

Raspberry Pi Stack Exchange is a question and answer site for users and developers of hardware and software for Raspberry Pi. It only takes a minute to sign up.

Anybody can ask a question



Anybody can answer

Sign up to join this community

The best answers are voted up and rise to the top



Access point as WiFi router/repeater, optional with bridge

Asked 4 years, 8 months ago Modified 2 years, 3 months ago Viewed 70k times



35

I need a wireless access point that is bridged with the wired ethernet port and that is connected to another wifi internet router with a client connection as uplink. All devices connected to the RasPi either by wifi or by wired ethernet should have access to the internet and should be able to connect to each other.



How can I achieve this with a Raspberry Pi?



wifi ethernet access-point systemd-networkd



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

43

asked Oct 9, 2018 at 22:50



Ingo

41.5k

15

82

192

Sorted by:

Highest score (default)





We are talking about an access point together with a wireless client connection on interface **wlan0** to another wifi network (wlan repeater) or an internet router.

53



If you want an access point together with an ethernet interface **eth0** for a wired uplink to a switch or an internet router then look at [Setting up a Raspberry Pi as an access point - the easy way](#).



Tested on a Raspberry Pi 4B with

[Raspberry Pi OS \(32-bit\) with desktop 2020-08-20](#) updated on 2020-09-01.

[Raspberry Pi OS \(32-bit\) with desktop 2020-05-27](#) updated on 2020-06-11.

[Raspbian Buster Lite 2020-02-13](#) updated on 2020-03-03.

Updates done with `sudo apt update && sudo apt full-upgrade && sudo reboot`.

Here you will find the [last tested revision for Raspbian Stretch Lite](#).

IMPORTANT UPDATE on 2020-09-02:

I have found that the image versions of the **Raspberry Pi OS** come with an unstable firmware for the built-in WiFi device. For further information look at my bug report at [WiFi firmware brcmfmac crashes when using virtual AP interface](#). The last stable is [firmware-brcm80211_20190114-1+rpt4_all.deb](#) coming with image **Raspbian Buster Lite 2020-02-13**.

So we have to downgrade the WiFi firmware to this stable version as long as the bug isn't fixed. After flashing the *Raspberry Pi OS* image with update/full-upgrade, follow this to do downgrade:

```
root@raspberrypi:~# wget http://archive.raspberrypi.org/debian/pool/main/f/firmware-nonfree
```

```
firmware-brcm80211_20190114-1+rpt4_all.deb
```



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

and simultaneously connect to another (as a general limitation. The client connection to the wireless distribution system) that is not possible. For further information look at [Raspberry Pi WiFi to use as a client](#) for the wifi client connection to the internet. The ethernet port can be bridged so all devices in the same domain on its common subnet.

For the Raspberry Pi OS net. All commands can simply be executed. For the blocks with EOF you can select

from `cat <incl the last EOF` and paste it to the RasPi's command line. With *Enter* it will be


```
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
EOF
```

```
rpi ~# chmod 600 /etc/hostapd/hostapd.conf
```

Create a service for the accesspoint with hostapd:

```
rpi ~# systemctl edit --force --full accesspoint@.service
```

In the empty editor insert these statements, save them and quit the editor:

```
[Unit]
Description=accesspoint with hostapd (interface-specific version)
Wants=wpa_supplicant@%i.service

[Service]
ExecStartPre=/sbin/iw dev %i interface add ap@%i type __ap
ExecStart=/usr/sbin/hostapd -i ap@%i /etc/hostapd/hostapd.conf
ExecStopPost=-/sbin/iw dev ap@%i del

[Install]
WantedBy=sys-subsystem-net-devices-%i.device
```

Enable the service:

```
rpi ~# systemctl enable accesspoint@wlan0.service
rpi ~# rfkill unblock wlan
```



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

Connection

and psk= and enable it:

```
wlan0.conf <<EOF
```

```
netdev
```

```
it-wlan0.conf
```

Extend wpa_supplicant with:

```
rpi ~# systemctl edit wpa_supplicant@wlan0.service
```

In the empty editor insert these statements. Save it and quit the editor:

```
[Unit]
BindsTo=accesspoint@%i.service
After=accesspoint@%i.service
```

Step 4: setup static interfaces

Create these files:

```
rpi ~# cat > /etc/systemd/network/08-wifi.network <<EOF
[Match]
Name=w1*
[Network]
LLMNR=no
MulticastDNS=yes
# If you need a static ip address, then toggle commenting next four lines (example)
DHCP=yes
#Address=192.168.50.60/24
#Gateway=192.168.50.1
#DNS=84.200.69.80 1.1.1.1
EOF
```

For those who want a bridge, do not need the next file and must continue with **Setup bridge**.

```
rpi ~# cat > /etc/systemd/network/12-ap.network <<EOF
```



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

st

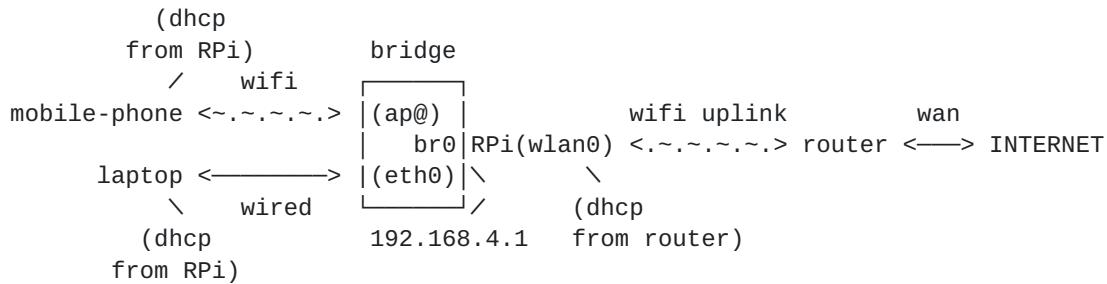
Accept all cookies

Necessary cookies only

Customize settings

Example for this setup:

Example for this setup.



Step 5: configure hostapd for a bridge

Add a line `bridge=br0` to `/etc/hostapd/hostapd.conf` with:

```
rpi ~# echo 'bridge=br0' >> /etc/hostapd/hostapd.conf
```

Step 6: setup wpa_supplicant for a bridge

Extend `wpa_supplicant` with:

```
rpi ~# systemctl edit wpa_supplicant@wlan0.service
```

In the editor append lines so it looks like the following. Have attention to the minus sign after equal `=-` on some statements. Save it and quit the editor:

```
[Unit]
```



```
online --interface=%i --timeout=60
```

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

`i up` is not a typo. On stopping `gain`.

these files:

```
:EOF
```

```
Name=br0
```

```
Kind=bridge
EOF
```

```
rpi ~# cat > /etc/systemd/network/04-eth0.network <<EOF
[Match]
Name=eth0
[Network]
Bridge=br0
EOF
```

```
rpi ~# cat > /etc/systemd/network/16-br0_up.network <<EOF
[Match]
Name=br0
[Network]
IPMasquerade=yes
Address=192.168.4.1/24
DHCPServer=yes
[DHCPServer]
DNS=84.200.69.80 1.1.1.1
EOF
```

Reboot.
That's it.

◆ Details

General

One problem is that we want to use the same device for a wifi client and for an access point. This must be supported by the wifi hardware. We can check this with

```
rpi ~$ sudo iw list | grep -A4 "valid interface combinations:"
```



```
<= 1, #{ P2P-client, P2P-GO } <=
: P2P-client } <= 1, #{ P2P-device
```

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

1, but you can also see that **this is**

wise, it won't work. If other

s point

```
ace ap@wlan0
```

```
ce wlan0
```

This order restricts the whole setup. It isn't possible to start *hostapd* when *wpa_supplicant* is running. You must first stop *wpa_supplicant* and then start *hostapd* and *wpa_supplicant* in this order. This is due to the WiFi driver `brcmfmac`.

Another quirk is that *wpa_supplicant* disables the interface **ap@wlan0** when it is managed, no matter if it starts or stops. So even after a start from *wpa_supplicant*, we have to enable **ap@wlan0** again.

Here are mainly four components involved: virtual interface **ap@wlan0**, *hostapd*, *wpa_supplicant*, and setup the bridge.

hostapd (Step 2)

This is the normal setup you will find all over the web. Please set the channel to the same value then **wlan0** is connected to the wifi from your router. It is the restriction from the hardware and may avoid confusion with channel settings. *hostapd* will always set the channel to the same value than from the client connection, no matter what you set here.

Please use a passphrase that is long enough, I would say at least with 8 characters. It was told that 4 characters are too small and *hostapd* will refuse to accept connections. You will not find the access point **RPiNet** on your mobile phone (thanks to [@Leo](#)).

We do not need to unmask the `hostapd.service` as described at `/usr/share/doc/hostapd/README.Debian` because we create our own `accesspoint@.service` for *hostapd*.

In this service the interface **ap@wlan0** is created before starting *hostapd*. This is respected to the starting order. Because **ap@wlan0** is bound to **wlan0** it gets the same mac address. You may consider to give it a unique mac address but that's not a good idea. On the network is



wlan0 its own mac address then it is less for connections to **ap@wlan0** is promiscuous mode to get it to work

so it can be created again on the next

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

[Customize settings](#)

You may consider to disable p2p with `no-p2p` in `hostapd-wlan0.conf` to avoid to get the interface not do that in this configuration with a *wpa_supplicant* and it will immediately connect to it. It endless connect and

Its extended unit is to strictly bind `wpa_supplicant` to `hostapd` so it will start and stop with it in the right order. We do it because running `wpa_supplicant` without `hostapd` is very complicated and may result in a total crash of the `brcmfmac` wifi driver, I have seen that on my tests.

Setup static interfaces (Step 4)

Here we define interface `wlan0` and the virtual interface `ap@wlan0` we need for a wifi repeater. For `wlan0` we get an IP address by DHCP from the internet router. Optional you can set a static IP address, but then you also have to define all other settings like gateway and DNS server because you don't get them anymore from the DHCP server.

Interface `ap@wlan0` needs a static IP address because it is the access point and the source of the DHCP server for the wifi network. There is no need that clients, connected to the internet router, must also connect to clients on the access point. So we can use NAT (`IPMasquerade=yes`) that simplifies setup a lot. Then we don't have to configure static routes and it is not needed to configure `wlan0` with static values (IP address, gateway, DNS server, etc.).

hostapd for bridge (Step 5)

We have setup `hostapd` in Step 2. Now we only need to append `bridge=br0` to its configuration so that it will dynamically manage to add `ap@wlan0` to the bridge.

wpa_supplicant for bridge (Step 6)

When `wpa_supplicant` is started it also brings up interface `wlan0`. Every time it is managed, it will bring up `wlan0` because it is not a real interface and



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

rier" and the link becomes ready (3).
`wpa_supplicant` has started. There is a helper
synchronous. We use it with a timeout
than 60 seconds to get up, then you
`etl -b -e` you can check if starting
ing up wifi interface `ap@wlan0`.
because otherwise, the bridge will not
he helpers options with `/lib/systemd`

`wlan0` must be brought up again so it

define static interfaces for bridge (Step 7)

These files define the bridge. **ap@wlan0** will dynamically be added by *hostapd*. **eth0** and **ap@wlan0** are slaves of the bridge and don't need an IP address. The bridge **br0** itself also does not need an IP address but here it has a static IP address because we need it for the DHCP server bound to **br0** so the DHCP server is available on the subnet of the bridge (interfaces eth0, ap@wlan0).

Conclusion

Update 2020-07-22:

[Bassline Soup](#) has made some [interesting tests](#) that may help to troubleshoot the setup.

You are now able to start, stop and restart *hostapd* and *wpa_supplicant*.

You should see the access point, here *RPiNet*, in your mobile phone and should get an IP address to connect to the access point. You should also get an IP address from the same subnet if you connect a laptop with a cable to the ethernet port and you can ping the mobile phone.

With this setup you have the following dependencies with starting and stopping:

hostapd	wpa_supplicant	wpa_supplicant	hostapd
starting	starts	starting	starts
stopping	stops	stopping	----
restarting	restarts	restarting	starts

This is because of the strict order of starting (see section *General*). You cannot run *hostapd* without *wpa_supplicant* by starting



... (or something else) it is mostly Lan0 up .

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

[Customize settings](#)

15

answered Oct 9, 2018 at 22:50



Ingo

41.5k

15

82

192



10

This is in addition to the [Ingo's outstanding description](#). Here we would like to use the default networking system of Raspbian (assuming to have previously configured it) rather than replacing it with [systemd-networkd](#).



The standard networking mode to configure TCP/IP in Raspbian/Raspberry Pi OS is mentioned in the [related documentation](#). At the time of writing, [dhcpcd is used](#) (*dhcpcd5* package). This mode replaced the former [Debian Networking](#) (*ifupdown* package), where the configuration was done in [/etc/network/interfaces](#). The [documentation](#) also describes [how to setup a Wi-Fi client/Wi-Fi station](#) via [wpa_supplicant](#) by configuring [/etc/wpa_supplicant/wpa_supplicant.conf](#) to connect the Raspberry Pi to an existing wireless LAN on its default *wlan0* interface; alternatively, it describes how to set up a [routed wireless access point](#) on the same *wlan0* interface (instead of the Wi-Fi station). We will use [this last document](#) as a reference, changing it in just a few points to configure the same wireless board to enable a *wlan0* interface (Wi-Fi Client/Station, connected to another wifi network or internet router) as well as a Wi-Fi Access Point named *uap0* (e.g., to define a wlan repeater):

- we create a copy of the *hostapd* service configuration, adding *ExecStartPre* and *ExecStopPost* settings to add/remove the "type AP" virtual interface via *iw*; we prefer using this rather than adding a rule in [/etc/udev/rules.d](#);
- we use *dhcpcd* to set a static IP address to the created interface, also modifying the *wlan0* configuration, to act as a DHCP client (without static addressing)

As DNS forwarder and DHCP Server, we will keep [dnsmasq](#), used in the previously mentioned Raspberry configuration. We will also keep [DNS Multicast](#) (*libnss-mdns*) and Zeroconf/Apple Bonjour (*avahi-daemon*). We will not use *systemd-resolved*.



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

not used, meaning that [/etc/network](#) files are defined, apart from source files or statements commented out. The [/etc/network/interfaces](#) directory is left empty. We also assume *systemd-resolved* is active.

```

-----+ (192.168.1.1
         | / +DHCP server)
Link    | /
-.-> router <---> INTERNET
                        wan
1.2

```

(documentation):

```
sudo -Es
apt install -y dnsmasq hostapd
```

Configure *hostapd* for the Access Point:

```
test -f /etc/hostapd/hostapd.conf || cat > /etc/hostapd/hostapd.conf <<\EOF
ctrl_interface=/var/run/hostapd
driver=nl80211
country_code=IT # Use your country code
ssid=YourSessionIdentifier
hw_mode=g
channel=7
auth_algs=1
wpa=2
wpa_passphrase=verySecretPassword
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
EOF
```

```
chmod 600 /etc/hostapd/hostapd.conf
# edit /etc/hostapd/hostapd.conf and verify it
```

Test creation and deletion of *uap0* virtual device on *wlan0*:

```
/sbin/iw dev wlan0 interface add uap0 type __ap
/sbin/iw dev uap0 del
```

No error should be produced. If the first command fails with *Device or resource busy (-16)*, it means that the device driver does not allow creating additional "Type AP" interfaces, which are busy for other services (e.g., an *AP* or a *P2P-GO* group is already active and has to be



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

[Customize settings](#)

managed are returned, the device
ing an AP interface. Output of the

Use this command to check the allowed combination options:

```
iw list | grep "valid interface combinations" -A 8
```

Every line contains alternative combinations. With the Broadcom BCM2711 SoC included in a Raspberry Pi 4 B, I get the following:

```
valid interface combinations:
    * #{ managed } <= 1, #{ P2P-device } <= 1, #{ P2P-client, P2P-GO }
<= 1,
    total <= 3, #channels <= 2
    * #{ managed } <= 1, #{ AP } <= 1, #{ P2P-client } <= 1, #{ P2P-
device } <= 1,
    total <= 4, #channels <= 1
Device supports scan flush.
Device supports randomizing MAC-addr in sched scans.
Supported extended features:
    * [ 4WAY_HANDSHAKE_STA_PSK ]: 4-way handshake with PSK in station
mode
```

It means that not more than one AP or P2P-GO interface can be configured at the same time.

If the test to create and delete the *uap0* virtual device succeeds, you can go on, starting the *uap0* AP virtual interface on *wlan0*:

```
SYSTEMD_EDITOR=tee systemctl edit --force --full uap@.service <<\EOF
[Unit]
Description=IEEE 802.11 %p%i AP on wlan%i with hostapd
After=network.target

[Service]
```



```
ti type __ap
d.pid -B $DAEMON_OPTS
```

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

```
ervice was active before
l service was enabled before
```

re, with:

```
sudo -Es
#export SYSTEMD_EDITOR=vi # uncomment this if you like "vi", otherwise "edit"
will be used
export SYSTEMD_EDITOR=vi systemctl edit --full uap@.service
exit
```

Some configurations (especially related to the old *ifupdown* mode) consider adding a static rule in `/etc/udev/rules.d` to bring up the `uap0` AP interface, like `ACTION=="add", SUBSYSTEM=="ieee80211", KERNEL=="phy0", RUN+="/sbin/iw phy %k interface add uap0 type __ap"`; as we are parametrizing this with the `ExecStartPre` and `ExecStopPost` statements of the `uap@` service definition, it is important that your configuration does not include such rule in `/etc/udev/rules.d`.

Now we setup `wpa_supplicant` for client connections.

We need to update `dhcpcd` (the default a DHCP client) by editing `/etc/dhcpcd.conf`, adding the following to the end (these settings replace the configuration suggested in Raspberry's site, which defined a static address to `wlan0` instead of a DHCP Client; ref. "Define the wireless interface IP configuration"):

```
interface wlan0 # these two lines are not strictly needed, as wlan0 uses the
default configuration
    dhcp

interface uap0 # this defines static addressing to uap0 and disables wpa_supplicant
for this interface
    static ip_address=192.168.50.1/24
    ipv4only
    nohook wpa_supplicant
```



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

So that, other than removing the static `interface wlan0`, so that

[/etc/dhcpcd-hooks directory](#): one of this `hook` files is discovered by `dhcpcd`.

the default one: `/etc/wpa_supplicant` is configured and working; otherwise you

```
cat >/etc/wpa_supplicant
```

```
netdev
```

```
network={
    ssid="TestNet"
    psk="reallyNotMyPassword"
    key_mgmt=WPA-PSK
}
EOF
# edit /etc/wpa_supplicant/wpa_supplicant.conf and verify it
```

```
chmod 600 /etc/wpa_supplicant/wpa_supplicant.conf
```

```
systemctl disable wpa_supplicant.service # not used, as the agent is hooked by dhcpcd
```

We can then follow Raspberry’s documentation to enable routing and IP masquerading:

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

Create a file using the following command, with the contents below:

```
test -f /etc/sysctl.d/routed-ap.conf || cat >/etc/sysctl.d/routed-ap.conf <<\EOF
# https://www.raspberrypi.org/documentation/configuration/wireless/access-point-
routed.md
# Enable IPv4 routing
net.ipv4.ip_forward=1
EOF
# edit /etc/sysctl.d/routed-ap.conf and verify it
```

Updating the firewall rules:

```
# Add firewall rules
```



```
iptables -A QUERADE
iptables -A QUERADE
iptables -A --state RELATED,ESTABLISHED
iptables
```

Your privacy

By clicking “Accept all cookies”, you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

```
cat /etc/resolv.conf for nameservers
```

Accept all cookies

Necessary cookies only

Customize settings

```
dhcp-option=3, 192.168.33.1
EOF
```

```
LS:
```

```
# edit /etc/dnsmasq.conf and verify it
```

The configuration is finished. You must reboot before testing.

```
reboot
```

The `systemctl` command on `uap@0.service` allows disabling, enabling and monitoring the AP function; for instance, to stop AP, to remove the virtual interface and to free up device driver resources, you can issue:

```
sudo systemctl stop uap@0
```

To start again the service:

```
sudo systemctl stop dhcpcd && sudo systemctl start uap@0 && sudo systemctl start dhcpcd
```

Notice that in both cases `wlan0` link goes down for some seconds and then goes up again; an active TCP connection on `wlan0` is not lost anyway.

Temporarily stopping `dhcpcd` before starting up `uap0` is needed to also stop `wpa_supplicant` (hooked by `dhcpcd`); this is in order to avoid the Broadcom driver issue "failed to enable fw supplicant" (for some weird reason the device driver attempts to reload an already loaded firmware supplicant for this interface).



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

function of the Raspberry Pi (e.g., a place `wlan0` with `wlan1`, `uap@0` with `uap@1` and `uap0` are mentioned.

For example (e.g., referring to `uap0`), change it

[Raspberry Pi as a bridged wireless](#)

the Raspberry to the upstream router (e.g., a DHCP server). In such configuration, the interface, but not by its *Ethernet* interface. It is not assigned an IP address; also, it must not be disconnected from the upstream router to bridge the Wi-Fi client connections

Testing wpa_supplicant permissions

Testing wpa_supplicant permissions

wpa_supplicant is interfaced to *wpa_cli* through UNIX sockets and when logging into the system with *pi* user, you should be able to successfully control *wpa_supplicant* and related configuration.

The following steps allow verifying *wpa_supplicant* permissions.

Check who starts *wpa_supplicant* and that you do not have more services activating this daemon. Assuming that the tested interface is *wlan0*, if you rely on *dhcpcd* you should be able to verify that *wpa_supplicant* does not start after temporarily adding `denyinterfaces wlan0` as the first line of */etc/dhcpcd.conf* (`sudo systemctl restart dhcpcd`).

After restoring */etc/dhcpcd.conf* and restarting the service, you should get something like the following with `ps -ef | grep wpa_supplicant`:

```
root      613      1  0 Feb20 ?          01:50:03 wpa_supplicant -B -c/etc
/wpa_supplicant/wpa_supplicant.conf -iwlan0 -Dnl80211,wext
```

Notice the *root* user, its dependency to the parent process id *1* and the command line arguments (`-B` means daemon, `-c` with the appropriate configuration file, `-i` with the related interface, `-D` with standard drivers).

Specifically, you need to verify the presence of `-i` command-line option if *wpa_supplicant.conf* declares `GROUP=netdev`.

Check also `journalctl -t dhcpcd-run-hooks`; you should get `dhcpcd-run-hooks[7548]: wlan0: starting wpa_supplicant`.



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

onf you should get the following:

_supplicant/wpa_supplicant.conf

is should verify

netdev

up.

get the following:

1/wpa_supplicant

Notice the permissions, the owner which should be *root* and the group which should be *netdev*, thanks to *GROUP=netdev* in *wpa_supplicant.conf*.

If you do `ls -l /var/run/wpa_supplicant`, you should get something like

```
total 0
srwxrwx--- 1 root netdev 0 Feb 20 11:00 p2p-dev-wlan0
srwxrwx--- 1 root netdev 0 Feb 20 11:00 wlan0
```

Notice the presence of the `wlan0` Unix socket (`srwxrwx---`), as well as the P2P Device (*p2p-dev-wlan0*).

Notice the owner, that shall be *root*, and the group, which is again *netdev*.

If you do `iw dev` with *wpa_supplicant* active, one of the returned item should be:

```
Unnamed/non-netdev interface
wdev 0x2
addr ee:11:6c:59:a3:d4
type P2P-device
txpower 31.00 dBm
```

which refers to the P2P device (`type P2P-device`).

Now edit `/etc/passwd`, check the presence of *root* users and *pi* users:

```
root:x:0:0:root:/root:/bin/bash
```

```
...
```



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

including *netdev*.

user, do `id`:

`ido`), 29(audio), 44(video), 46(plugdev)

Verify that you can successfully run `wpa_cli`, without a need of `sudo`:

```
wpa_cli -i wlan0 ping
wpa_cli -i p2p-dev-wlan0 ping
```

You should get *PONG* twice.

Run `wpa_cli -i p2p-dev-wlan0 reconfigure`; you should get *OK*.

Share Improve this answer

edited Oct 7, 2021 at 7:34

answered Dec 3, 2020 at 12:46

Follow



Community Bot

1



ircama

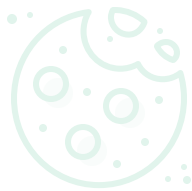
168 1 7

Phew. This was a bit over my head but a lifesaver. Thank you! I had to add the down and restart systemctl commands to rc.local to get wlan0 working. Any idea why? I'd previously done things on Buster Lite with github.com/lukicdarkoo/rpi-wifi and it appeared to have the same general problem. pulling the plug would get things to boot in the right order but no soft command seems to init properly. – limeandcoconut Jan 31, 2021 at 21:18

The rc.local statement you mention to get wlan0 working should not be needed. I recommend starting with a fresh install, then updating the OS to the latest version, then applying the network configuration according to the Raspberry documentation: raspberrypi.org/documentation/configuration/wireless/README.md and [raspberrypi.org/documentation/configuration/wireless/...](https://raspberrypi.org/documentation/configuration/wireless/). After checking that it works, you can revise the configuration basing on the notes in this topic. – ircama Feb 3, 2021 at 8:14

Also one thing if i now change my wpa_supplicant.conf i have to call systemctl restart dhcpcd.service instead of wpa_cli -i wlan0 reconfigure . Is there a better solution!? – delijati Feb 16, 2021 at 19:56

systemctl restart dhcpcd.service is a way to kill wpa_supplicant and restart it, when wpa_supplicant is run as a hook of dhcpcd. wpa_cli -i wlan0 reconfigure asks the currently running wpa_supplicant to read again the configuration file. Both should be viable alternatives.



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

... reconfigure to work as expected i
plicant daemon and restart it. After
i wlan0 ... reconfigure the changes
connecting. I'm not sure what the problem
t that worked :/ ... And yes i used -p
ecified in wpa_supplicant.conf .



8



This answer is not thought for setup. For setup look at the other answer to this question starting with **Setup wifi repeater**. This is mainly for troubleshooting and to show how it works with detailed commands step by step and checkpoints but without background information. I suppose you **have a monitor, keyboard and mouse attached** to your Raspberry Pi and **have a DHCP server running** on your local wireless network. Ip addresses and other settings in this check up are examples. You have to use yours. You can always look at `journalctl -b -e` for logged information.

Download ZIP image [Raspbian Stretch Lite 2018-11-13](#) to your linux pc.

Checkpoint 1: Compare the checksum with that on the download site.

```
pc ~$ sha256sum 2018-11-13-raspbian-stretch-lite.zip
```

```
47ef1b2501d0e5002675a50b6868074e693f78829822eef64f3878487953234d 2018-11-13-raspbian-stretch-lite.zip
```

Next step: Burn the image to an attached SD Card. I assume it is attached to `/dev/sdb`. Look with `lsblk` to what device file your SD Card is attached.

```
pc ~$ unzip -p 2018-11-13-raspbian-stretch-lite.zip | sudo dd of=/dev/sdb bs=4M conv=fsync
```

Checkpoint 2: check partitions on the SD Card:

```
pc ~$ sudo parted /dev/sdb print
```



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

```
:em  Flags
     lba
```

applicant.conf file:

untry= , ssid= and psk= , save it and

```

country=DE
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="your wifi network"
    psk="YourSecretPassword"
}

pc ~$ sudo umount boot/

```

Put the SD Card into your RasPi and boot.

Checkpoint 3: After login, check wifi connection and internet access. The **wlan0** interface must have an ip address:

```

rpi ~$ ip -4 addr show dev wlan0
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    inet 192.168.10.112/24 brd 192.168.10.255 scope global wlan0
        valid_lft forever preferred_lft forever

rpi ~$ ping -I wlan0 -c3 google.com
PING google.com (216.58.210.14) from 192.168.10.112 wlan0: 56(84) bytes of data.
64 bytes from fra16s07-in-f14.1e100.net (216.58.210.14): icmp_seq=1 ttl=57 time=213
ms
64 bytes from fra16s07-in-f14.1e100.net (216.58.210.14): icmp_seq=2 ttl=57
time=14.2 ms
64 bytes from fra16s07-in-f14.1e100.net (216.58.210.14): icmp_seq=3 ttl=57
time=12.2 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
           0 ms

```



some trouble then there is a general [working/WiFi/static IP address?](#) and fix

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only



Customize settings

```

rpi ~# sudo mv /etc/network/interfaces /etc/network/interfaces~

```

```
rpi ~# sed -i '11 resolvconf=NO' /etc/resolvconf.conf

rpi ~# systemctl enable systemd-networkd.service
rpi ~# systemctl enable systemd-resolved.service
rpi ~# ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf
```

Create these files for interfaces **wlan0** and **ap0** with your settings:

```
rpi ~# cat > /etc/systemd/network/08-wlan0.network <<EOF
[Match]
Name=wlan0
[Network]
IPForward=yes
DHCP=yes
EOF

rpi ~# cat > /etc/systemd/network/12-ap0.network <<EOF
[Match]
Name=ap0
[Network]
Address=192.168.4.1/24
DHCP=yes
[DHCP]
DNS=84.200.69.80 84.200.70.40
EOF
```

Rename */etc/wpa_supplicant/wpa_supplicant.conf* so it matches the interface setting and reboot:

```
rpi ~# mv /etc/wpa_supplicant/wpa_supplicant.conf /etc/wpa_supplicant
wpa_supplicant-wlan0.conf
rpi ~# systemctl disable wpa_supplicant@wlan0.service
rpi ~# systemctl reboot
```



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

tion with *systemd-networkd*:

```
service
lce
emon (interface-specific version)
ant@.service; disabled; vendor
```

11:07:53 GMT; 24s ago

```
lce/wpa_supplicant@wlan0.service
pplicant/wpa_supplicant-
```

PA supplicant daemon (interface-

successfully initialized

wp-dev-wlan0: CTRL-EVENT-REGDOM-

```
Jan 29 11:07:50 raspberrypi wpa_supplicant[415]: wlan0: Trying to associate with
31:31:c4:c7:f2:74 (SSID='wlan@hoefft-online.de' freq
```

```
34:31:c4:c7:f2:74 (SSID= wlan@b0c1c-0111nc.de) freq
Jan 29 11:07:56 raspberrypi wpa_supplicant[415]: wlan0: Associated with
34:31:c4:c7:f2:74
Jan 29 11:07:56 raspberrypi wpa_supplicant[415]: wlan0: WPA: Key negotiation
completed with 34:31:c4:c7:f2:74 [PTK=CCMP GTK=CCMP]
Jan 29 11:07:56 raspberrypi wpa_supplicant[415]: wlan0: CTRL-EVENT-CONNECTED -
Connection to 34:31:c4:c7:f2:74 completed [id=0 id_st
Jan 29 11:07:56 raspberrypi wpa_supplicant[415]: p2p-dev-wlan0: CTRL-EVENT-REGDOM-
CHANGE init=COUNTRY_IE type=COUNTRY alpha2=DE
```

```
rpi ~$ ip -4 addr show dev wlan0
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    inet 192.168.10.112/24 brd 192.168.10.255 scope global wlan0
        valid_lft forever preferred_lft forever
```

```
rpi ~$ ping -I wlan0 -c3 google.com
PING google.com (172.217.21.238) from 192.168.10.112 wlan0: 56(84) bytes of data.
64 bytes from fra16s13-in-f238.1e100.net (172.217.21.238): icmp_seq=1 ttl=56
time=14.1 ms
```

Checkpoint 5: IMPORTANT! At this point it must be able to set and delete the virtual interface **ap0** without error messages:

```
rpi ~$ sudo iw dev wlan0 interface add ap0 type __ap
rpi ~$ sudo iw dev ap0 info
Interface ap0
    ifindex 5
    wdev 0x5
    addr b8:27:eb:06:e8:8b
    type AP
    wiphy 0
    channel 1 (2412 MHz), width: 20 MHz, center1: 2412 MHz
    txpower 31.00 dBm

rpi ~$ sudo iw dev ap0 del
rpi ~$
```



Your privacy

By clicking “Accept all cookies”, you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

ssphrase= :

Accept all cookies

Necessary cookies only

Customize settings

auth_algs=1

```
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=verySecretPassword
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
EOF

rpi ~# chmod 600 /etc/hostapd/hostapd.conf
```

Set DAEMON_CONF="/etc/hostapd/hostapd.conf" in /etc/default/hostapd with:

```
rpi ~# sed -i 's/^#DAEMON_CONF=.*$/DAEMON_CONF="/etc/hostapd/hostapd.conf"/'
/etc/default/hostapd
```

In /etc/init.d/hostapd in section *INIT INFO* you have to comment the line # Should-Start: \$network to ## Should-Start: \$network :

```
rpi ~# sed -i 's/^\(# Should-Start:\s*$network\)$/#\1/' /etc/init.d/hostapd
```

Edit *hostapd.service* with `systemctl edit hostapd.service` and in the empty editor insert this, save and quit:

```
[Service]
ExecStartPre=/sbin/iw dev wlan0 interface add ap0 type __ap
ExecStopPost=-/sbin/iw dev ap0 del
```

Reboot.



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

with access point. It is very important
Otherwise the wifi driver crashes and

and **ap0** has state UP .

```
1500 qdisc pfifo_fast state DOWN
```

```
if:ff
```

```
) qdisc pfifo_fast state UP group
```

```
lobal ap0
```

Checkpoint II Check wifi on your mobile phone. You should find RPiNet and you can

connect to it. Your mobile phone cannot get into the internet because routing isn't configured yet.

Next step: Start `wpa_supplicant` after `hostapd`:

```
rpi ~$ sudo systemctl start wpa_supplicant@wlan0.service
```

Checkpoint 8: Interfaces `wlan0` and `ap0` must have state `UP` and you can ping `google.com` from the RasPi:

```
rpi ~$ ip -4 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    inet 192.168.10.112/24 brd 192.168.10.255 scope global dynamic wlan0
        valid_lft 25393sec preferred_lft 25393sec
4: ap0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    inet 192.168.4.1/24 brd 192.168.4.255 scope global ap0
        valid_lft forever preferred_lft forever
```

```
rpi ~$ ping -c3 google.com
PING google.com (172.217.21.238) 56(84) bytes of data.
64 bytes from fra16s13-in-f238.1e100.net (172.217.21.238): icmp_seq=1 ttl=56 time=13.7 ms
64 bytes from fra16s13-in-f238.1e100.net (172.217.21.238): icmp_seq=2 ttl=56 time=14.1 ms
64 bytes from fra16s13-in-f238.1e100.net (172.217.21.238): icmp_seq=3 ttl=56 time=15.9 ms
```



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

Start `wpa_supplicant` again:

```
time 2002ms
ms
```

so routed packages can return to
h:

```
ervice
ervice
```

ion to the minus sign after equal ==

```
; -o wlan0 -j MASQUERADE
; -o wlan0 -j MASQUERADE
```

Start wpa_supplicant again.

```
rpi ~$ sudo systemctl daemon-reload
rpi ~$ sudo systemctl start wpa_supplicant@wlan0.service
```

Checkpoint 9: Now you should be able to get into the internet with the mobile phone.

That's it.

If you run into trouble you can try a new attempt with:

```
rpi ~$ sudo systemctl disable wpa_supplicant@wlan0.service
rpi ~$ sudo systemctl disable hostapd.service
rpi ~$ sudo systemctl reboot
```

This is not a working installation. You have to edit `hostapd.service` and `wpa_supplicant@wlan0.service` as shown in my other answer.

Share Improve this answer

edited Feb 18, 2019 at 15:13

answered Jan 29, 2019 at 8:52

Follow




Ingo

41.5k

15

82

192

 **Highly active question.** Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

[Customize settings](#)