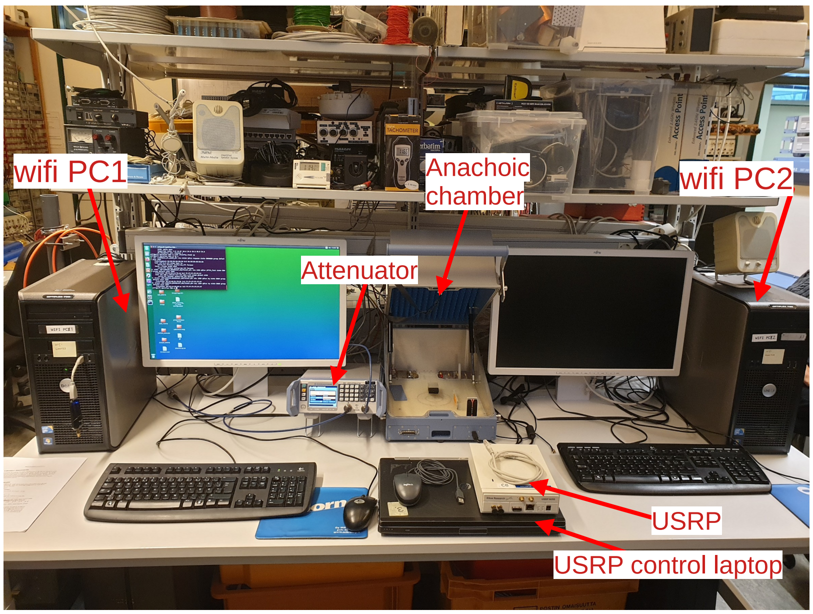
Laboratory measurements

Instructions for making laboratory mesurements of the WLAN laboratory work. Laboratory measurements consist of two parts. In part A we study the impact of packet size and signal strength to the WLAN channel throughput. In part B we study WIFI packet communication with over the air measurements. In both parts the measurement system is constructed from attenuator and anechoic chamber.

You should answer the post laboratory work questions and return them together with the measurements reports.



Part A

L1: initial configuration

In this exercise you configure the *"wifi"* driver and collect information about WLAN access points in the surroundings.

In this measuement we use in "wifi PC1" PCI WLAN adapter. The PCI WLAN adapter connection is behind the computer. Check that the adapater is connected to attenuator and attenuator is connected to anachoic chamber.

1.1 Scanning the environment

– Stop the network-manager in ”wifi PC1”

```
sudo service network-manager stop
```

– Start the wlan1 device

```
sudo ifconfig wlan1 up
```

– Scan for WLAN access points and redirect to a text file

```
sudo iw wlan1 scan > wlanscan.txt
```

Check the contents of the text file and find ”aalto open” access point with MAC address F4:DB:E6:87:2A:41. Scan again if absent.

1.1.1 Post laboratory tasks:

- Make a table for all transmitters you observe: name, used channel, power level.
- For the ”aalto open” network on access point MAC estimate what is the wall attenuation constant if the transmitter and receiver are located as in figure 1. The transmitter power is 10 dBm and before the antenna connection loss is 10 dB. The antenna gain is 0 dB. The attenuation depends on distance r and the attenuation constant α: $r^{-\alpha}$ is assumed to be $\alpha = 3.5$.

1.2 Setting up Ad-Hoc network in wlan1 interface.

– In both computers ”wifi PC1” and ”wifi PC2” stop the interface

```
sudo ifconfig wlan1 down
```

– Configure the interface to Ad-Hoc mode on channel 3 (2422 MHz)

```
sudo iw wlan1 set type ibss
sudo ifconfig wlan1 up
sudo iw wlan1 ibss join test 2422
```

– Start the interfaces

In ”wifi PC1”

```
sudo ifconfig wlan1 10.0.0.3 netmask 255.255.255.0
```

In ”wifi PC2”

```
sudo ifconfig wlan1 10.0.0.2 netmask 255.255.255.0
```

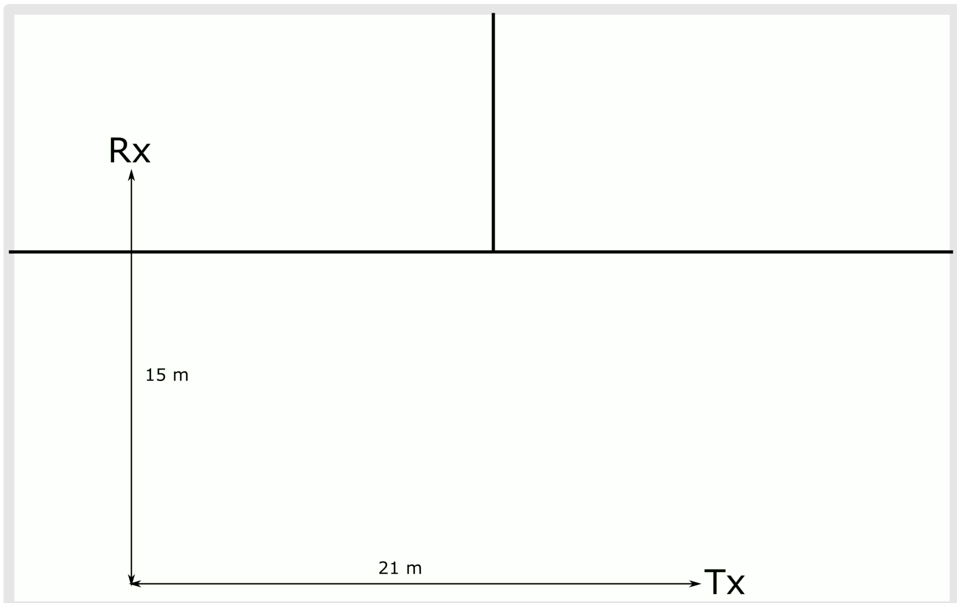


Figure 1: Locations of the transmitter and receiver.

PC	IP address
wifi PC1	10.0.0.3
wifi PC2	10.0.0.2

- Scan the interface in ”wifi PC2” and record the signal level.

  ```
  sudo iw wlan1 scan
  ```

  While recording the value set the attenuator to minimum attenuation value (0 dB) and check that the anechoic chamber is closed.

L2: Packet size and throughput

In this exercise we measure how different packet sizes impact radio interface througput. Throughput is measured with OFDM modulated 802.11g and 802.11n standards at bitrates of 54 Mbps and 65/72.2 Mbps respectively (20 MHz channel, no MIMO).

In a good channel we assume that no packet is lost due to the noise and therefore differences in throughput are only due to the packet sizes and different overheads.

The measurements are made with UDP protocol using iperf program. The ”wifi PC2” is set to be server while ”wifi PC1” is a client. In iperf the data flows from client to server, i.e. the WLAN adapter in ”wifi PC1” will be sending data frames over the channel to be received by WLAN adapter in ”wifi PC2”.

Set the ”wifi PC2” to be iperf UDP server

```
iperf -s -u -i5
```

switch	explanation
-s	server
-u	uses UDP protocol
-i	repetition time for measurements

In client side (”wifi PC1”) measure with

```
iperf -c 10.0.0.2 -b 54M -u -l1470 -t30
```

switch	explanation
-c	server address
-b	data rate
-u	UDP mode
-l	packet size
-t	time to measure

2.1 802.11g OFDM signal, 54 Mbps bitrate without RTS/CTS exchange
Disable RTS/CTS in ”wifi PC1” by

```
sudo iw phy0 set rts off
```

Measure the data rate for 5 different packet sizes between 300 and 1470.

```
iperf -c 10.0.0.2 -b 54M -u -lPacketSize -t30
```

where *PacketSize* takes your selected packet size values.

2.2 802.11g OFDM signal, 54 Mbps bitrate with RTS/CTS exchange
Enable RTS/CTS in ”wifi PC1” by setting the RTS threshold to 250 bytes.

```
sudo iw phy0 set rts 250
```

Check the `rts` value

```
sudo iw phy0 info | grep RTS
```

2.3 802.11n OFDM signal, 65/72.2 Mbps bitrate with RTS/CTS exchange

In this measurement we ”wifi PC1” computer USB connected WLAN adapter. The internal WLAN adapter supports only 802.11g but the USB connected adapter supports 802.11n mode.

RTS/CTS is already enabled in 802.11n mode to facilitate compatibility with legacy devices (802.11b/g).

In ”wifi PC2” leave the Ad-Hoc network

```
sudo iw wlan1 ibss leave
```

In ”wifi PC1” create an 802.11n Ad-Hoc network on the wlan2 device

```
sudo ifconfig wlan1 down
sudo ifconfig wlan2 down
sudo iw wlan2 set type ibss
sudo ifconfig wlan2 up
sudo iw wlan2 ibss join test 2422 HT20
```

In ”wifi PC2” join the newly created 802.11n Ad-Hoc network

```
sudo iw wlan1 ibss join test 2422 HT20
```

In ”wifi PC1” set IP address on the wlan2 device

```
sudo ifconfig wlan2 10.0.0.4 netmask 255.255.255.0
```

Measure the data rate for 5 different packet sizes between 300 and 1470.

```
iperf -c 10.0.0.2 -b 72M -u -lPacketSize -t30
```

where *PacketSize* takes your selected packet size values.

2.4 Laboratory report tasks

2.4.1 Include tables of all the measured cases.

2.4.2 Plot all the results into one figure.

2.4.3 Comment on the results.

* Why is the throughput lower with shorter packet sizes?
* Both 802.11g 54 Mbps and 802.11n 65/72.2 Mbps modes use 64QAM modulated OFDM subcarriers with 5/6 error coding rate. What is the reason for difference in nominal bitrate? What could explain the even larger difference in practical throughput? [Hint: Study the differences between 802.11n and 802.11g physical layer and MAC layer features.]

L3: Attenuation and throughput

In this exercises we measure link throughput as a function of attenuation. Check that the anechoic chamber is closed. The signal strength is manipulated by the variable attenator attached to the "wifi PC1" WLAN adapter.

3.1 Calibration of initial measurement.

- Check what is the initial signal level at "wifi PC2". Set the attenuator to a low value, for example 10 dB, and measure the received signal level.

  ```
  sudo iw wlan1 scan
  ```

- Increase attenuation by 10 dB and validate that the received signal level has changed accordingly.
- Now you know how the attenuation is mapped to the received signal level. During the measurements it is sufficient to change attenuation only and to compute the received signal level numerically.

3.2 UDP througput as function of attenuation, 802.11n 65/72.2 Mbps.

- Again set "wifi PC2" to be iperf UDP server and "wifi PC1" as iperf client.
- Measure from "wifi PC2" UDP throughput as function of received signal level.

  [Hint] In order to have interesting curve you have to identify at what signal level the connection starts to deteriorate. Take values around this level such that you can plot nice dependency curve. When the throughput begins to fall, change the attenuation in 0.5 dB steps

3.3 TCP througput as function of attenuaution.

Repeat the measurement for TCP connection.
For making TCP measurements you can configure `iperf` at the server "wifi PC2":

```
iperf -s -i5
```

and at the client ”wifi PC1”:

```
iperf -c 10.0.0.2 -t5
```

3.4 Laboratory report tasks

3.4.1 Give the tables of the attenuation versus throughput.

3.4.2 Plot the throughput as a function of received signal level.

[Note] If you recorded attenuations you have to evaluate what were the corresponding received signal levels.

3.4.3 Assuming the same parameters as in part 1.1.1 regarding power levels and losses, what would be the cell size if

3.4.3.1 the attenuation model is $r^{-\alpha}$ and $\alpha = 3.5$.

3.4.3.1 the attenuation model is $r^{-\alpha}$ where $\alpha = 3.5$ and the signal goes through one wall. Where the attenuation through the wall is computed in part 1.1.1.

Part B

In this section we examine the wifi transmissions at the physical layer. RF power inside the chamber is captured using a third antenna connected to a receiver, USRP N200. While traffic is generated by iperf over the wifi link, a short segment is sampled and converted to baseband signal.

L1: Setting up the measurement

- Connect USRP port RF1 to an SMA connector at the back of chamber. Open chamber to see which connector is connected to the antenna.
- Connect power to USRP and make connection to laptop using Ethernet cable.
- Open a terminal on laptop and navigate to *WLAN_LAB* directory. Open the measurement script *measurement.py* in text editor. Check that the IP address in script matches USRP (check label). Check the IP address of laptop is in same subnet as USRP, i.e. if USRP has 192.168.10.2, the laptop shall have 192.168.10.x.

```
ifconfig
```

– Run the script from terminal and check that there are no errors and a figure is plotted

```
python3 measurement.py
```

The script receives baseband samples recorded at 25 MS/s rate from USRP driver. Power is calculated over 10 samples yielding an array with reduced rate of 2.5 MS/s. Power is plotted over time in dB scale. Power and time vector are saved to a Matlab/Octave compatible file for further analysis.

4.1 802.11n OFDM signal, 65/72.2 Mbps data rate with RTS/CTS exchange

Reduce attenuator to 0 dB and generate traffic over the wifi link using iperf as in previous parts. Use an appropriate filename in the measurement script and run the script. Zoom in on the figure. You should be able to detect data frames, acknowledgements and RTS/CTS frames. You can detect the origin of each frame (either "wifi PC1" or "wifi PC2") by their different power levels.

4.2 802.11g OFDM signal, 54 Mbps data rate with RTS/CTS exchange

To measure with 802.11g connect the internal WLAN adapter of "wifi PC1" to the attenuator instead of the USB WLAN adapter. Make following configuration in "wifi PC1":

```
sudo wlan2 down
sudo iw wlan1 set type ibss
sudo ifconfig wlan1 up
sudo iw wlan1 ibss join test 2422
sudo ifconfig wlan1 10.0.0.3 netmask 255.255.255.0
sudo iw phy0 set rts 250
```

Generate traffic using iperf, change the filename in measurement script and run the script. Examine the figure as before and check the WLAN frames are visible.

4.3 802.11g OFDM signal, 54 Mbps data rate without RTS/CTS exchange

In "wifi PC1" disable RTS/CTS exchange

```
sudo iw phy0 set rts off
```

Again, generate traffic and use the measurement script. Remember to change the filename in the script. This time the RTS/CTS frames should be missing, leaving only the data frame and acknowledgement.

4.4 Laboratory report tasks

4.4.1 In each case plot power over time and zoom into the plot to see couple of data frames. Indicate in the plots the following: data frame, acknowledgements, RTS and CTS frames (where applicable), short inter-frame space (SIFS), DCF inter-frame space (DIFS).

4.4.2 Calculate data frame, acknowledgement, RTS, CTS, SIFS and DIFS lengths.

4.4.3 How much overhead (in time) does RTS/CTS exchange cause? Compare with measurements in part 2.1 and 2.2.

4.4.4 Compare 802.11n and 802.11g efficiency.