

In Lab 5 we used python3 to scan the WiFi channels and print out information on any newly discovered networks by sniffing for Beacon frames.

For the homework assignment you will extend your sniffing to include the remaining management frames. Your task **[6 pt]**:

- **[0.5 pts]** Run your python script (it actually works i.e. no crashes and such). The current template should run (if the pcap files are in the correct location) and display every Beacon frame in the selected pcap.
- Instead of live sniffing, the data stored in pcap files will be used, therefore your script should open a pcap file [files available on Moodle – see contents of each file below]
  - This makes testing easier and eliminates the issue of trying to get the adapter to work on your device
  - `mgt_frames.pcap` contains *Association Response*, *Re-association Request*, *Re-association Response*, *Authentication* and *Deauthentication* frames
  - `test_pcap2.pcap` contains *Probe Request* and *Probe Response* frames
  - `test_pcap3.pcap` contains *Association Request*, *Association Response*, *Probe Request*, *Probe Response*, *Disassociate*, *Authentication* and *Action* frames
  - `test_pcap4.pcap` contains *Probe Request* and *Probe Response* frames
- **[1 pt]** Every **Probe Response** and **Probe Request** frame that is detected is displayed in the terminal.
  - For the Probe Response print the BSSID, SSID, Channel and Signal Strength
  - For the Probe Request print the SSID, Channel and Signal Strength
- **[1 pt]** Every **Authentication** and **Deauthentication** frame that is detected is displayed in the terminal
  - For both frames print the BSSID, Channel and Signal Strength
- **[1 pt]** Every **Association Request** and **Association Response** frame that is detected is displayed in the terminal
  - For the Association Request print the BSSID, SSID, Channel and Signal Strength
  - For the Association Response print the BSSID, Channel and Signal Strength
- **[1 pt]** Every **Re-association Request** and **Re-association Response** frame that is detected is displayed in the terminal
  - For the Re-association Request print the BSSID, SSID, Channel and Signal Strength
  - For the Re-association Response print the BSSID, Channel and Signal Strength
- **[1 pt]** Every **Disassociate** and **Action** frame that is detected is displayed in the terminal
  - For both frames print the BSSID, Channel and Signal Strength
- **[0.5 pts]** Use colorama (<https://pypi.org/project/colorama/>) to assign different colours to the frame groups (grouped above by points).

- Example:
  - Probe request and Probe response frames could be **blue**,
  - Authentication and De-authentication frames could be **green**,
  - Association Request and Association Response frames could be **yellow**,
  - Re-association Request and Re-association Response frames could be **red**,
  - Disassociate and Action frames could be **magenta**
- Available colours: black, white, red, green, yellow, blue, magenta and cyan
- dim, normal and bright can also be applied

### Sample Output

#### PROBE RESPONSE

```
-----
BSSID: 5C:71:0D:4C:17:40
SSID: eduroam
Channel: 1
Signal Strength: -92dBm
```

#### PROBE REQUEST

```
-----
SSID: eduroam
Channel: 6
Signal Strength: -94dBm
```

#### DEAUTHENTICATION

```
-----
BSSID: 5C:71:0D:4C:16:40
Channel: 6
Signal Strength: -89dBm
```

```
AUTHENTICATION
-----
Channel: 11
Signal strength: -108dBm
Addresses:
      BSSID: D0:15:A6:64:22:00
```

```
REASSOCIATION REQUEST
-----
Channel: 11
Signal strength: -60dBm
SSID: ut-public
Addresses:
      BSSID: 5C:71:0D:4C:67:81
```

### Additional Notes

- The template can be ignored, create your own file
- Every frame should be displayed, do not store previously discovered SSIDs as is done in the lab file

### Modifications

- You can use any colour you desire for the frames (once the colour does change)
- You can display your information differently, just include the specified fields

### Bonus Points

- **[2 pts]** For the management frames above
  - Add a **Destination** and **Source** address to the output for the Probe Response, Probe Request, Authentication, Deauthentication, Association Request, Association Response, Re-association Request, Re-association Response, Disassociate and Action frames

```
ACTION
-----
Channel: 5
Signal strength: -78dBm
Addresses:
      Source: 8C:0F:6F:70:70:89
      Destination: 18:E7:F4:C8:56:AC
      BSSID: 8C:0F:6F:70:70:89
```

```
ASSOCIATION REQUEST
-----
Channel: 5
Signal strength: -32dBm
SSID: 7cc198
Addresses:
      Source: 18:E7:F4:C8:56:AC
      Destination: 8C:0F:6F:70:70:89
      BSSID: 8C:0F:6F:70:70:89
```

```
PROBE REQUEST
-----
Channel: 5
Signal strength: -32dBm
SSID: 7cc198
Addresses:
      Source: 18:E7:F4:C8:56:AC
      Destination: FF:FF:FF:FF:FF:FF
```

- **[1.5 pts]** Every RTS, CTS and ACK Control Frame that is discovered is printed. The colour for all control frames can be the same.
  - For the RTS frames print the Channel, Signal Strength, Receiver Address and Transmitter Address
  - For both the CTS and ACK frames print the Channel, Signal Strength and Receiver Address
  - Files with RTS, CTS and ACK frames: `ctl_frames.pcap` and `test_pcap[1 - 4].pcap`

```
RTS
-----
Channel: 11
Signal strength: -89dBm
Addresses:
    Transmitter: BC:A5:8B:A5:F3:0E
    Receiver: 5C:71:0D:4C:4E:E0

CTS
-----
Channel: 11
Signal strength: -93dBm
Addresses:
    Receiver: BC:A5:8B:A5:F3:0E
```

```
ACK
-----
Channel: 5
Signal strength: -78dBm
Addresses:
    Receiver: 54:B2:03:14:60:07
```

- **[0.5 pts]** Every Data Frame that is discovered is printed. The colour for all data frames can be the same.
  - For each frame print the Channel, Signal Strength, Address 1, Address 2 and Address 3
    - We'll assume address 4 does not exist
  - Files with Data Frames: `test_pcap[1 - 4].pcap`

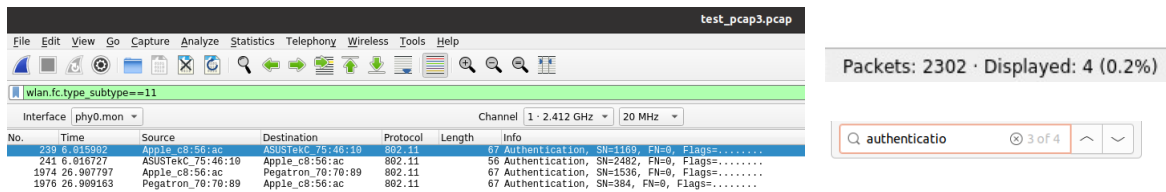
```
DATA FRAME
-----
Channel: 5
Signal Strength: -90dBm
Addresses:
    Address 1: 38:6B:1C:60:FD:36
    Address 2: 5C:E5:0C:C3:83:49
    Address 3: 38:6B:1C:60:FD:36
```

## Hints

- Use the AND operation to isolate individual bits
- Use Wireshark to figure out where data lies in the 802.11 frame

```
▼ Flags: 0x00
    ....000 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x0)
    ....0.. = More Fragments: This is the last fragment
    ....0... = Retry: Frame is not being retransmitted
    ...0 .... = PWR MGT: STA will stay up
    ..0. .... = More Data: No data buffered
    .0.. .... = Protected flag: Data is not protected
    0... .... = +HTC/Order flag: Not strictly ordered
    .000 0000 0000 0000 = Duration: 0 microseconds
Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
Transmitter address: Cisco_4c:67:80 (5c:71:0d:4c:67:80)
Source address: Cisco_4c:67:80 (5c:71:0d:4c:67:80)
BSS Id: Cisco_4c:67:80 (5c:71:0d:4c:67:80)
.... .... 0000 = Fragment number: 0
1010 1000 1010 .... = Sequence number: 2698
▶ IEEE 802.11 Wireless Management
0000 00 00 1a 00 2f 48 00 00 2f e1 6b 01 00 00 00 00 .... /H.. /
0010 00 0c 9e 09 c0 00 c0 00 00 00 80 00 00 00 ff ff .....
0020 ff ff ff ff 5c 71 0d 4c 67 80 5c 71 0d 4c 67 80 ... \q·L
0030 a0 a8 7c e1 6b 01 0c 0b 00 00 64 00 11 15 00 07 ... |·k...
0040 65 64 75 72 6f 61 6d 01 07 8c 98 24 30 48 60 6c eduroam-
0050 03 01 0b 05 04 00 01 00 00 07 06 45 45 04 01 0d .....
```

- Save the output of your scrip to a file. E.g. `python3 packet_reader.py > pcap3_output.txt`. The number of frames (e.g. Beacon, Probes) can then be counted easily.
- Use Wireshark to determine the number of frames (e.g. Authentication) that should be present in the file.



**Due Date:** Check Moodle for date

**Late Submission Deadline:** Check Moodle for date (usually 1 week after due date)

**Submission Files:** .py file(s)

**Location:** Moodle Homework 5 link