Linux Wireless

Go back –> iw documentation

mac80211 Multiple Virtual Interface (vif) Support

The mac80211 subsystem in the linux kernel supports multiple wireless interfaces to be created with one physical wireless card. This depends on the driver implementing this. This could allow you to join multiple networks at once, or connect to one network while routing traffic from an access point interface.

Notes on Terminology and Omissions

This document only covers the creation and configuration of the wireless side of the network. Once complete, the interfaces are configured the same as you would configure a wired interface on your system, and those details are documented very well, and in great depth by most distributions.

Syntax

```
iw dev <devname> interface add <name> type <type> [mesh_id <meshid>] [4addr on|off] [flags <flag>*]
```

or

```
iw phy <phyname> interface add <name> type <type> [mesh_id <meshid>] [4addr on|off] [flags <flag>*]
```

while the basic syntax to destroy an interface is:

```
iw dev <devname> del
```

- The <*devname*> and <*phyname*> fields represent the virtual interface (devname) and physical device (phyname). <*devname*> is often wlan0, wlan1, etc, while <*phyname*> is phy0, phy1, etc.
- The <*name*> field is where you choose what to call the new virtual interface (it's new devname).
- The <*type*> field is where you define what sort of interface you're creating. It is one of:
- * ibss - IBSS/ad-hoc
- * mp - Mesh Point
- * monitor - Monitor
- * ap - Access Point (special steps required)
- * station - Station/Client

Interface Types

IBSS/ad-hoc

An IBSS (Independent Basic Service Set) network, often called an ad-hoc network, is a way to have a group of devices talk to each other wirelessly, without a central controller. All devices talk directly to each other, with no inherent relaying.

All Devices can talk to each other:

```
  A
 / \
B---C
```

A can talk to B, B can talk to A and C, C can talk to B, but A and C can not talk directly (without setting up routes):

```
A---B---C
```

To create an ad-hoc network, you first create an ad-hoc interface (in this example named ah0):

```
iw phy <phyname> interface add <devname> type ibss
```

Next, you join/create the ibss:

```
iw dev <devname> ibss join <SSID> <freq in MHz> [fixed-freq] [<fixed bssid>] [key d:0:abcde]
```

example:

```
iw phy phy0 interface add ah0 type ibss
ifconfig ah0 up
iw dev ah0 ibss join AdHocNetworkName 2412
```

Repeat these steps on all devices you want to network together.

To leave an ibss you do:

```
iw dev <devname> ibss leave
```

example:

```
dev ah0 ibss leave
```

Mesh Point

A Mesh Point is a device in an 802.11s mesh network. Mesh networks work similarly to ad-hoc networks, with the added benefit of supporting routing between devices that can't talk directly to each other. All Devices can talk to each other:

```
  A
 / \
B---C
```

A can talk to B and C by relaying through B, B can talk to A and C directly, C can talk to B and A by relaying through B:

```
A---B---C
```

To create a mesh network, you first create a mesh interface(named mp0):

```
iw phy <phyname> interface add <devname> type mp mesh_id <meshid>
```

example:

```
iw phy phy0 interface add mp0 type mp mesh_id MeshNetwork
```

Unlike ad-hoc network, you do not need to actively join the Mesh. Once you've created a mesh point interface, and brought it up, it will automatically begin discovering the network topology, and relaying traffic as needed.

Repeat these steps on all devices you want to network together.

There are advanced commands you can use to control the routing over the mesh:

```
iw dev <devname> mpath dump

iw dev <devname> mpath set <destination MAC address> next_hop <next hop MAC address>

iw dev <devname> mpath new <destination MAC address> next_hop <next hop MAC address>

iw dev <devname> mpath del <MAC address>

iw dev <devname> mpath get <MAC address> del <MAC address>
```

TODO: learn to use those so I can document them

Monitor

Monitor mode is a passive-only mode, no packets are transmitted. All incoming packets are handed over to the host computer completely unfiltered. This mode is useful to see what's going on on the network.

To create a monitor interface you use the command:

```
phy <phyname> interface add <name> type <type> monitor [flags <flag>*]
```

Example:

```
iw phy phy0 interface add mon0 type monitor
```

Next you set the channel or frequency:

```
iw dev <devname> set channel <channel>
```

or

```
iw dev <devname> set freq <freq>
```

(in MHz)

Example:

```
iw dev mon0 set channel 7
```

or

```
iw dev mon0 set freq 2442
```

You can then use a utility like tcpdump, wireshark or tshark to view packets.

Access Point

Station/Client

---

en/users/documentation/iw/vif.txt · Last modified: 2015/01/26 09:49 (external edit)