18-452/750 Wireless Networks and Applications

Project 1 *Due Feb 19, 5pm; upload to Canvas*

1. Objectives

This is an individual project: you must complete the project on your own.

- Experimentally monitor the physical layer characteristics of a wireless channel between multiple devices operating on a wireless local area network (Wi-Fi). Specifically you will perform:
  - **LOS Experiment**: An experiment observing the change in (1) signal strength and (2) data rate as distance is varied when the receiver and transmitter are in line of sight of each other
  - **NLOS Experiment**: An experiment observing the change in (1) signal strength and (2) data rate as distance is varied when the receiver and transmitter are not in line of sight of each other; and contrast these measurements with the line of sight data
  - **Your Experiment**: An experiment of your design to collect interesting data and present your findings
- Use the results of the experiments to gain insights into how the physical environment impacts radio frequency (RF) signals as they travel from the transmitter to the receiver and how this impacts performance.

2. Overview

Each experiment has a number of questions associated with it. The 'pre-experiment analysis' questions must be answered before performing the experiment to aid your design of the experiment, and the 'post-experiment data analysis' questions must be answered after the experiment.

1. Pre-Experiment: Modelling/Intuition Prior to each experiment, refer to the relevant lecture material to develop an intuition about what the experimental data should look like. Initially, you can assume that the experiments will be be conducted in an ideal environment,

with constant noise and no environmental variability. Later experiments will require you to consider more factors.

2. Experiments: Data Collection For this lab, you will collect data using two methods: (1) continuously collecting data during the experiment (referred to as "Continuous Data Collection") and (2) data collection at discrete points (referred to as "Discrete Data Collection").

1. In Continuous Data Collection, data is collected as an experimental variable that changes while you change the relative location of the devices, specifically you will increase the distance between the transmitter and receiver. Specifically, starting with stationary devices, you will then slowly and continuously move one device at a constant rate. This emulates a user moving around (e.g. walking down a hallway and checking e-mail) communicating with a stationary access point..

2. In Discrete Data Collection, a predetermined number of samples from predetermined points are collected. This can be performed by placing the transmitter and receiver at specific locations and collecting a number of data points at each location. This emulates a user who uses a laptop in multiple locations while sitting or standing still.

3. Post-Experiment: Data Analysis & Presentation

The physical layer data that you collect for this lab are discrete samples (one sample per packet) of a continuous signal. The channel is constantly changing due to many external factors, that can create significant variability in the samples, since the sampling rate is considerably lower than the rate at which the continuous signal changes. Thus it is important to have a number of sample data points to account for variations in the measured data. To normalize these variations, you can generate a moving average of the data samples. Techniques for presenting the data (using both the Continuous and Discrete Data Collection methods) are illustrated below. Please also reference the recitation presentation for additional information.

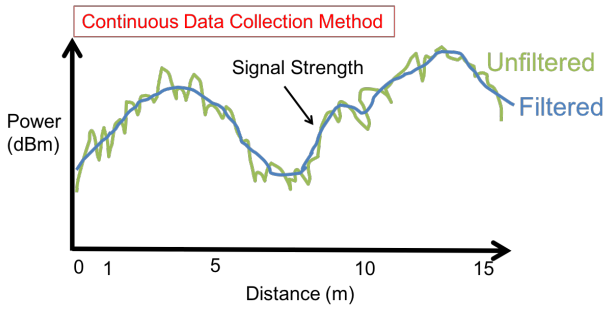

Figure 1: Continuous Data Collection with Filtering

- **Continuous Data Collection:** Figure 1 shows an example of signal strength data that is collected when we continuously changing the distance between the transmitter

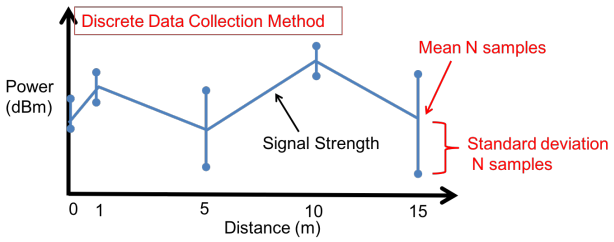

Figure 2: Discrete Data Collection plotting the Mean and Standard Deviation

and receiver. Due to the discrete sampling of a continuous signal, the data fluctuates rapidly (as shown by the green unfiltered curve in Figure 1). To aid the interpretation of the data and to approximate the true value of the continuous signal, it is beneficial to filter the data (as shown by the blue filtered data line in Figure 1). There are many types of filters that can be applied to the data (e.g. moving average, Savitzky-Golay filtering, local regression, etc). The filter selection should be based on the data, noise, and collection procedures.

- **Discrete Data Collection:** Figure 2 shows an example of signal strength data that has been collected by sampling over a period of time at multiple discrete locations. At each location, multiple signal strength data samples are collected. Depending on the environment, the samples at each location can vary significantly. To aid interpretation, the data's mean and standard deviation can be plotted for each location, as shown in the figure. Alternatively, a box-and-whisker plot could be generated for each collection event.

When observing experimental data, it can be beneficial to analyze the data from multiple perspectives. For example, when observing changes in the received signal strength, it can be interesting to also look at other metrics such as data rate (the bit rate used by WiFi) or packet loss rate at the data link layer, or throughput observed by the application. Alternatively, one can compare the results for the same experiment conducted at different locations or at different times, when the environment is more or less busy, etc.

3. Experiments

We now describe the three experiments you have to perform as part of this lab:

1. **Line of Sight (LOS) Experiment.** With the 2 devices in line of sight (LOS) of each other, use Wireshark to collect physical layer information about the Wi-Fi packets received. Start collecting measurements when the devices are very close together (a few inches if possible) and then increase the distance to at least 5 meters, or more if possible (the further, the better).

Pre-Experiment Analysis

- LOS signal strength can be modeled using the Friis transmission equation (Free Space Path Loss in the lecture slides). Create a model of your experiment showing how the signal strength will vary with distance, and generate a plot from this model.

Collect Experiment Data
The key data being collected are (1) signal strength and (2) data rate over distance. Please refer to the Appendix to check which data variables you should collect for each experiment using Wireshark. Collect this data using the following methods:

- Continuous data collection method
- Discrete data collection method

Post-Experiment Analysis

- Write 1-2 sentences describing your experimental setup
- Generate a plot of your experimental data with both (1) signal strength and (2) data rate over distance from both the continuous and discrete collection methods (one plot for each method). Signal strength and data rate can be plotted on the same figure where the left y-axis is one unit (e.g. dBm) and the right y-axis is another unit (e.g. Mb/s). This can be performed using MATLAB (specifically, the 'yyaxis' command) or using a different software.
- Write 2-3 sentences noting trends or abnormalities in the plots and describing how you processed the data.
- Write 1-2 sentences describing any differences between data collected using the continuous and discrete data collection methods.

2. **Non Line of Sight (NLOS) Experiment.** With the 2 devices *NOT* in line of sight (NLOS) of each other, collect information about the received Wi-Fi packets at varying distances, starting with the devices close together (a few inches if possible) and then increase the distance to at least 5 meters, or more if possible (the further, the better). You can create non-LOS conditions in a variety of ways, e.g., have one device around a corner, at the other side of a wall, or behind a door. Your results will depend on what type of non-LOS situation you have, which is perfectly fine.

Pre-Experiment Analysis

- Write 1-2 sentences describing your intuition about how the (1) signal strength and (2) the data rate over distance will be different from the LOS experiment.

Collect Experiment Data
The key data being collected is (1) signal strength and (2) data rate over distance. Collect this data using the continuous data collection method.

Post-Experiment Analysis

- Write 1-2 sentences describing your experimental setup
- Generate a plot of your experimental data with both (1) signal strength and (2) data rate over distance, using similar filtering techniques to those used for the LOS data.
- Write 2-3 sentences noting trends or abnormalities in the plots and describing how you processed the data.
- Generate a plot comparing the LOS and NLOS data (1) signal strength and (2) data rate over distance from both the continuous collection method.
- Write 1-2 sentences noting how NLOS data compares to LOS data. Did the results match your intuition?
- Based upon the data from the LOS and NLOS experiments, does the data rate change with signal strength? Was this expected?

3. **Your experiment.** Create your own experiment or choose among these:

   - For a fixed distance, experiment in different scenarios such as when they are obstructed by different materials (and/or combinations of materials) such as a metal, a person, a thin wall, a concrete wall.
   - Run the LOS experiment in a different space (e.g., outdoor/indoor, hallway/apartment, bedroom/bathroom/living room, etc.) and compare your results.

   Pre-Experiment Analysis

   - Write 1-2 sentences describing your intuition about your experiment.

   Collect Experiment Data
   Determine the key data that is most relevant to your experiment, and collect data using the method you determine to be best for your experiment.

   Post-Experiment Analysis

   - Write 1-2 sentences describing your experimental setup.
   - Generate a plot of your experimental data.
   - Write 2-3 sentences noting trends or abnormalities in the plots and describing how you processed the data.
   - Write 1-2 sentences describing any conclusions you can draw from your results.

4. Submission

Your lab submission should include the following items for each experiment. Note that when generating graphs, you have to decide what type of graph to use to best support your analysis.

1. **Line of Sight (LOS) Experiment.**
   (a) Plot of your model signal strength and your experimental data signal strength collected using both continuous and discrete techniques compared to distance.
   (b) Plot of your experimental data (1) signal strength and (2) data rate compared to distance collected using both the continuous and discrete techniques.
   (c) Answers to the following questions:
      i. Write 1-2 sentences describing your experimental setup.
      ii. Write 2-3 sentences noting trends or abnormalities in the plot.
      iii. Write 1-2 sentences describing any differences between data collected using the continuous and the discrete data collection methods.
2. **Non Line of Sight (NLOS) Experiment.**
   (a) Plot of your experimental data (1) signal strength and (2) data rate compared to distance collected using the continuous technique.
   (b) Plot a comparison of the LOS and NLOS (1) signal strength and (2) data rate compared to distance using the data collected with the continuous technique.
   (c) Answers to the following questions:
      i. Write 1-2 sentences describing your intuition about how the (1) signal strength and (2) noise level over distance will be different from the LOS experiment (from before performing the experiment).
      ii. Write 1-2 sentences describing your experimental setup.
      iii. Write 2-3 sentences noting trends or abnormalities in the plot.
      iv. Write 1-2 sentences noting how NLOS data compares to the LOS data. Did the results match your intuition?
      v. Based on the data from the LOS and NLOS experiments, does the data rate change with signal strength? Was this expected?
3. **Your experiment.**
   (a) Plot your findings.
   (b) Answers to the following questions:
      i. Write 1-2 sentences describing your intuition about the data you will collect.
      ii. Write 1-2 sentences describing your experimental setup.

iii. Write 2-3 sentences on reasoning about any trends or abnormalities in the plot.

iv. Write 1-2 sentences describing any conclusions you can draw from your results.

Appendix

A. Check for Monitor Mode Support

Monitor mode allows a wireless card to monitor **all** the traffic using the wireless channel that is used by your WiFi card. This includes not only the traffic addressed to your devices, but also all traffic sent by your access point and all WiFi devices using your access point to access the Internet. You will have access to both physical and datalink layer information, both of which are of interest.

Unfortunately, not all laptops support "monitor" mode, which is needed to be able to collect this information - on the wireless card and chipset, and the OS you use:

- For MacOS laptops: they use Apple's own wireless adapters and all of them support Monitor Mode (10.4.x and above). Therefore, we highly recommend you use a MacOS laptop for this project if you have one. See this website for more information: `https://wiki.wireshark.org/CaptureSetup/WLAN#macos-mac-os-x`. MacOS users should pay special attention to the notes on how to disassociate from a network since *the adapters in newer machines do not support monitor mode while associated with an access point.*

- For PC laptops running Linux or Windows, see the following instructions to check whether your laptop supports monitor mode and to enable it (in the case of Linux).

Linux

1. Use the **iw dev** command to find out the device name of your wireless card, for example, phy0. Here you can also find the default wireless interface running on managed mode (the mode used in normal operation), e.g., wlan0. Take note of the frequency this interface is using, e.g., 2437 MHz.

2. Use **iw phy *DEVICE* info** command, e.g., **iw phy phy0 info** to list interface modes that your card supports. If you find "monitor" in the Supported interface modes section, then monitor mode is supported, so you can follow the next steps to manually enable monitor mode.

3. Add a new wireless interface named mon0, which runs on monitor mode using the following command: **sudo iw phy phy0 interface add mon0 type monitor**.

4. Use **iw dev** command again and make sure that the mon0 interface is shown.

5. Disable the default interface with **sudo ip link set *INTERFACE* down** (where *INTERFACE* is the wireless interface that is configured in managed mode, e.g. wlan0). **sudo ip link set mon0 up** and **sudo iw dev mon0 set freq 2437** command to enable the new interface and set its frequency to the same frequency as the wlan0 interface (replace 2437 with your frequency).

6. Continue to the Wireshark instructions below.

Windows

1. Use the **netsh wlan show wirelesscapabilities** command and find "Network monitor mode" under your Wi-Fi interface name to see if your wireless card supports monitor mode. Windows is quite limited. There are many cases where a wireless card supports monitor mode in Linux but not in Windows. This website provides additional information about monitor mode support on Windows: `https://wiki.wireshark.org/CaptureSetup/WLAN#windows`. If monitor mode is supported, you can enable it in Wireshark.

2. Continue to the Wireshark instructions.

Linux VM + USB Wi-Fi Adapter

If you have a Windows laptop that does not support monitor mode, you can try to use a USB Wi-Fi adapter and run Wireshark on a Linux VM on top of a Windows host OS. The setup instructions are as follows.

1. Install VirtualBox.

2. Configure a Linux VM (e.g. Ubuntu, Kali Linux), i.e. set the disk, memory and CPU settings and then launch the VM to run the OS installer.

3. Install Wireshark on the VM.

4. Make sure that the USB Wi-Fi adapter is not being used by the host OS. The best way to ensure this is to disable Wi-Fi in the host OS.

5. On VirtualBox's main menu, select your VM and then select 'Settings'. Head over to the 'USB' section. In the rightmost column, select the option to add a new USB filter. Select your USB adapter. Your USB adapter should now be listed under 'USB Device Filters'. On the same menu, right-click the newly added adapter and then select 'Yes' under the 'Remote' option.

6. Launch the VM and make sure that your USB adapter is selected under 'Devices' - 'USB' in the top menu bar.

7. On the terminal, execute the set of commands that are listed above, under the 'Linux' section.

Figure 3: Diagram of Experimental Setup: Device A is running Wireshark while Device B is any device transmitting packets in the same channel.

B. Wireshark Setup

Big Picture: Monitor two devices communicating over Wi-Fi using Wireshark (installed on device A as discussed earlier).

In your setup, Device A is your laptop, while device B can either be your WiFi access point, or another wireless device that is associated with your access point. The wireless channel you are monitoring is the channel between Device B (access point or one of these clients) and your device. Each of these channels will have different properties.

This appendix explains how to use Wireshark in Monitor Mode to capture 802 network traffic in Device A's WiFi channel.

Instructions

1. **Preparation:**

   (a) As explained earlier, you need to set up Wireshark with monitoring mode (Installation: `https://www.wireshark.org/download.html`) on your laptop.

   (b) You need to select the WiFi device (Device B in Figure. 3) whose traffic you will monitor. We will assume you monitor the packets transmitted by the wifi access point, but you can also use another device using the same channel. Note that access points typically send more traffic than associated devices. that transmits Wi-Fi packets that are received by Device A.

2. **Wireshark in Monitor Mode**.

   (a) Open Wireshark on Device A. In the drop-down menu for "Capture", ensure that 'Wireless' is selected. It should display 'Wi-Fi' below with a graphical display of the number of messages received over time next to it.

   (b) Click on the 'Capture' drop-down menu, and select 'Options ...'. Looking at the new window, ensure that for the Wi-Fi Interface, 'Monitor' box is checked. With the 'Wi-Fi' Interface highlighted, click the 'Start' button.

   - Note: Not all combinations of hardware and OSes support Wireshark in Monitor Mode. Please refer to this website for additional troubleshooting assistance for specific OSes and wireless hardware: `https://wiki.wireshark.org/CaptureSetup/WLAN#Turning_on_monitor_mode`. MacOS users should pay special attention to the notes on how to disassociate from a network since

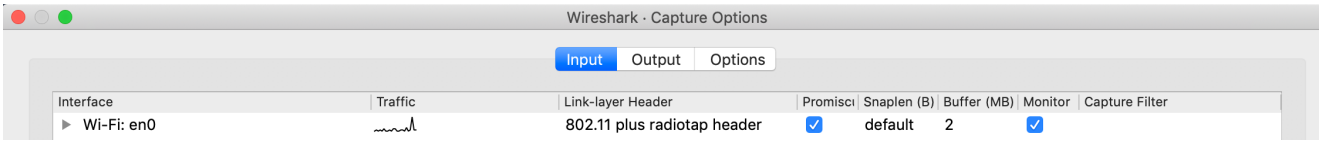

Figure 4: Setting Monitor Mode on Wireshark

the adapters in newer machines do not support capturing in monitor mode while associated.

3. **Check for Physical Layer Data**. If Wireshark and the wireless hardware are operating in Monitor Mode, when a packet is selected (i.e. clicked on) a packet dissection will be shown below. In the packet dissection, there should be a category titled '802.11 radio information'. Select this to view the values obtained.

    - If there is no '802.11 radio information' category, Wireshark is not operating in Monitor Mode. Go back to the previous step to ensure Wireshark is operating in Monitor Mode.

    The '802.11 radio information' category should include data for: 'PHY type', 'Data rate', 'Frequency', 'Signal strength (dBm)', 'Noise level (dBm)', 'Bandwidth', and 'TSF timestamp'. Observe the value of 'Signal strength (dBm)'.

4. **Sanity Check** that the measurements are acting as expected. Create a larger separation between the devices, the value of the 'Signal strength (dBm)' should decrease.

5. **Apply a Wireshark Display Filter**. Wireshark allows filtering the display so that only packets with specific properties are displayed. The goal of P1 is to measure the properties of a **single** wireless channel, you **must** use the Wireshark filter to eliminate traffic from other transmitters. To apply a filter for only viewing packets with a source MAC address of Device B, in the 'Apply a display filter ...' window you can type:

```
1 wlan.sa == 20:c0:47:20:c:7f
```

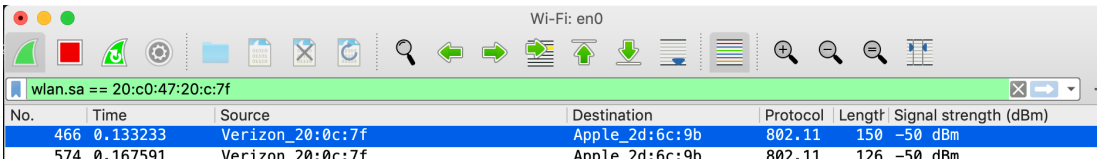

Figure 5: Apply the filter

If you are using your Wi-Fi router as the Device B, then you can use the following command to check its MAC address:

```
1 arp -a
```

```
jx-iMac:~ jx$ arp -a
fios_quantum_gateway.fios-router.home (192.168.1.1) at 20:c0:47:20:c:7f on en1 ifscope [ethernet]
```

Figure 6: Check the mac address of your connected Wi-Fi router

More info about displaying filters: `https://wiki.wireshark.org/DisplayFilters`

Note that it is interesting to briefly look at some of the physical layer information of packets when you don't use a filter. You will be monitoring multiple channels so channel properties will be very diverse. This is optional.

6. **Save packets to a file**. To process and display the data collected during experiments, save the data to a file. To save the data collected from a Wireshark capture, select the 'File' drop down menu > 'Export Packet Dissections' > select data format. Table 1 lists the available data formats.

Table 1: Data Formats Available in Wireshark

Format	**Notes**
Plain Text	Able to save all the data in the packet dissection
CSV	Only saves the data in the display window (above packet dissection information)
"C" Array	
PSML/PDML XML	
JSON	

7. **Adding Data Headers to Display Window**. Wireshark allows for data in the packet dissections to be added to the display window. This can be beneficial for quick analysis and for post-processing the data using one of the many available file formats. To add a data value from the packet dissections to the display window, right-click on the data item > select 'Apply as Column'. This will add the value to the display window, and cause those values to be saved to the CSV, "C" Array, PSML/PDML XML, and JSON output files.

C. Data to be Collected

Information being Collected:

For each experiment, collect the data variables shown in Table 2.

Note: the TSF timestamp is the measure of clock ticks where 1 tick is equivalent to 1 microsecond. The difference between two packets' TSF timestamps is the time difference between the reception of the two packets in microseconds. The Wireshark time value is within ± 5 microseconds of the difference between TSF timestamps.

Table 2: Data Categories & Example Values

Data Category	Example Value
PHY type	802.11a/n
Data rate	6.0 Mb/s
Signal strength (dBm)	-50 dBm
Noise level (dBm)	-80 dBm
Frequency	2412 MHz
Bandwidth	20 MHz
TSF Timestamp	2634373546

D. Alternative Approach

If you are using another laptop as Device B, you need to actively transmit packets in the same channel that is being monitored by Device A. The simplest way to achieve this is to execute 'ping' in a command prompt. First, determine each devices' IP address by running the 'ifconfig' command (in MacOS/Linux). For example, if Device A has an IP address of 192.168.1.1, the following command can be executed on Device B to *ping* Device A:

```
$ ping 192.168.1.1
```

It is recommended to send a flood of ping requests to Device A. To trigger a flood of ping requests from Device B, the following command can be used:

```
sudo ping -f 192.168.1.25
```

Note that "flood ping" is pretty brutal on the network, so please limit its use. An alternative is to transfer data using TCP, e.g., using iperf. The packet rate will be lower but it will also give you a throughput.

E. FAQs

- *I can't observe all of the radio tap headers ('Signal strength (dBm)', 'Noise level (dBm)', 'Bandwidth', ...) on Wireshark. What can I do?*
  - The wireless adapter (or its driver) injects these headers into the raw frames that it captures. A number of radio tap headers are defined, but not all of them are injected. It really depends on the wireless card and the driver. You can check which headers are supported by your setup by selecting a frame in Wireshark and navigating to 'Radiotap Header' -¿ 'Present flags'. Nonetheless, note that you don't need all of the headers in order to carry out this project.
- *I can capture frames on Wireshark but when I filter by the MAC address of the transmitter I can't observe any frames*

- Check: are you monitoring the right channel? You should be monitoring on the same channel that the transmitter is using. On MacOS, for example, this can be achieved by executing *airport –channel=XX* on the terminal.
- Check: did you select the right MAC address? Note that if you are using an iPhone as a transmitter, then you should disable MAC address randomization.