DOCUMENTATION > CONFIGURATION > WIRELESS > ACCESS-POINT

Setting up a Raspberry Pi as a Wireless Access Point

Before proceeding, please ensure your Raspberry Pi is up to date and rebooted.

Setting up a Raspberry Pi as an access point in a standalone network (NAT)

The Raspberry Pi can be used as a wireless access point, running a standalone network. This can be done using the inbuilt wireless features of the Raspberry Pi 3 or Raspberry Pi Zero W, or by using a suitable USB wireless dongle that supports access points.

Note that this documentation was tested on a Raspberry Pi 3, and it is possible that some USB dongles may need slight changes to their settings. If you are having trouble with a USB wireless dongle, please check the forums.

To add a Raspberry Pi-based access point to an existing network, see this section.

In order to work as an access point, the Raspberry Pi will need to have access point software installed, along with DHCP server software to provide connecting devices with a network address.

To create an access point, we'll need DNSMasq and HostAPD. Install all the required software in one go with this command:

```
sudo apt install dnsmasq hostapd
```

Since the configuration files are not ready yet, turn the new software off as follows:

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

Configuring a static IP

We are configuring a standalone network to act as a server, so the Raspberry Pi needs to have a static IP address assigned to the wireless port. This documentation assumes that we are using the standard 192.168.x.x IP addresses for our wireless network, so we will assign the server the IP address 192.168.4.1. It is also assumed that the wireless device being used is `wlan0`.

To configure the static IP address, edit the dhcpcd configuration file with:

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

Go to the end of the file and edit it so that it looks like the following:

```
interface wlan0
    static ip_address=192.168.4.1/24
    nohook wpa_supplicant
```

Now restart the dhcpcd daemon and set up the new `wlan0` configuration:

```
sudo service dhcpcd restart
```

Configuring the DHCP server (dnsmasq)

The DHCP service is provided by dnsmasq. By default, the configuration file contains a lot of information that is not needed, and it is easier to start from scratch. Rename this configuration file, and edit a new one:

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

Type or copy the following information into the dnsmasq configuration file and save it:

```
interface=wlan0      # Use the require wireless interface - usually wlan0
dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h
```

So for `wlan0`, we are going to provide IP addresses between 192.168.4.2 and 192.168.4.20, with a lease time of 24 hours. If you are providing DHCP services for other network devices (e.g. eth0), you could add more sections with the appropriate interface header, with the range of addresses you intend to provide to that interface.

There are many more options for dnsmasq; see the dnsmasq documentation for more details.

Start `dnsmasq` (it was stopped), it will now use the updated configuration:

```
sudo systemctl start dnsmasq
```

Configuring the access point host software (hostapd)

You need to edit the hostapd configuration file, located at /etc/hostapd /hostapd.conf, to add the various parameters for your wireless network. After initial install, this will be a new/empty file.

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

Add the information below to the configuration file. This configuration assumes we are using channel 7, with a network name of NameOfNetwork, and a password AardvarkBadgerHedgehog. Note that the name and password should **not** have quotes around them. The passphrase should be between 8 and 64 characters in length.

To use the 5 GHz band, you can change the operations mode from hw_mode=g to hw_mode=a. Possible values for hw_mode are:

- a = IEEE 802.11a (5 GHz)

- b = IEEE 802.11b (2.4 GHz)
- g = IEEE 802.11g (2.4 GHz)
- ad = IEEE 802.11ad (60 GHz) (Not available on the Raspberry Pi)

```
interface=wlan0
driver=nl80211
ssid=NameOfNetwork
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=AardvarkBadgerHedgehog
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

We now need to tell the system where to find this configuration file.

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

Find the line with #DAEMON_CONF, and replace it with this:

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

Start it up

Now enable and start `hostapd`:

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl start hostapd
```

Do a quick check of their status to ensure they are active and running:

```
sudo systemctl status hostapd
sudo systemctl status dnsmasq
```

Add routing and masquerade

Edit /etc/sysctl.conf and uncomment this line:

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

Add a masquerade for outbound traffic on eth0:

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

Save the iptables rule.

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

Edit /etc/rc.local and add this just above "exit 0" to install these rules on boot.

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

Reboot and ensure it still functions.

Using a wireless device, search for networks. The network SSID you specified in the hostapd configuration should now be present, and it should be accessible with the specified password.

If SSH is enabled on the Raspberry Pi access point, it should be possible to connect to it from another Linux box (or a system with SSH connectivity present) as follows, assuming the `pi` account is present:

```
ssh pi@192.168.4.1
```

By this point, the Raspberry Pi is acting as an access point, and other devices can associate with it. Associated devices can access the Raspberry Pi access point via its IP address for operations such as `rsync`, `scp`, or `ssh`.

Using the Raspberry Pi as an access point to share an internet connection (bridge)

One common use of the Raspberry Pi as an access point is to provide wireless connections to a wired Ethernet connection, so that anyone logged into the access point can access the internet, providing of course that the wired Ethernet on the Pi can connect to the internet via some sort of router.

To do this, a 'bridge' needs to put in place between the wireless device and the Ethernet device on the access point Raspberry Pi. This bridge will pass all traffic between the two interfaces. Install the following packages to enable the access point setup and bridging.

```
sudo apt install hostapd bridge-utils
```

Since the configuration files are not ready yet, turn the new software off as follows:

```
sudo systemctl stop hostapd
```

Bridging creates a higher-level construct over the two ports being bridged. It is the bridge that is the network device, so we need to stop the `eth0` and `wlan0` ports being allocated IP addresses by the DHCP client on the Raspberry Pi.

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

Add `denyinterfaces wlan0` and `denyinterfaces eth0` to the end of the file (but above any other added `interface` lines) and save the file.

Add a new bridge, which in this case is called `br0`.

```
sudo brctl addbr br0
```

Connect the network ports. In this case, connect `eth0` to the bridge `br0`.

```
sudo brctl addif br0 eth0
```

Now the interfaces file needs to be edited to adjust the various devices to work with bridging. To make this work with the newer systemd configuration options, you'll need to create a set of network configuration files.

If you want to create a Linux bridge (br0) and add a physical interface (eth0) to the bridge, create the following configuration.

```
sudo nano /etc/systemd/network/bridge-br0.netdev

[NetDev]
Name=br0
Kind=bridge
```

Then configure the bridge interface br0 and the slave interface eth0 using .network files as follows:

```
sudo nano /etc/systemd/network/bridge-br0-slave.network

[Match]
Name=eth0

[Network]
Bridge=br0
```

```
sudo nano /etc/systemd/network/bridge-br0.network

[Match]
Name=br0

[Network]
Address=192.168.10.100/24
Gateway=192.168.10.1
DNS=8.8.8.8
```

Finally, restart systemd-networkd:

```
sudo systemctl restart systemd-networkd
```

You can also use the brctl tool to verify that a bridge br0 has been created.

The access point setup is almost the same as that shown in the previous section. Follow the instructions above to set up the `hostapd.conf` file, but add `bridge=br0` below the `interface=wlan0` line, and remove or comment out the driver line. The passphrase must be between 8 and 64 characters long.

To use the 5 GHz band, you can change the operations mode from 'hw_mode=g' to 'hw_mode=a'. The possible values for hw_mode are:

- a = IEEE 802.11a (5 GHz)
- b = IEEE 802.11b (2.4 GHz)
- g = IEEE 802.11g (2.4 GHz)
- ad = IEEE 802.11ad (60 GHz) (Not available on the Raspberry Pi)

```
interface=wlan0
bridge=br0
#driver=nl80211
ssid=NameOfNetwork
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=AardvarkBadgerHedgehog
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Now reboot the Raspberry Pi.

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl start hostapd
```

There should now be a functioning bridge between the wireless LAN and the Ethernet connection on the Raspberry Pi, and any device associated with the Raspberry Pi access point will act as if it is connected to the access point's wired Ethernet.

The bridge will have been allocated an IP address via the wired Ethernet's DHCP server. Do a quick check of the network interfaces configuration via:

```
ip addr
```

The `wlan0` and `eth0` no longer have IP addresses, as they are now controlled by the bridge. It is possible to use a static IP address for the bridge if required, but generally, if the Raspberry Pi access point is connected to an ADSL router, the DHCP address will be fine.

VIE

READ OUR USAGE

ABOUT US

About us
Our team
Governance
Safeguarding
Our supporters
Jobs
Contact us

SUPPORT

Help
Documentation
Projects
Training
Downloads
Research
FAQ

SIGN UP TO OUR NEWSLETTER

Your email here

SUBSCRIBE

RASPBERRY PI FOUNDATION
UK REGISTERED CHARITY 1129409

Privacy Cookies Trademark rules and brand guidelines