WirelessLab WS 2016/17

Homework 5: IEEE 802.11 Physical and Link Layer measurements

Group 6

Gasper Kojek, Jens Klein

Question 1: Weak vs. strong signal detection

Useful links:

- `iw` command
- about antenna options
- ani.h file - gitlab
- debug commands info - also useful info on how to set stuff in general on debugfs files
- supported ani modes

Setup:

1. Disable ANI on both `ath5k` cards: **NOTE: if nodes loose connection the ani settings will reset**

   ```
   echo ani-off > /sys/kernel/debug/ieee80211/phy0/ath5k/ani
   ```

   This changes value in `/sys/kernel/debug/ieee80211/phy0/ath5k/ani` from `AUTO` to `OFF`

   Other options for ani are:

   ```
   echo ani-auto > /sys/kernel/debug/ieee80211/phy0/ath5k/ani
   echo ani-on > /sys/kernel/debug/ieee80211/phy0/ath5k/ani

   * set lowest sensitivity (=highest noise immunity):
       echo sens-low > /sys/kernel/debug/ieee80211/phy0/ath5k/ani
   * set highest sensitivity (=lowest noise immunity):
       echo sens-high > /sys/kernel/debug/ieee80211/phy0/ath5k/ani
   * automatically control immunity (default):
       echo ani-on > /sys/kernel/debug/ieee80211/phy0/ath5k/ani

     --------------- These sometimes? don't all work as expected ----------------
   * Noise immunity level
       echo noise-high > /sys/kernel/debug/ieee80211/phy0/ath5k/ani
       echo noise-low >/sys/kernel/debug/ieee80211/phy0/ath5k/ani
   * Control OFDM weak signal detection
       echo ofdm-on > /sys/kernel/debug/ieee80211/phy0/ath5k/ani
       echo ofdm-off > /sys/kernel/debug/ieee80211/phy0/ath5k/ani
   * Control CCK weak signal detection
       echo cck-on > /sys/kernel/debug/ieee80211/phy0/ath5k/ani
       echo cck-off > /sys/kernel/debug/ieee80211/phy0/ath5k/ani
   ```

2. Select antenna port B on both `ath5k` cards. Antenna on B is connected, port A is empty.

```
echo fixed-b > /sys/kernel/debug/ieee80211/phy0/ath5k/antenna
```

   All options:

```
echo diversity >  /sys/kernel/debug/ieee80211/phy0/ath5k/antenna     use defau
lt antenna mode (RX and TX diversity) - NOTE: this does not reset the antenna
file to default value
echo fixed-a >  /sys/kernel/debug/ieee80211/phy0/ath5k/antenna       use fixed
 antenna A for RX and TX
echo fixed-b >  /sys/kernel/debug/ieee80211/phy0/ath5k/antenna       use fixed
 antenna B for RX and TX
echo clear >  /sys/kernel/debug/ieee80211/phy0/ath5k/antenna         reset ant
enna statistics
```

3. Set modulation on both cards to 24 Mbps:

```
iw wlan0 set bitrates legacy-2.4 24
```

4. Set tx power and check it, check bitrate:

```
iw dev wlan0 set txpower fixed 100    (100* mBm == 1 dBm)
```

5. Check txpower, bitrate, antenna and ani settings:

```
iw wlan0 info | grep txpower

iw wlan0 station dump | grep 'tx bitrate'

cat /sys/kernel/debug/ieee80211/phy0/ath5k/antenna

cat /sys/kernel/debug/ieee80211/phy0/ath5k/ani | grep "operating\|OFDM\|CCK" |
 grep -v "errors"
```

6. Check that nodes are connected using ping or iperf

7. Start iperf server on Node 6

   `iperf -s -u`

8. Start nc on SteppingStone

   `nc -l -p 8080 > filename.cap`

9. Start tcpdump piped to nc on Node 6

`tcpdump -i wlan1 -w- | nc 172.17.3.1 8080`

10. Iperf command on Node 15

```
for i in `seq 10`; do
  iperf -c 172.17.5.10 -u -b 25M -t 30 -l 1024
  sleep 2s
done
```

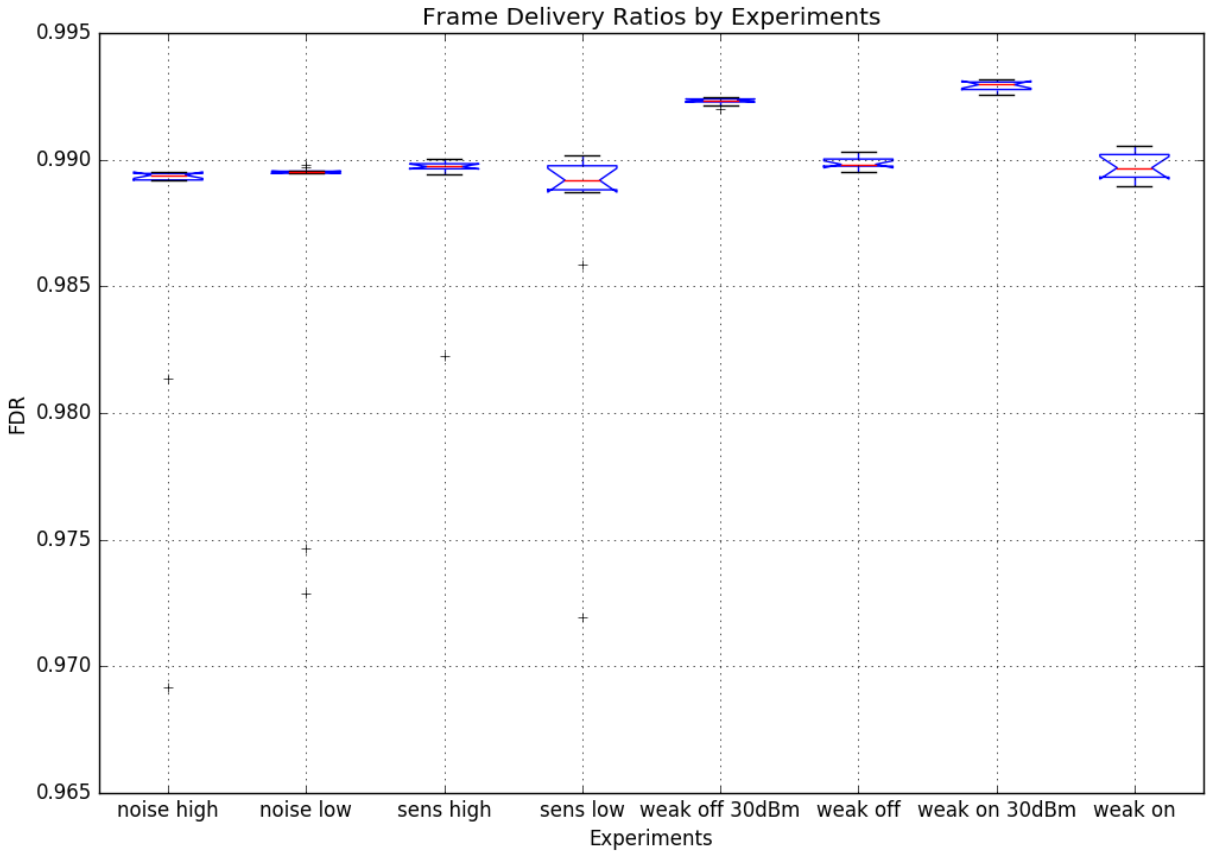
Runs:

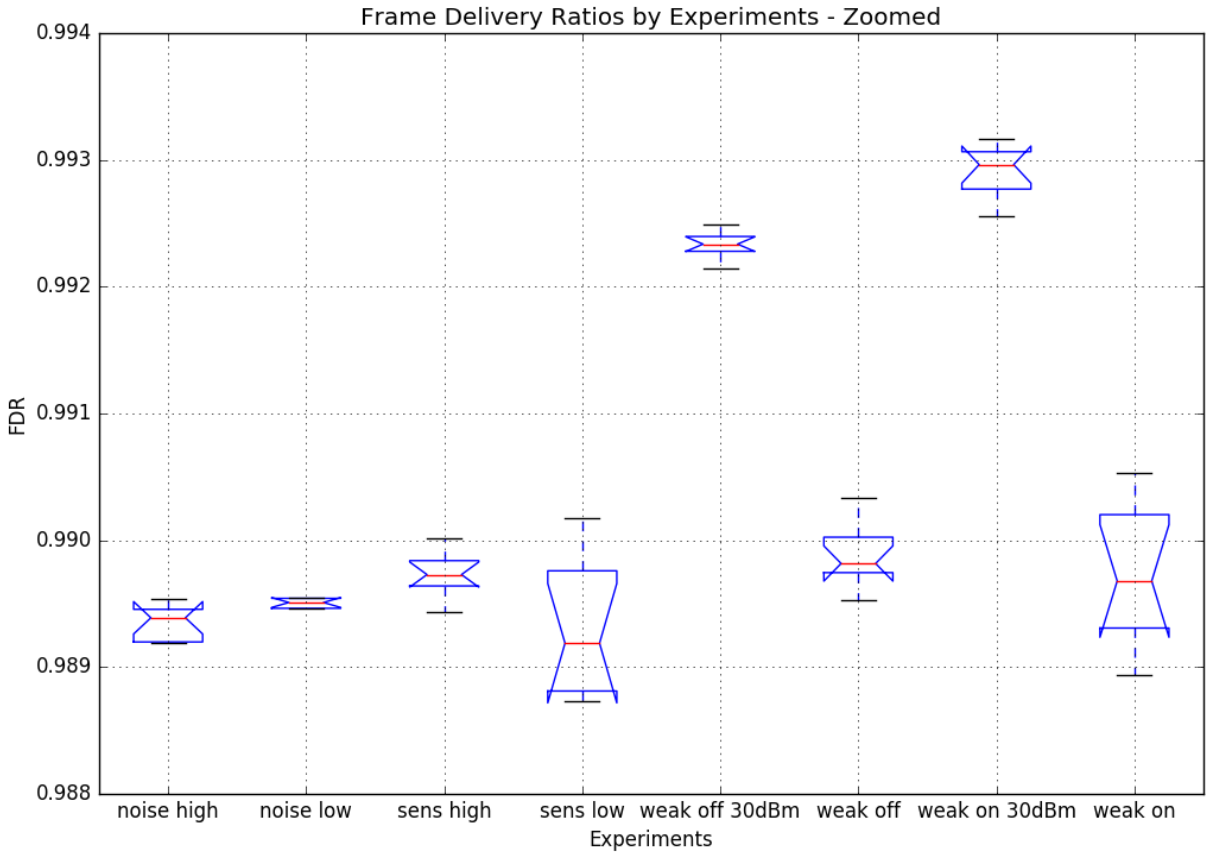
We had the same settings on both nodes (sender and receiver), to maximise the impact of different settings, as this setup impacts the frame delivery ratio for ACK packets and beacons as well. We couldn't make `antenna-a` option to connect, even with highest power, so we used `antenna=b` with lowest transmission power (except for runs 3. and 4., which are required to have substantially different power).

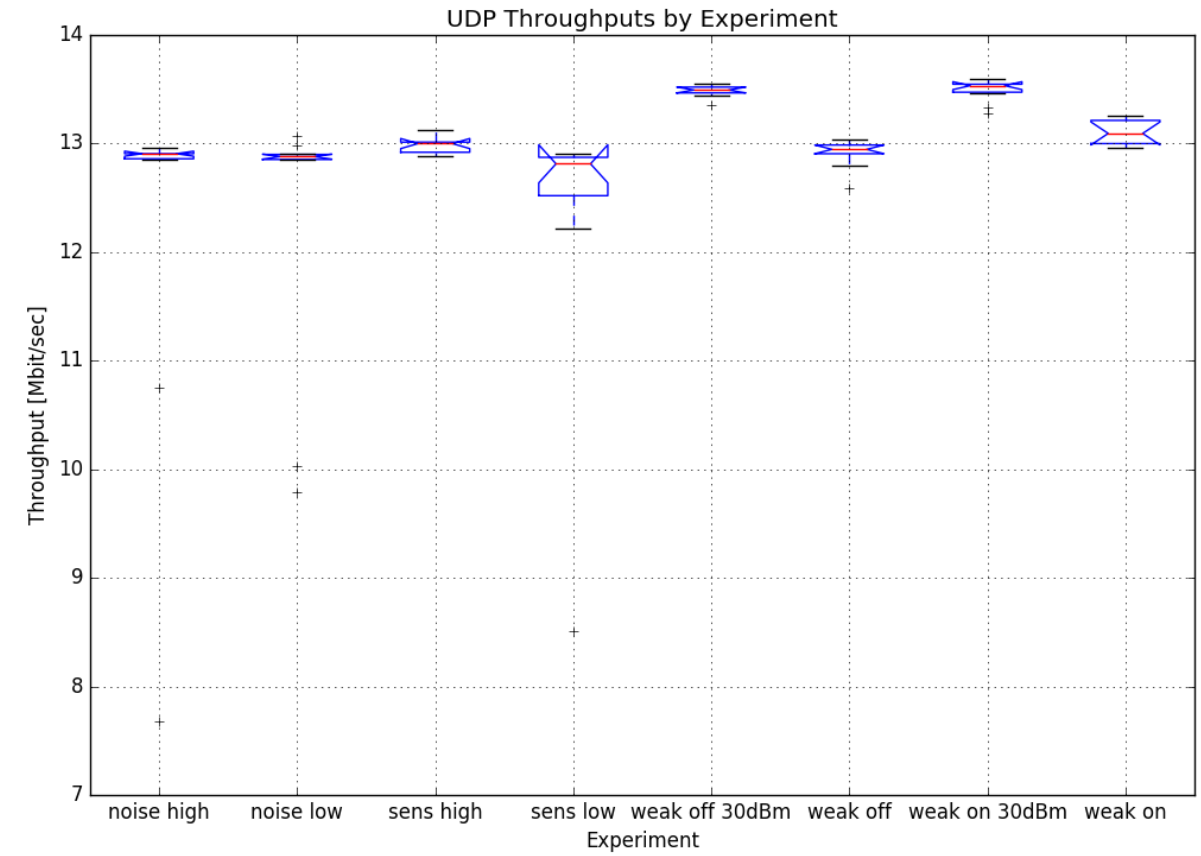
Settings not listed below were left at default values.

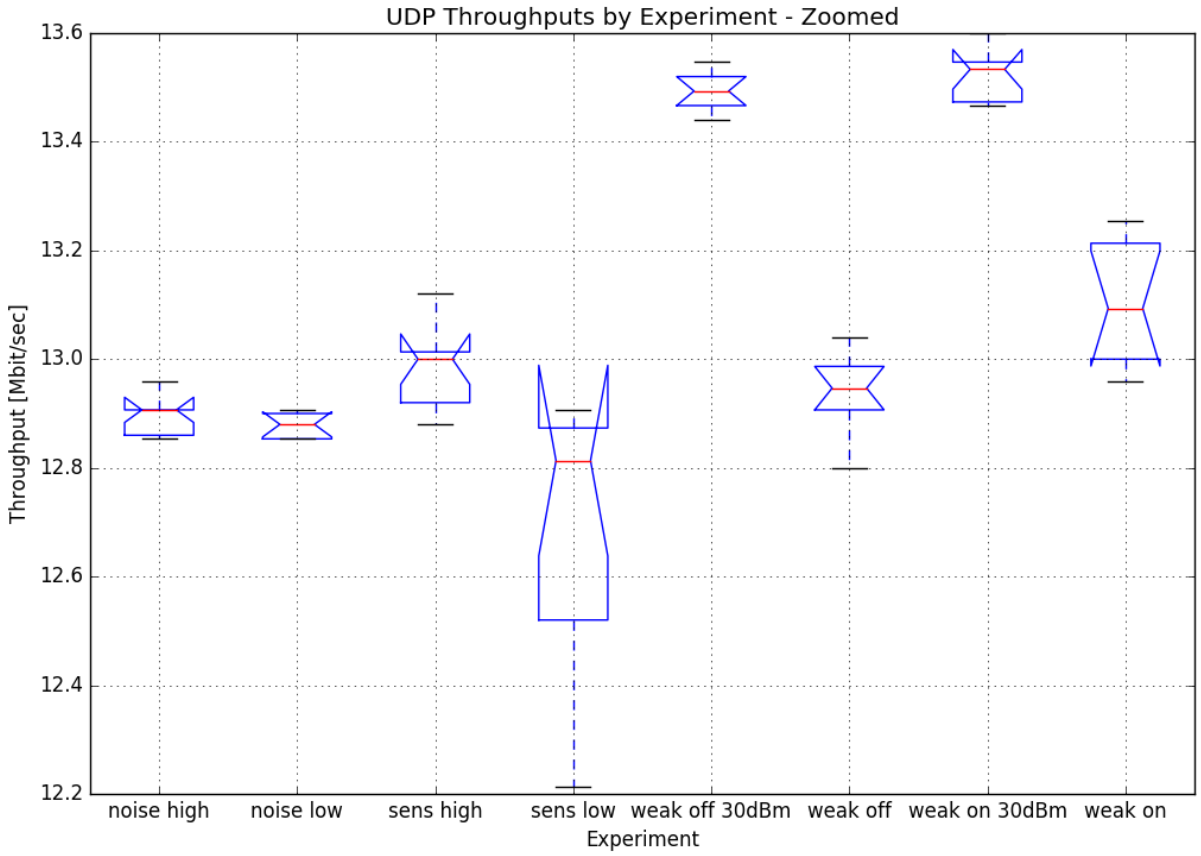
1. Ani-off, cck-off, ofdm-off, Antenna-b, txpower 0dBm: `weak-off.cap`
2. Ani-off, cck-on, ofdm-on, Antenna-b, txpower 0dBm: `weak-on.cap`
3. Ani-off, cck-off, ofdm-off, Antenna-b, txpower 30dBm: `weak-off-30dBm.cap`
4. Ani-off, cck-on, ofdm-on, Antenna-b, txpower 30dBm: `weak-on-30dBm.cap`
5. Ani-off, cck-on, ofdm-on, Antenna-b, txpower 0dBm, sens-high: `sens-high.cap`
6. Ani-off, cck-on, ofdm-on, Antenna-b, txpower 0dBm, sens-low: `sens-low.cap`
    - on sender node (node15) we had to run ani-on as otherwise the nodes wouldn't connect.
7. Ani-off, cck-on, ofdm-on, Antenna-b, txpower 0dBm, noise-high: `noise-high.cap`
8. Ani-off, cck-on, ofdm-on, Antenna-b, txpower 0dBm, noise-low: `noise-low.cap`

Results:


Frame Delivery Ratios by Experiments
FDR
0.995
0.990
0.985
0.980
0.975
0.970
0.965
noise high
noise low
sens high
sens low
weak off 30dBm
weak off
weak on 30dBm
weak on
Experiments


Frame Delivery Ratios by Experiments - Zoomed
FDR
0.994
0.993
0.992
0.991
0.990
0.989
0.988
noise high
noise low
sens high
sens low
weak off 30dBm
weak off
weak on 30dBm
weak on
Experiments


UDP Throughputs by Experiment
Throughput [Mbit/sec]
14
13
12
11
10
9
8
7
noise high
noise low
sens high
sens low
weak off 30dBm
weak off
weak on 30dBm
weak on
Experiment


UDP Throughputs by Experiment - Zoomed
Throughput [Mbit/sec]
13.6
13.4
13.2
13.0
12.8
12.6
12.4
12.2
noise high
noise low
sens high
sens low
weak off 30dBm
weak off
weak on 30dBm
weak on
Experiment

Conclusions:

Frame delivery ratio comparisons:

- When we compare runs with 0dBm and 30dBm transmission power, we can see that the 30dBm runs have a higher FDR, which was expected.
- Comparing runs with weak signal detection on and off at 30dBm, we can see that setting it to ON delivers a slighlty better FDR, as expected. What is unexpected is that when we look at 0dBm runs, there isn't an even bigger difference, but instead setting weak signal detection to on delivers slightly worse results, with a worse CI. This was really unexpected, as we thought that when the signal is lower, setting weak signal detection to on makes a bigger difference, and more importantly that it would improve the FDR.
- When comparing sensitivity levels, as expected we see that a high sensitivity performs better than low sensitivity. Low sensitivity also has a much broader median CI, which can be attributed to bigger differences between runs because of a lower SNR. Higher sensitivity has a narrower median CI, which would mean that a higher sensitivity amplifies the noise a bit less than the wanted signal, thus giving a better SNR and a better FDR.
- When comparing high and low noise immunity we would expect that a high noise immunity would perform better, but from the results we see that a low noise immunity performs better. We don't know why is that, one possibility is that the noise wasn't that bad when we did our tests and the higher immunity influenced good frame reception as well, giving a worse FDR.

All in all we see some surprises, but the biggest surprise is how small the differences between different experiments are, as the difference in FDR between the worst and the best case is below 0.5%.

Throughput comparisons:

- Here too, the runs with 30dBm power perform a lot better, which was expected.
- Comparing weak signal detection runs we see that setting this to on has a positive impact on throughput, both with 0dBm and even more with 30dBm, as expected, with a bigger performance boost with the lower power level, which makes sense, as there the signal is lower and a weak signal detection might recognize packets with an SNR lower than 14dB needed for strong signal detection.
- Comparing high and low sensitivity we again see that a higher sensitivity performs better with the added benefit of median CI being much better than low sensitivity, which can also be expected, as a lower sensitivity would have bigger differences between runs.
- Comparing high and low noise immunity we see a marginal difference with a high noise immunity being probably slightly better, but since the confidence intervals and quartiles are overlapping this

might not be the case. Our test conclude that noise immunity has the lowest impact of all settings, which is a bit surprising, as we expected it to have a bigger impact given the name of the setting.

The throughput results are much more in line with our expectations, with a stronger transmission power giving a big boost, as well as turning weak signal detection, as we expected. Furthemore, a high sensitivity setting also gives a performance boost, as well as better stability than low sensitivity, which we also expected.

Maybe the biggest suprise is that we can not directly relate FDR to throughput with a 100% confidence, as we can see here that weak signal detection at 0dBm has a worse FDR than strong signal detection, but has a much better throughput.

Question 2: Spectral Scan

The ath9k supports spectral scans. This card is installed on Node6, thus all following commands are issued on that device if not stated otherwise.

Configuration

1. Show current configuration for the scan options:

   `tail -n +1 /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_*`

   Output:

   ```
   ==> /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_count <==
   8

   ==> /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_fft_period <==
   15

   ==> /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_period <==
   255

   ==> /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_scan0 <==

   ==> /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_scan_ctl <==
   disable
   ==> /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_short_repeat <==
   1
   ```

   Review of some parameters:

   - `spectral_count` : Nr of requested scans
   - `spectral_fft_period` : Period the PHY-Layer passed frames to the overlying MAC Layer. (fft_period+1) * 4uS

     In this case every 64 micro seconds

   - `spectral_short_repeat` defines the spectral scan mode.
     - 1 = 4usec scan mode
     - 0 = 204 usec scan mode

2. Configure wlan1 to make it able to scan

- Edit /etc/config/wireless

```
config wifi-iface 'default_radio1'
    option device 'radio1'
    option network 'lan'
    option mode 'sta'
```

- Check that it scan:

  `iw dev wlan 1 scan | head -n 9` (just show first 9 lines to show if it worked)

  Output:

```
BSS a8:54:b2:71:d3:8b(on wlan1)
        TSF: 956888621281 usec (11d, 01:48:08)
        freq: 2462
        beacon interval: 100 TUs
        capability: ESS (0x0421)
        signal: -21.00 dBm
        last seen: 2370 ms ago
        Information elements from Probe Response frame:
        SSID: LEDE
```

  Yes, it worked!

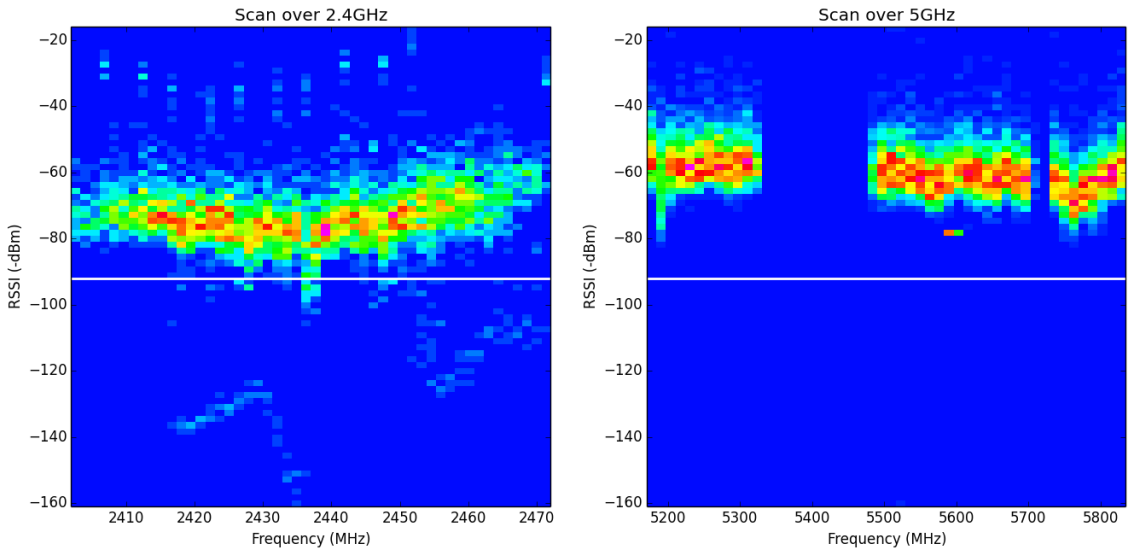
Run

1. Create shell script to run spectral scan

   `cat /root/hw05/spectral_scan`

```
#! /bin/ash

# switch on chanscan on phy1
echo chanscan > /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_scan_ctl

# perfom scan on wlan1 and drop output
iw dev wlan1 scan > /dev/null 2>&1

# save the output in /root/hw05/spectral_scan_wlan1.data
cat /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_scan0 \
  > /root/hw05/spectral_scan_wlan1.data

# disable spectral scan
echo disable > /sys/kernel/debug/ieee80211/phy1/ath9k/spectral_scan_ctl
```

2. Run `spectral_analysis.py` script to produce the heatmap



The figure shows two heatmaps. One for each spectrum (2.4 and 5GHz) from a single scan. The y-axis are syncrhonized to make a better comparison. The white horizontal line represents the noise floor of the experiment which is -94 dBm. The hue of the color represents how many fft samples were registered in discrete range of x and y, where blue are fewer hits and red more frequent hits.

In the 2.4GHz spectrum most frames were detected within -60 dBm and ne the noise floor at -94 dBm across the whole spectrum. Below this there are only a couple of detections which might be some interfering noise because it spreads across multiple channels. In between -20 and -40 dBm there are some distinctive strong signals equally distributed. They occur not on the exactly on the channels center frequency which can be a hint of **OFDM** (Orthogonal Frequency Division Multiplexing) is being used here.

The 5GHz spectrum has most of its signals with the range of -40 - 80 dBm. Which is slightly stronger then the 2.4GHz spectrum. There is also less noise on that heatmap which is natural for these frequencies, since 5GHz waves have a shorter range and less devices send on that channels. Also there are two visible gaps of signals from 5350 to 5500 MHz and around 5700 Mhz. These frequencies could be used for new devices on the medium since they are is no staturation.