Project 1
Probing Wireless Channels

Peter Steenkiste and Justin Niweteto
18-452/750 - Recitation
Spring 2020

Overview

• Goals and experiments
• Experiment set up
• Analyzing the data
Big Picture

• Objective: Experimentally observe SNR of a wireless channel
• Observe how it changes depending on the physical environment
• Experiments:
  • Line of sight channels
  • Non-line of sight channels
  • Impact of distance and frequency band on throughput
• We are interested in signal strength, bit rate, and throughput
  • Use Wireshark tool to look at low level headers and signal strength
  • Use iPerf to measure throughput at the application level

How does the channel change?

- Varies over time
- Crowded space
- Obstructions
- Mobility
Tools

• The Panda wireless adaptor collects channel information
  • Must be passed up to the OS
• Wireshark is an open source tool that monitors network traffic and parses the network headers for standard protocols
  • Widely used; all layers of the stack
• iPerf is an application tool to measure throughput

Simple Set up

• Two laptops connected to Wifi using the build-in Wifi interface
  • Exchange traffic using ping
• Panda adaptor set to listen on the same frequency band
• Wireshark captures headers and collects signal strength info
  • Only monitors the channel
• The channel being monitored is that between the Panda adapter and any transmitter in range
Discussion

- The challenge is setting up Wireshark on your laptop
  - Set up depends on the OS and sometimes also on the hardware
  - This should work for some people but it some teams will have problems
  - Different set up with a Raspberry Pi as an alternative (next slide)
- Make sure that you understand what channel you are monitoring
  - Wireshark will show you traffic on many channel
  - You can use the IEEE MAC addresses to identify traffic on “your” channel
  - When changing the environment (obstacles, distance, ...) you must be aware of where the two end points are
- Flood ping should only be used between laptops
  - Can be interpreted as an attack – don’t attack servers!

Set up with Raspberry Pi

- Raspberry Pi comes with the software you need to run the experiments
- Instructions for set up are in the handout
- Ethernet provides reliable way to ssh into Pi, although WiFi should work
- Again: you need must be aware of what channel(s) you are monitoring
  - Note that the Pi needs power so you cannot move it
Overview

• Goals and experiments
• Experiment set up
• Analyzing the data

Sampling of a continuous signal

• Channel WiFi signal is a continuous signal that you are sampling at discrete time intervals—this creates variations in the measurements
Experimental Data Collection

• 2 Methods
  • Continuous Data Collection
    • Measure continuously while moving one device at a constant rate
  • Discrete Data Collection
    • Measure discrete points in the channel

What is the Signal to Noise Ratio (SNR) across this channel?

Continuous Data Collection

• Example Setup:
  • Planned path, movement rate

• Execution:
  • `sudo ping -f 192.168.1.1`
  • Walk path: moving one device (reference animation)
Continuous Data Collection Example

- Setup:
  - Planned path:
    - Devices begin together, walk away from the device, go into adjacent room, close the door, pause, open the door, walk devices towards each other, ...

- Generate data: using ping flood
  - `sudo ping -f 192.168.1.1`

- Data collected using Wireshark in monitor mode on Raspberry Pi (stationary)

- Intuition: SNR will decrease as distance increases, closing door will also decrease SNR, etc

---

Raw Data Connected

- Can connect samples with a link
- Confusing what is the data and what is the connecting line
- Matlab is a great tool to plot and process data
  - Any tool should do; scripts for processing
  - Some Matlab scripts in only copy of the slides
Matlab Code – Raw Data - Connected

% Plot SNR for all data points --connected
figure(1)
plot(data(:,time),SNR)
ylabel('SNR (dB)')
xlabel('time (seconds)')

Raw Data Discrete Pts

• All samples individual dots

• Significant variation due to sampling

• Outliers
Matlab Code – Raw Data – Discrete Points

% Plot SNR for all data points (discrete points)
figure(1)
plot(data(:,time),SNR,'.')
ylabel('SNR (dB)')
xlabel('time (seconds)')

Filtered Data

• Moving window average of data
• Selection of the window selection impacts output
  • E.g., slow fading effect for longer windows
• Multiple functions
Matlab Code – Moving Average

```matlab
figure(2)
window = 117;
p=plot(data(:,time),SNR,'b',...
    data(:,time),movmean(SNR,window,'omitnan'),'k-');
legend('raw data','moving average')
p(2).LineWidth = 3;
xlabel('time (seconds)')
ylabel('SNR (dB)')
```

Matlab – Other moving average functions

- movmean
- sgolayfilt
Discrete Data Collection

• Example Setup:
  • Planned measurement locations, planned data collection (e.g. number of packets, time)

• Execution:
  • Using ping (to send a specific number of ICMP packets):
    • Device 2: `sudo ping -f 192.168.1.1 -c 10000`
  • Using iperf3 (to send TCP packets for a specific amount of time):
    • Device 1: `iperf3 -s`
    • Device 2: `iperf3 -c 192.168.1.1 -t 30`
  • Repeat data generation operation at each measurement location

Discrete Data Collection Example

• Setup:
  • Collection Points: Selected 3 locations from Continuous data collection experiment
    • Devices next to each other
    • Devices maximum separation point
    • Devices with door closed between them
  • Generate data: using iperf3 ([https://iperf.fr/iperf-download.php](https://iperf.fr/iperf-download.php))
    • Raspberry Pi: `iperf3 -s`
    • Laptop: `iperf3 -c 192.168.1.1 -t 10`
  • Data collected using Wireshark in monitor mode on Raspberry Pi (stationary), 10 seconds for each location
  • Intuition: SNR will decrease as distance increases, closing door will also decrease SNR
    • Results should be similar to continuous data collection
Raw Data

• 3 discrete test events in one file

• Location constant for each discrete iperf3 test

• Can be grouped into each specific test location

Grouped Data

• Plot mean with error bars of standard deviation

• Allows to generalize what the sample data value is about a condition
Matlab code – Data grouping

```matlab
% Group data
% Test event times [3.9834 14.1687], [27.0304 37.2376], [51.8701 62.0550]
group1 = data;
group1(group1(:,time)>14.2,:)=[];
group1_SNR=group1(:,signal_strength)-group1(:,noise);
group1_SNR_avg = mean(group1_SNR);
group1_SNR_std = std(group1_SNR);

group2 = data;
group2(group2(:,time)<14.2,:)=[];
group2(group2(:,time)>51.5,:)=[];
group2_SNR=group2(:,signal_strength)-group2(:,noise);
group2_SNR_avg = mean(group2_SNR);
group2_SNR_std = std(group2_SNR);

group3 = data;
group3(group3(:,time)<51.5,:)=[];
group3_SNR=group3(:,signal_strength)-group3(:,noise);
group3_SNR_avg = mean(group3_SNR);
group3_SNR_std = std(group3_SNR);
```

Matlab code - Plotting

```matlab
avgs = [group1_SNR_avg group2_SNR_avg group3_SNR_avg];
stds = [group1_SNR_std group2_SNR_std group3_SNR_std];
x=[(3.9834+14.1687)/2,(27.0304+37.2376)/2,(51.8701+62.0550)/2];
figure(1)
plot(data(:,time),SNR,'.')
Hold on
errorbar(x, avgs, stds,'k*','LineWidth', 3)
Hold off
ylabel('SNR (dB)')
xlabel('test locations')
legend('raw data','grouped data')
```
Comparison

- Plot of both data collection events together to allow for easier analysis

Analysis Thoughts

- Be creative in how you analyze the data
  - Compare to simple channel models presented in class
  - Compare the results of different environments

- Can be beneficial to observe different characteristics of the data
  - Example: how to signal strength and bitrate compare
  - Bit rate versus throughput
Model Comparison

- List Assumptions
  - Tx Power: 20 dBm
  - G: -10 dB

- Remember: all models are wrong, some are useful

Components

- Can be beneficial to analyze data from different aspects