Using iw to Manage Wireless LAN in Linux

A wireless LAN (WiFi) interface is configured in a slightly different manner than a wired LAN (Ethernet) interface in Linux command line. Although `ifconfig` and `dhclient` can still be used to view/configure an IP address of the interface, other commands are needed to associate the wireless LAN device to an access point (AP). There are two main commands:

1. `iwconfig` can be used in a similar manner as `ifconfig`, allowing setting of various WiFi parameters.
2. `iw` is a newer command which is more powerful than `iwconfig`, but different syntax that `ifconfig`/`iwconfig`. (In fact there is an analogous command called `ip` which is meant to replace `ifconfig` for wired interfaces, but I do not use it much).

In this document I will show how to use `iw` to manage the WiFi in the Linux command line, with only one or two examples of `iwconfig`.

WiFi Interface Names

Ethernet interfaces are named `eth0`, `eth1`, ..., whereas WiFi interfaces are named `wlan0`, `wlan1`, and so on. But be careful that as WiFi devices are often less permanent than Ethernet devices (e.g. WiFi USB sticks can be inserted/removed on a regular basis) its more likely that the `wlan` interface number will be change. That is, don't always assume the interface is `wlan0`, even if there is just one WiFi device. You can check with `ifconfig`:

```
$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr f0:de:f1:61:04:b7
...
eth1      Link encap:Ethernet  HWaddr f0:de:f1:61:04:b8
...
eth2      Link encap:Ethernet  HWaddr f0:de:f1:61:04:b9
...
lo        Link encap:Local Loopback
...
wlan0     Link encap:Ethernet  HWaddr 8c:a9:82:b1:38:90
...
```

In fact with WiFi it is important to distinguish between the WiFi hardware (or PHYsical layer) and the WiFi interface (or MAC layer). Usually we set parameter values for the PHYsical layer, such as frequency or channel, and then associate the WiFi interface with an AP at the MAC layer. The command `iw` distinguishes between the hardware (also called *phy*) and the interface (also called *dev*, in the same manner as Ethernet).

To view the available WiFi hardware/interfaces:

```
$ iw dev
phy#0
        Interface wlan0
                ifindex 3
                type managed
```

This shows I have hardware called `phy0` and an interface called `wlan0`. Check the names for your computer and use them in the following instructions.

Associate with an AP

The process of connecting your wireless interface with an Access Point is called *association*. APs are given an Extended Service Set Identifier (ESSID) which identifies the network that they provide a connection to. For example, many APs inside SIIT have the ESSID of `wsiit`. Normally software will automatically show you the available ESSIDs within range of your interface, allowing you to click on the one you want to connect to. This triggers your interface to associate with an AP with that ESSID. On the lab computers we

have disabled this software, requiring us to manually connect to a ESSID. The following steps show how. Remember the examples use `wlan0` - you should replace with your interface name.

Make sure the wireless interface is up:

```
$ sudo ifconfig wlan0 up
```

Tell your interface to connect to a specified ESSID (e.g. `wsiit` or `NetlabBlueAC2`):

```
$ sudo iw dev wlan0 connect wsiit
```

You can check the link details:

```
$ iw dev wlan0 link
Connected to 00:19:e6:8d:55:64 (on wlan0)
        SSID: wsiit
        freq: 2437
        RX: 18444610 bytes (94857 packets)
        TX: 2554688 bytes (17365 packets)
        signal: -60 dBm
        tx bitrate: 54.0 MBit/s

        bss flags:      short-preamble short-slot-time
        dtim period:    0
        beacon int:     100
```

Another way to see the link information is using the old `iwconfig` command:

```
$ iwconfig wlan0
wlan0     IEEE 802.11bgn  ESSID:"wsiit"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 20:AA:4B:A3:63:39
          Bit Rate=54 Mb/s   Tx-Power=14 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=67/70  Signal level=-43 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:218   Missed beacon:0
```

Assuming the wireless AP is configured as a DHCP server, you can request an IP address (lternatively you can set your own IP address with `ifconfig`.):

```
$ sudo dhclient wlan0
```

You should now have IP connectivity. Test using ping or some other application

Disconnect from an AP

To disconnect from an AP run:

```
$ sudo iw dev wlan0 disconnect
```

Diagnosing Problems with AP Connectivity

If after you try to connect to a WiFi network but have problems with the above steps you should try to diagnose the problem using `ping` and `arp`. For example, ping the AP and from another computer ping your WiFi computer. Check that the ARP table contains entries. If it still doesn't work you can try again by:

```
$ sudo iw dev wlan0 disconnect
$sudo ifconfig wlan0 down
$sudo ifconfig wlan0 up
$sudo iw dev wlan0 connect wsiit
$sudo dhclient -r wlan0
$sudo dhclient wlan0
```

Capture Other Peoples Packets in Monitor Mode

The broadcast nature of wireless LAN transmissions means that WiFi hardware can receive any transmission on the same frequency from other devices nearby. However WiFi interfaces are normally configured to ignore frames for which your device is not the intended destination (e.g. if the destination address does not match yours or the broadcast address). Some devices and drivers support switching the WiFi interface to a special mode where it will not ignore packets that aren't destined to you. This is called *monitor mode*.

Putting a WiFi interface into monitor mode means you can capture packets that other devices are sending to an AP (and vice versa), again only if you are using the same frequency and are within range of the other devices. Of course, this is a potential security threat and it is probably illegal to intercept other peoples packets without their permission. Hence only perform these steps in the lab.

Monitor mode is only supported by some devices/drivers. In the lab, the TP-Link white USB adapters do not fully support monitor mode. However we have some other no-brand black USB adapters that do. Use those adapters for monitor mode.

To use monitor mode we will create a new interface for our hardware, and delete the normal (`wlan0`) interface:

```
$ sudo iw phy phy0 interface add mon0 type monitor
$ sudo iw dev wlan0 del
$ sudo ifconfig mon0 up
```

If you use `ifconfig`, `iwconfig` or `iw dev` now, you should see the `mon0` interface.

To capture other peoples traffic your monitor interface must use the same frequency as theirs. Once you know the frequency they are using (e.g. 2.437 GHz) you can set the frequency for the monitor interface:

```
$ sudo iw dev mon0 set freq 2437
```

If you want to see that you are in monitor mode and the frequency being used, a quick way to check is with `iwconfig`:

```
$ iwconffig mon0
```

Now you can capture with `tcpdump`:

```
$ sudo tcpdump -i mon0 -n -w wireless.cap
```

Disable Monitor Mode

You cannot use your WiFi device for both transmitting and monitoring at the same time. If you want to return to normal (*managed*) mode after performing monitoring then delete the monitoring interface down and add the normal interface

```
$ sudo iw dev mon0 del
$ sudo iw phy phy0 interface add wlan0 type managed
$ sudo ifconfig wlan0 up
```

Use `iw` and other commands to connect to a WiFi network again.