | Tips & Tricks (https://switchedtolinux.com/category/tips-tricks/) | No Comments Yet

The Article

This guide will teach you how to use a Raspberry Pi (https://amzn.to/3hR36Zf) to connect to a wireless network such as a hotspot, phone, or a free wireless access point, and pass the Internet to your router to share the Internet connection with all the resources on your network. This is a good backup solution for your network, or a means to share Internet with an off-grid setup.

The data flow in this project will connect the pi to the Internet via the wlan port and pass the data through a DHCP server passing the connection to the eth port on the pi.

Equipment

You will need a Raspberry Pi that has a wireless port and an ethernet port. I am using a Raspberry Pi 4 with 4GB RAM. While the Pi 3b may work, the fourth series has a gigabit port and a better wireless chip.

We also need a screen, and one with a portable setup will work better. Since this device sits above your network, you will not be able to SSH into it for regular work. I am using a generic 7 inch ISP screen (https://amzn.to/3jxZDiN) that is powered by a USB port on the pi.

To assist with the portability of the system, I am using a Rii mini X1 keyboard (https://amzn.to/3WLHlJa). This small device has a mouse and keyboard built on one unit. Of course, any keyboard will work, but I kept my system to the smallest form factor.

Make sure you have a good Ethernet cable, cat 5e or cat 6 for the best data transfers possible.

For my power, I usually use my solar generator, but that is overkill if you don't already have one. A portable cell phone charger capable of 3A power output (https://amzn.to/3GqcNHi) is a good portable solution or a basic AC adapter will work if you have a solid power supply in your setup.

Procedure

My procedure was adapted from Konamiman's article on WiFi for Ethernet-only devices via Raspberry Pi. His setup was for a specific application, and his also did not include any consideration for VPNs, so I made some adaptations here. Also, the original guide has a few errors, most of which are corrected in the comments, so I will fix those in my guide.

First, we need to edit this file:

`/etc/dhcpcd.conf`

Add:

```
interface eth0
static ip_address=192.168.34.1/24
```

COMMENT OUT everything else in the file. The other lines in the file define things the eth0 port will manage that will interfere with the setup.

Next, we need to install and configure the ISC DHCP Server.

Install with this command:

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

Configure the DHCP server to setup the subnet, netmask, range, and other options by editing the following file:

`/etc/dhcp/dhcpd.conf`

Add this to the bottom of the file:

```
authoritative;
subnet 192.168.34.0 netmask 255.255.255.0 {
 range 192.168.34.10 192.168.34.250;
 option broadcast-address 192.168.34.255;
 option routers 192.168.34.1;
 default-lease-time 600;
 max-lease-time 7200;
 option domain-name "local-network";
 option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

Assign the ethernet port as the default interface for this DHCP server by editing this file:

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

Update the INTERFACESv4 channel

```
INTERFACESv4="eth0"
```

(add eth0 into the blank INTERFACESv4, leave v6 blank)

With all the changes to the server configurations, we are ready to start (or restart) the server:

```
service isc-dhcp-server start
```

Now we need to enable the server to forward the eth0 port that we set above. We will set how the forwarding works below, this just sets up the server ability to forward. Edit this file.

`/etc/sysctl.conf`

Add this line:

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

The following step is where I am setting up my system for manually starting the server. I want to run my server manually because I need to contend with captive portal logins on free networking, AND because the forwarding stack will change depending on whether I am using the VPN or not. If you have a consistent wireless network, the following file can be called with a cron job on startup, but I will be executing the port forwarding manually.

To run this manually, I start by creating a file on the desktop (or some other place to easily execute). Here is the contents of the file:

```
echo Starting DHCP server
service isc-dhcp-server start

echo Setting NAT routing
iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
iptables -A FORWARD -i wlan0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth0 -o wlan0 -j ACCEPT

DEFAULT_IFACE=`route -n | grep -E "^0.0.0.0 .+UG" | awk '{print $8}'`
if [ "$DEFAULT_IFACE" != "wlan0" ]
then
  GW=`route -n | grep -E "^0.0.0.0 .+UG .+wlan0$" | awk '{print $2}'`
  echo Setting default route to wlan0 via $GW
  route del default $DEFAULT_IFACE
  route add default gw $GW wlan0
fi
```

I named my file:

`isc-server`

Start the server with this command:

```
sudo bash isc-server
```

This will start the server, forward the IP tables, and make sure that the wireless is the default port to bring in WAN via wlan0 and pass the Internet to the LAN via eth0

Now your server will be setup and passing your wireless network through the ethernet port.

Add the VPN

In my case, I need to use a VPN because I will frequently using public wireless. The VPN will keep my connections secure. In this case, I am using my private VPN built on a Linode server. This protocol will work with any VPN provider that supplies a .ovpn file to connect to the VPN.

Here are the steps.

Start by installing openvpn:

```
sudo apt install openvpn
```

Download the .ovpn access file and call the VPN connection:

```
sudo openvpn {file}.ovpn
```

Enter your username and password, wait for the connecton.

The VPN will be passing all network traffic through a new route: tun0 (as opposed to eth0 or wlan0). To account for this, I will create a new file based on the previous one and replace all instances of wlan0 with tun0:

```
echo Starting DHCP server
service isc-dhcp-server start

echo Setting NAT routing
iptables -t nat -A POSTROUTING -o tun0 -j MASQUERADE
iptables -A FORWARD -i tun0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i eth0 -o tun0 -j ACCEPT

DEFAULT_IFACE=`route -n | grep -E "^0.0.0.0 .+UG" | awk '{print $8}'`
if [ "$DEFAULT_IFACE" != "tun0" ]
then
  GW=`route -n | grep -E "^0.0.0.0 .+UG .+tun0$" | awk '{print $2}'`
  echo Setting default route to tun0 via $GW
  route del default $DEFAULT_IFACE
  route add default gw $GW tun0
fi
```

I will call this file the same as I called the prior one:

```
sudo bash isc-server-vpn
```

Now my system will pass the VPN data into the ethernet port, and if the VPN connection fails, all network traffic will stop.

Wrap Up

This setup is helps me save on Internet bandwidth when I am in a city with public wifi available. In other situations it can help with backup Internet, off-grid setups where your Internet connect is not routed directly to your location, or even as a means to pass newer Starlink Internet to a wired network if your Starlink router does not have an ethernet port.

« This REAL Facial Recognition Case is Terrifying (https://switchedtolinux.com/this-real-facial-recognition-case-is-terrifying/)

TikTok Is Used to Spy on US Reporters (https://switchedtolinux.com/tiktok-is-used-to-spy-on-us-reporters/) »

Video

Notes and References

Affilates

Support Switched to Linux with our affiliates:

Amazon (http://amzn.to/2oFr4Wa)

Web Hosting:
A2Hosting (https://tlm.li/a2h)
siteground (https://tlm.li/sgh)

VPNs:
Get a VPN to stay private online.
NordVPN (https://tlm.li/nord)
Private Internet Access (https://tlm.li/pia)

Podcasting:
Looking at Podcasting? These links will help with either hosting the podcasts or viewing stats.
Blubrry Podcast Hosting (http://create.blubrry.com/resources/podcast-media-hosting/?code=tlm)
Blubrry Podcasting Stats (http://create.blubrry.com/resources/podcast-media-download-statistics/?code=tlm)

Search Website

Search

Sponsored

Recent Articles

TikTok Is Used to Spy on US Reporters (https://switchedtolinux.com/tiktok-is-used-to-spy-on-us-reporters/)
Wireless Internet Passed to Ethernet with Raspberry Pi (https://switchedtolinux.com/wireless-internet-passed-to-ethernet-with-raspberry-pi/)
This REAL Facial Recognition Case is Terrifying (https://switchedtolinux.com/this-real-facial-recognition-case-is-terrifying/)
Social Media is Under Attack (https://switchedtolinux.com/social-media-is-under-attack/)
Is Your Car Infotainment System Safe? (https://switchedtolinux.com/is-your-car-infotainment-system-safe/)

Site logo

Our site logo is inspired by the Font Awesome 'Toggle On (http://fontawesome.io/icon/toggle-on/)' button, the Linux color scheme, and the official Linux mascot, 'Tux' the Penguin. Tux is used with acknowledged permission of Larry Ewing (http://isc.tamu.edu/%7Elewing/linux/) and the GIMP Project (http://www.gimp.org/). Basically we wanted the logo to say, "Linux is ON".

YouTube (https://www.youtube.com/channel/UCoryWpk4QVYKFCJul9KBdyw)

Privacy Policy (https://switchedtolinux.com/privacy-policy/)

Terms of Use (https://switchedtolinux.com/terms-of-use/)

Quick Menu

Home (https://switchedtolinux.com/)

Help with Linux (https://switchedtolinux.com/help-with-linux/)

Contact (https://switchedtolinux.com/another-linux-site/contact/)

Recent Articles (https://switchedtolinux.com/recent-updates/all-categories/)

How to Get Started

(https://switchedtolinux.com/how-to-get-started/)