pi-networking-interfaces

Raspberry pi supports lan as well as wlan.

There are two network configuration possibilities.

dhcpcd (has been around for some time)
network-manager (relatively new)

We will focus on dhcpcd.

To list your network interfaces

```
pi@pi ~# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
mode DEFAULT group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
mode DEFAULT group default qlen 1000
link/ether e4:5f:01:fc:c6:be brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP mode DORMANT group default qlen 1000
link/ether e4:5f:01:fc:c6:bf brd ff:ff:ff:ff:ff:ff
```

To list your ip addresses

```
pi@pi ~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000
    link/ether e4:5f:01:fc:c6:be brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.64/24 brd 192.168.4.255 scope global dynamic eth0
       valid_lft 53765sec preferred_lft 53765sec

3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether e4:5f:01:fc:c6:bf brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.249/24 brd 192.168.4.255 scope global wlan0
       valid_lft forever preferred_lft forever
```

To list your routes:

```
pi@pi ~# ip route show
default via 192.168.4.1 dev eth0 src 192.168.4.159 metric 202
default via 192.168.4.1 dev wlan0 proto dhcp src 192.168.4.59 metric 303
10.2.25.0/24 dev tun0 proto kernel scope link src 10.2.25.1
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
192.168.4.0/24 dev eth0 proto dhcp scope link src 192.168.4.159 metric 202
192.168.4.0/24 dev wlan0 proto dhcp scope link src 192.168.4.59 metric 303
```

The config files:

lan: **/etc/dhcpcd.conf**
wlan: **/etc/wpa_supplicant/wpa_supplicant.conf**

To add a static ip address…

```
# Example static IP configuration:

interface eth0

static ip_address=192.168.4.158/24

static ip6_address=fd51:42f8:caae:d92e::ff/64

static routers=192.168.4.1

static domain_name_servers=192.168.4.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

To add a persistent static route, create or edit the following file

```
/lib/dhcpcd/dhcpcd-hooks/40-route
```

and use the following syntax

```
ip route add 192.168.100.0/24 via 192.168.0.2
```

To setup your wlan interface:

Edit the following config file: `/etc/wpa_supplicant/wpa_supplicant.conf`

```
update_config=1

country=BE

network={

       ssid="consilium"

       #psk="th1s!s@p@ss"

       key_mgmt=WPA-PSK

}

Or to set up without keymanagement:

update_config=1

country=BE

network={

       ssid="consilium"

       key_mgmt=NONE

}
```

Some more examples:

```
WPA-Personal (PSK) as home network and WPA-Enterprise with EAP-TLS as work network.
# allow frontend (e.g., wpa_cli) to be used by all users in 'wheel' group
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=wheel
#
# home network; allow all valid ciphers
network={
   ssid="home"
   scan_ssid=1
   key_mgmt=WPA-PSK
   psk="very secret passphrase"
}
=====
```

```
# work network; use EAP-TLS with WPA; allow only CCMP and TKIP ciphers
network={
   ssid="work"
   scan_ssid=1
   key_mgmt=WPA-EAP
   pairwise=CCMP TKIP
   group=CCMP TKIP
   eap=TLS
   identity="user@example.com"
   ca_cert="/etc/cert/ca.pem"
   client_cert="/etc/cert/user.pem"
   private_key="/etc/cert/user.prv"
   private_key_passwd="password"
}

======

WPA-RADIUS/EAP-PEAP/MSCHAPv2 with RADIUS servers that use old peaplabel (e.g., Funk
Odyssey and SBR, Meetinghouse Aegis, Interlink RAD-Series)
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=wheel
network={
   ssid="example"
   scan_ssid=1
   key_mgmt=WPA-EAP
   eap=PEAP
   identity="user@example.com"
   password="foobar"
   ca_cert="/etc/cert/ca.pem"
   phase1="peaplabel=0"
   phase2="auth=MSCHAPV2"
}
=====
EAP-TTLS/EAP-MD5-Challenge configuration with anonymous identity for the unencrypted
use. Real identity is sent only within an encrypted TLS tunnel.
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=wheel
network={
   ssid="example"
   scan_ssid=1
   key_mgmt=WPA-EAP
   eap=TTLS
   identity="user@example.com"
   anonymous_identity="anonymous@example.com"
   password="foobar"
   ca_cert="/etc/cert/ca.pem"
   phase2="auth=MD5"
}

=====

IEEE 802.1X (i.e., no WPA) with dynamic WEP keys (require both unicast and broadcast);
use EAP-TLS for authentication
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=wheel
network={
   ssid="1x-test"
```

```
   scan_ssid=1
   key_mgmt=IEEE8021X
   eap=TLS
   identity="user@example.com"
   ca_cert="/etc/cert/ca.pem"
   client_cert="/etc/cert/user.pem"
   private_key="/etc/cert/user.prv"
   private_key_passwd="password"
   eapol_flags=3
}

======

Catch all example that allows more or less all configuration modes. The configuration
options are used based on what security policy is used in the selected SSID. This is
mostly for testing and is not recommended for normal use.
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=wheel
network={
   ssid="example"
   scan_ssid=1
   key_mgmt=WPA-EAP WPA-PSK IEEE8021X NONE
   pairwise=CCMP TKIP
   group=CCMP TKIP WEP104 WEP40
   psk="very secret passphrase"
   eap=TTLS PEAP TLS
   identity="user@example.com"
   password="foobar"
   ca_cert="/etc/cert/ca.pem"
   client_cert="/etc/cert/user.pem"
   private_key="/etc/cert/user.prv"
   private_key_passwd="password"
   phase1="peaplabel=0"
   ca_cert2="/etc/cert/ca2.pem"
   client_cert2="/etc/cer/user.pem"
   private_key2="/etc/cer/user.prv"
   private_key2_passwd="password"
}

=====

Authentication for wired Ethernet. This can be used with wired or roboswitch interface
(-Dwired or -Droboswitch on command line).
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=wheel
ap_scan=0
network={
   key_mgmt=IEEE8021X
   eap=MD5
   identity="user"
   password="password"
   eapol_flags=0
}
```