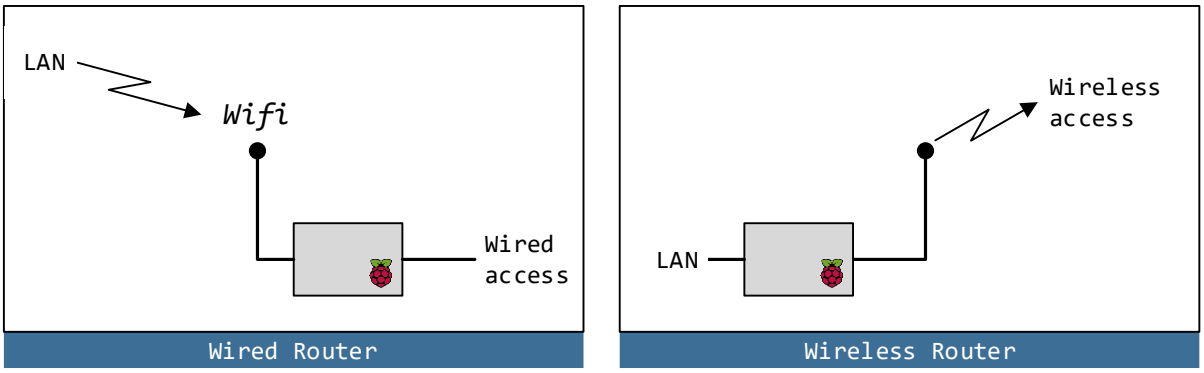
Configure a Raspberry Pi as a Router

Author: *Emanuel Oliveira*

1 INTRODUCTION

This document describes the steps to configure a Raspberry Pi as a wired and wireless router.



Throughout this document, the following convention will be used:

Commands:

```
$ command
```

File contents:

```
4 This is line 4 of the file
5 This is line 5
```

The line number might not match the actual line number in the file.

2 REQUIREMENTS

The following will be required for this procedure:

- A Raspberry Pi already set up with internet access.
  See *Basic Configuration of a Raspberry Pi* [TN1255001].

3 WIRED ROUTER

It will be required internet access on the Raspberry Pi through the WiFi connection.

3.1 Install Software

Start by installing the packages **dnsmasq** required to manage the network (DNS and DHCP).

```
$ sudo apt install dnsmasq
```

Install **netfilter-persistent** and its plugin **iptables-persistent**.

The following helps by saving firewall rules and restoring them when the Raspberry Pi boots.

```
$ sudo DEBIAN_FRONTEND=noninteractive apt install -y netfilter-persistent iptables-persistent
```

3.2 Set up the network router

To configure the IP address of the wired connection, edit the file `/etc/dhcpcd.conf` with the following contents:

```
44 interface eth0
45 static ip_address=192.168.4.2/24
```

3.3 Enable routing and IP masquerading

To enable routing, i.e., to allow traffic to flow from one network to the other in the Raspberry Pi, create a file `/etc/sysctl.d/routed-ap.conf` with the following contents:

```
1 # https://www.raspberrypi.org/documentation/configuration/wireless/access-point-routed.md
2 # Enable IPv4 routing
3 net.ipv4.ip_forward=1
```

Add the following single firewall rule and save it:

```
$ sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
$ sudo netfilter-persistent save
```

3.4 Configure the DHCP and DNS services for the wireless network

Save the default configuration file with the following command:

```
$ sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.sav
```

Them, edit (create) the file `/etc/dnsmasq.conf` with the following contents:

```
1 interface=eth0 # Listening interface
2 dhcp-range=192.168.4.64,192.168.4.95,255.255.255.0,24h
3                # Pool of IP addresses served via DHCP
4 domain=lan     # Local wireless DNS domain
5 address=/gw.lan/192.168.4.2
5                # Alias for this router
```

4 WIRELESS ROUTER

It will be required internet access on the Raspberry Pi through the wired connection.

The steps presented in this section summarise the steps presented in the official Raspberry Pi website.

4.1 Install Software

Start by installing the packages `hostapd` and `dnsmasq`, which are required to manage the access point and network (DNS and DHCP), respectively.

```
$ sudo apt install hostapd
$ sudo apt install dnsmasq
```

Enable the wireless access point service and set it to start at boot time.

```
$ sudo systemctl unmask hostapd
$ sudo systemctl enable hostapd
```

Install `netfilter-persistent` and its plugin `iptables-persistent`.

The following helps by saving firewall rules and restoring them when the Raspberry Pi boots.

```
$ sudo DEBIAN_FRONTEND=noninteractive apt install -y netfilter-persistent iptables-persistent
```


4.2 Set up the network router

To configure the Access Point (static) IP address, edit the file `/etc/dhcpcd.conf` with the following contents:

```
64 interface wlan0
65 static ip_address=192.168.5.1/24
66 nohook wpa_supplicant
```

4.3 Enable routing and IP masquerading

To enable routing, i.e., to allow traffic to flow from one network to the other in the Raspberry Pi, create a file `/etc/sysctl.d/routed-ap.conf` with the following contents:

```
1 # https://www.raspberrypi.org/documentation/configuration/wireless/access-point-routed.md
2 # Enable IPv4 routing
3 net.ipv4.ip_forward=1
```

To allow traffic between clients on this foreign wireless network and the internet without changing the main router's configuration, the Raspberry Pi can substitute the IP address of wireless clients with its own IP address on the LAN using a "masquerade" firewall rule.

- The main router will see all outgoing traffic from wireless clients as coming from the Raspberry Pi, allowing communication with the internet.
- The Raspberry Pi will receive all incoming traffic, substitute the IP addresses back, and forward traffic to the original wireless client.

This process is configured by adding a single firewall rule:

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

Next, just save the rule[1].

```
$ sudo netfilter-persistent save
```

4.4 Configure the DHCP and DNS services for the wireless network

Save the default configuration file with the following command:

```
$ sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.sav
```

Them, edit (create) the file `/etc/dnsmasq.conf` with the following contents:

```
1 interface=wlan0 # Listening interface
2 dhcp-range=192.168.5.2,192.168.5.20,255.255.255.0,24h
3                 # Pool of IP addresses served via DHCP
4 domain=wlan     # Local wireless DNS domain
5 address=/gw.wlan/192.168.5.1
5                 # Alias for this router
```

The list of DHCP leases can be found in the file `/var/lib/misc/dnsmasq.leases`.

4.5 Ensure wireless operation

```
sudo rfkill unblock wlan
```

4.6 Configure the access point software

Create the file `/etc/hostapd/hostapd.conf` with the following contents:

```
 1 country_code=GB
 2 interface=wlan0
```

[1] Filtering rules are saved to the directory /etc/iptables/.

```
 3 ssid=NameOfNetwork
 4 hw_mode=g
 5 channel=7
 6 macaddr_acl=0
 7 auth_algs=1
 8 ignore_broadcast_ssid=0
 9 wpa=2
10 wpa_passphrase=ThisIsThePassword
11 wpa_key_mgmt=WPA-PSK
12 wpa_pairwise=TKIP
13 rsn_pairwise=CCMP
```

The value hw_mode[2] is set according to the band to be used as per the following:

- a = IEEE 802.11a (5 GHz)
- b = IEEE 802.11b (2.4 GHz)
- g = IEEE 802.11g (2.4 GHz)

The configuration is now completed. Reboot.

[2] Note that when changing the hw_mode, you may need to also change the channel – see **Error! Reference source not found.**.

NOTES: