"Sure fire wifi connection and troubleshooting techniques"

By Steve Litt
http://www.troubleshooters.com

Benefits

- Troubleshooting Transparency
- Easy to follow connection milestones
- Few dependencies
- Small toolset
- No window manager or X required
- Solid connection/roaming
- Scriptable

Why Bother?

- Other, more automatic wifi softwares work perfectly.
- Until they don't.
- Then you're trial and erroring a black box.
- You need accessible access points.

What it Is / How it's Built

- wpa_supplicant and dhcpcd run as daemons.
  - wpa_supplicant must always start first.
- wpa_cli, ip and init/process supervisor for control.
- wpa_cli, ip and init/process supervisor yield internal info.
- Once in a little while, iwlist for rundundancy.

System Requirements

- ip command stack (iproute2).
- wpa_supplicant
- wpa_cli
- dhcpcd
- Init system or process supervisor capable of respawning and settling process dependencies.
  - Runit, s6, Epoch, systemd, maybe sysvinit,
  - All others (if you add daemontools)

Vocabulary

- Access point = nearby Wifi transmitter
  - To which you might be able to connect.
  - Visible from wpa_cli scan; sleep 3; wpa_cli scan_results
  - Referenced by ssid

- Network = wpa_supplicant data record
  - Necessary info to connect to an access point.
  - Visible from wpa_cli list_networks.
  - Always referenced by an integer.

Some Software Names

- wpa_supplicant: Software daemon to log into WPA and open access points and deliver a carrier (link)
- wpa_cli: User interface to wpa_supplicant.
- dhcpcd: Software daemon to obtain a dhcp lease, IP address
- ip: Command for troubleshooting network connections

The New Device Names

- Example: wlp0s22f2u1u3
- Called “Predictable Network Interface Names”
- From Freedesktop.org
- No problem.
- To discover: ip link
- To script:
  - ip -o link | cut -d ' ' -f2 | grep ^w | tr -d ':'

Connection Milestones

- wpa_supplicant connectivity
- Radio reception
- Acquire carrier
- Acquire IP address
- Network Connectivity
- DNS

Milestone Detection

- wpa_supplicant connectivity
  - wpa_cli ping
- Radio reception
  - wpa_cli scan; sleep 3; wpa_cli scan_results
- Acquire carrier
  - No “NO CARRIER” in angle brackets
  - See also wpa_cli status
- Acquire IP address
  - ip addr or wpa_cli status
- Network Connectivity
  - ping 8.8.8.8
- DNS
  - ping google.com

Rules of the Road

- If an access point's strength is more negative than -70, don't even bother: It will be intermittent.
- Password must be at least 8 chars
- Sometimes you must disable wired Ethernet to access wifi.
- wpa_cli command names must be long enough to be unique.
- 168.254.0.0/16 (Link Local) is not valid

More Rules of the Road

- Most of this doable as normal user, but beyond the scope of this presentation.
- If interactive wpa_cli keeps throwing text do this:
  - select_network 0;disable_network 0
- Wifi dongle names change if plugged into different usb port
  - wpa_cli terminate
    - When this is likely, in order to restart wpa_supplicant
    - Exit and restart any interactive wpa_cli sessions

On Boot

- wpa_supplicant must run as respawnable daemon.
- dhcpcd must run as respawnable daemon.
  - Only after wpa_supplicant is operational

```
if wpa_cli ping; then
  exec dhcpcd -B ${OPTS:=-M} 1>&2
fi
sleep 1
exit 1
```

Idiomatic Connection Recipe
Top Level

- If already satisfactorily connected, you're done.
- wpa_ping
  - To test if wpa_supplicant is running.
- If you already have a satisfactory network, connect to it.
- Else make a new network out of a nearby access point.

Idiomatic Connection Recipe

Connecting to an Existing Network

- In this example it's network number 5.
- If ping fails, TROUBLESHOOT.

```
wpa_cli select 5
ip link set dev <devname> up
wpa_cli select_n 5
wpa_cli enable_n 5
ping 8.8.8.8
ping google.com
```

Idiomatic Connection Recipe

Make New Network From Nearby Access Point

- wpa_cli scan; sleep 3; wpa_cli scan_results
  - Pick one based on needed encryption and signal strength
- wpa_cli add_network
  - For this example assume it prints 8
- wpa_cli set_network 8 ssid '"cool_access_point"'
- wpa_cli set_network 8 psk '"my_secret"'
- wpa_cli select_network 8
- wpa_cli enable_network 8
- ip addr (to check for up, network, etc)
- ping google.com
- If it doesn't work, TROUBLESHOOT

Stages of Connection

- wpa_supplicant functional
- Radio functional
- Appropriate access point received
- “Link” or “Carrier”
- IP address
  - Link local address not valid: 169.254.0.0/16
- Internet connectivity via IP address
- Internet activity via URL

Troubleshooting

- ping 8.8.8.8
- wpa_cli status
- wpa_cli list_networks
  - To find out existing network numbers
- wpa_cli ping
- wpa_cli scan <dev>; sleep 3; wpa_cli scan_results
- wpa_cli add_network
  - To add a network

More Troubleshooting

- wpa_cli save_config
  - To peer into memory
  - To make network permanent
- wpa_cli dump
- wpa_cli reconfigure
- wpa_cli terminate
- wpa_cli disable_network <netno>
  - Stop infinite scan error loops

Still More Troubleshooting

- ip addr
  - “NO-CARRIER” in angle brackets?
  - ipv4 address listed?
- resolvfix.sh:
  - Back up first
  - Run as root

```
#!/bin/sh
echo nameserver 8.8.8.8 > /etc/resolv.conf
echo nameserver 8.8.4.4 >>
/etc/resolv.conf
```

- Shut down networkmanager and wicd
- dhcpcd -B
- wpa_supplicant

```
wpa_supplicant \
-c /etc/wpa_supplicant/wpa_supplicant.conf \
-i <devname> -d
```

Summary

- Wifi with few moving parts or dependencies
- Runs anywhere
- Transparent troubleshooting
- Requires both wpa_supplicant and dhcpcd to be daemons
  - Or explicitly run in foreground for troubleshooting