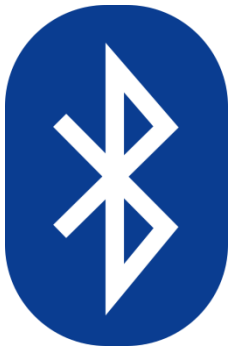
Wireless Data Transmission

CS M117 Laboratory Exercises 2 & 3



Nathan Tung
004-059-195

Lab Partners:
Alex Sanciango
Brandy Jutovsky
Mark Iskandar

Abstract

In this laboratory report, we will explore the two leading technologies for wireless data transmission – IEEE 802.11 Wireless (WiFi) LAN and Bluetooth. Since WiFi and Bluetooth share some common characteristics, both communication standards are susceptible to similar factors responsible attenuated data throughput, such as distance between nodes or interference. Nevertheless, they follow different protocols, so WiFi and Bluetooth can be expected to respond different. In Lab 2, we investigated the effect of distance, signal-to-noise ratio (SNR), and finally microwave-generated noise on the data rate of UDP and TCP protocols in 802.11n. Similarly, in Lab 3 we examined the effect of distance, multi-connection (one master node connected to multiple slave nodes), and connection interference with respect to the data rate of DH1, DH3, and DH5 packet types in Bluetooth. Finally, we take it a step further to see how Bluetooth and WiFi react when their transmission paths are crossed. This report will strive to analyze the laboratory results, explain how external factors affect data transmission, and show how WiFi and Bluetooth compare and interact with one another.

Theoretical Background

IEEE 802.11 wireless uses a special Media Access Control (MAC) layer with two modes with which to function, depending whether a base station (or access point) connected to a wired network is available. If it is, communications utilize that access point. Without that base station, computers simply send packets to one another in a mode called Ad Hoc networking. Multiplexing via dividing by dimensions (such as space, frequency, time, or code) is carried out to minimize interference and maximize utilization in channels for exchanging data. Wireless also uses the CSMA/CA (Carrier Sense Multiple Access, Collision Avoidance) protocol to exponentially back off when attempting to send data across a busy channel. CSMA/CA operates under WiFi's Distributed Coordination Function (DCF). Furthermore, in DCF, WiFi can be unreliable due to noise at similar frequencies, such as microwave interference. Therefore, frames are sent in smaller lengths in order to be received undamaged; these become fragments, each with its own checksum. Fragments sent in a row collectively become a fragment burst, increasing throughput. The alternative is Point Coordination Function (PCF), which uses the base station to control frame sending activity, thereby preventing collision altogether.

In its Transport layer, WiFi uses two main protocols: User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). TCP is a connection-based, reliable protocol, and using CSMA/CA allows it to respond to losses due to interference, fading, multipath effects, etc. Because of its acknowledgements, congestion control, and collision avoidance, TCP can be expensive in terms of retransmissions (high round-trip times). On the other hand, UDP is connection-less – it does not need to handshake with the other side to establish a communication path – and is therefore unreliable. It simply attempts to sends off packets with the assumption that the target is listening. TCP is used where data cannot be lost at all, whereas UDP is used for continuous, high-traffic communication, such as voice or video. Although TCP and UDP are used for many different things, we will be accounting for both transport protocols in our experiments regarding 802.11 WiFi.

Bluetooth (considered IEEE 802.15) is a newer technology commonly used for mobile phones and portable devices. Using radio links, Bluetooth allows low-cost, low-power, and short-range (10 meters) communication to be established without line-of-sight to form an Ad Hoc "Personal Area Network", or PAN. Like WiFi, Bluetooth also transmits data over the global 2.4 GHz frequency.

In Bluetooth, devices are either a master node or a slave node. A collection of up to seven slave nodes connected to a single master node becomes a piconet, and the entire piconet matches the master node's frequency hopping sequence and timing. Interference immunity is important to address, especially in light

of external sources like microwaves and lighting devices, and can be achieved by interference suppression (coding or direct sequence spreading) or interference avoidance (transmit signal at frequency/time when interference is low enough). In terms of its MAC layer, Bluetooth uses Frequency-Hopping Spread Spectrum (FHSS), the switching of a carrier across 79 frequency channels to transmit radio signals in data bursts – this occurs extremely quickly at a rate of 1600 hops (or channel changes) per second. This also aids in minimizing interference from a noisy channel. Finally, frames are transmitted over links, or logical channels, between masters and slaves. These links are either ACL, which are best-effort and available at irregular intervals, or SCO, which are used for real-time data and can be combined with forward error correction for high reliability.

Experiments

Lab 2 deals with 802.11 wireless LAN data transmission using both UDP and TCP. To set up the wireless connections, we use laptops with 802.11n wireless cards and the "iperf" Linux command to initialize a TCP/UDP server or client out of the laptops. We will need the target computer's IP address to form the connection with iperf. Everything is done in the terminals, and iperf will tell us if our UDP stream has lost any datagram packets.

For **Part A**, in order to determine the relationship between distance, signal strength and SNR, and data throughput, we need to laptops. One "server" laptop remains stationary while the "client" laptop moves around Boelter Hall to achieve distances of 30, 60, and 90 feet before sending data over UDP or TCP.

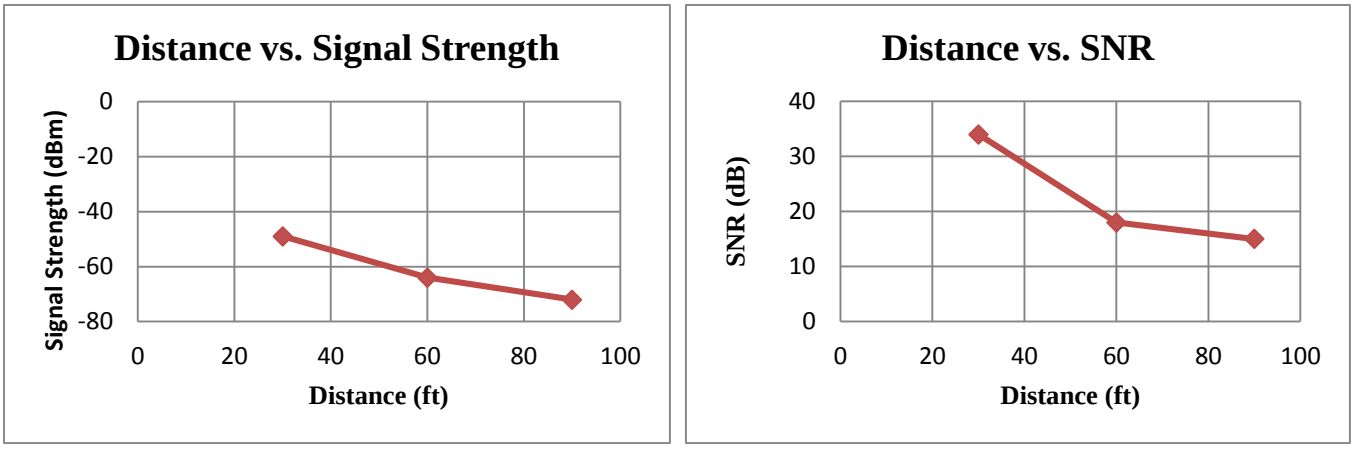


Figure 1. Distance vs. Signal Strength and SNR shows an inverse relationship

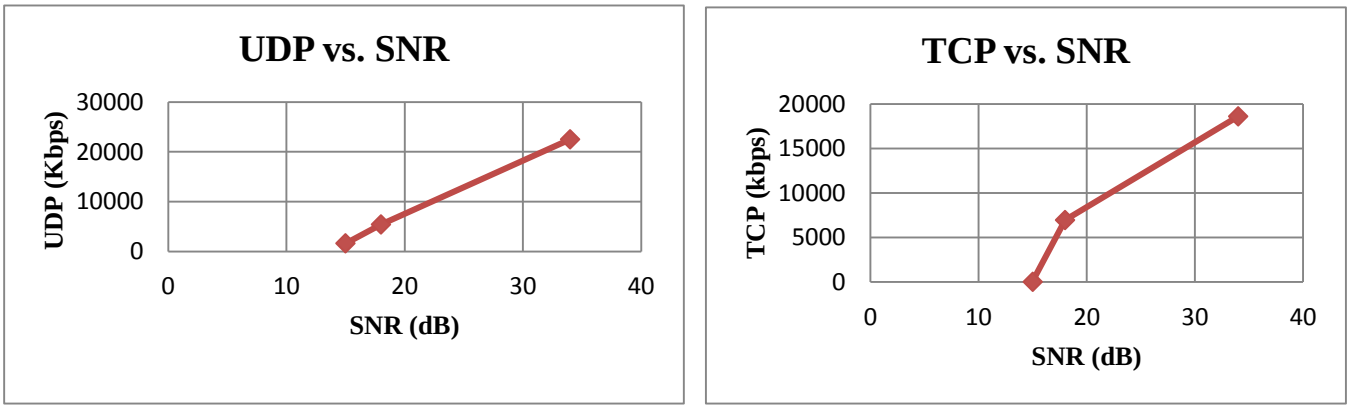


Figure 2. UDP and TCP vs. SNR shows a direct relationship

As Figure 1 shows, increasing the distance decreases the signal strength, and by extension, SNR. Figure 2 builds on this result and shows us that UDP and TCP have a direct relationship with SNR. That is, the higher the signal in relation to noise, the higher the data throughput. This makes sense, since the signal is representative of data propagation. Figures 1 and 2 together tell us that both UDP and TCP signals are attenuated by an increase in distance between two nodes.

For **Part B**, we use the microwave power level as a variable – by physically microwaving nothing in the microwave for a minute at the specified setting – to determine whether the UDP or TCP data rate changes if distance is kept constant (at 6 feet, in this case).

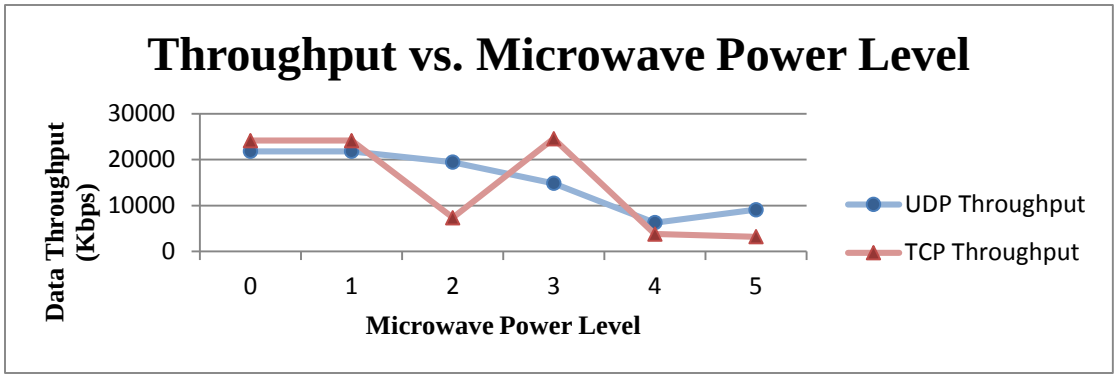
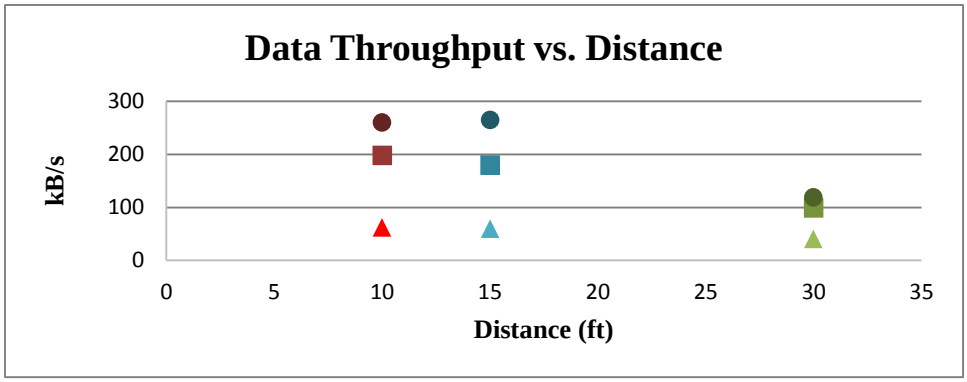


Figure 3. Throughput of UDP and TCP is inversely related with the microwave power level

In general, the graph above tells us what we might expect. Since microwaves use a frequency in the 2.4 GHz range, incrementing its power will logically increase noise at that frequency. Therefore, computers using 2.4 GHz (WiFi, in this case) to communicate will be presented with a lower SNR value. And as we see in Figure 2, that means lower data transmission rates. Specifically in UDP, high microwave interference also means datagrams are loss at a high rate.

Lab 3 deals with Bluetooth data transmission using packet types DH1, DH3, and DH5. Eventually we also use an existing WiFi connection to determine how these different data transmission types interact. To set up the wireless connections for Bluetooth, we use laptops with the Belkin USB Bluetooth card and Bluez installed and use the "l2test" and "hciconfig" Linux commands to select packet type and initialize the client. Again, everything is contained within the terminals.

To determine the effect of distance on signal strength/BER and data throughput in **Part A**, we use l2test to set up the master side. Then the slave node sends DH1, DH3, and DH5 packets over to the master node at varying distances like before, but this time at 10, 15, and 30 feet, for a total of 9 different tests.

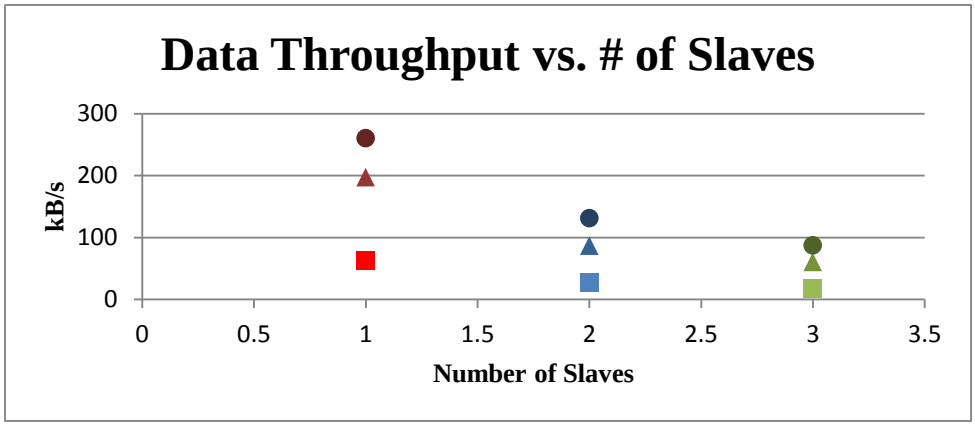


Key	
Symbol	Packet Type
▲	DH1
■	DH3
●	DH5

Figure 4. Data throughput of all Bluetooth packet types is inversely related to distance

Figure 4 drives home the fact that distance is the enemy of both WiFi and Bluetooth signal. Regardless of packet type (DH1, DH3, or DH5), all data throughputs weaken as distance between the communicating nodes increases.

In **Part B**, we group laptops together in order to join a varying number of slave nodes to the master, one per terminal. Each time a new slave node is added to the piconet, each slave again tries to transmit DH1, DH3, and DH5 packets, until we reach a total of 3 slave nodes. Again, this provides us with data on 9 tests.



Key	
Symbol	Packet Type
■	DH1
▲	DH3
●	DH5

Figure 5. Data throughput of all Bluetooth packet types is inversely related to the number of slaves

Similarly, from the chart above we can tell that Bluetooth data transmission rates also lessen as the number of slaves on a piconet increases. This logically follows from the fact that there is a limited amount of resources that is now being shared by more and more communicating nodes.

Part C requires a pair of laptops to attempt Bluetooth data transmission alone before attempting the data transmission again with two other Bluetooth-connected communication paths crossing in between.

Measurement Case	Data Rate (kB/s)	Other Observations
Before interference	262.41	Fluctuation between 170-260 kB/s
3 connections crossing	176.85	Mostly around 170 kB/s but some fluctuating
	178.17	1 rare low spike at about 90 kB/s
	176.15	Fluctuations around 70, 100, 200 and 260 kB/s

Figure 6. This chart shows the attenuation of Bluetooth transmission in the face of external Bluetooth interference

Figure 6 shows how all three crossing connections have dropped in transmission rate from the original, pre-interference data rate. Therefore, the more interference there is from unrelated but crossed-over communication paths between pairs (or more) of Bluetooth devices, the lower the data throughput that each device gets. We also note that the data rates of each device post-interference are very similar, so we conclude that the slices of throughput are quite fair.

Similarly, in **Part D**, we use a pair of laptops to attempt Bluetooth data transmission alone; next, we attempt the Bluetooth data transmission again while a pair of WiFi-connected laptops also try to communicate with their communication paths crossed.

	Data Rate (kB/s)	Other Observations
Bluetooth Throughput	174.67 kB/s	Also fluctuates between 260-282 kB/s
802.11b TCP throughput	931 kB/s	Single connection, so no fluctuations

Figure 7. This chart shows Bluetooth and WiFi TCP throughput when their paths interfere with one another

In this last part, we find that there is no fairness between Bluetooth transmission and 802.11n WiFi TCP transmission when the two connections cross over and interfere with each other. In fact, WiFi has almost 747 kB/s more in data rate and takes up 82% of the total data transfer taking place at that time!

Results

Our experiments tell us that both WiFi and Bluetooth signals can be attenuated by factors such as distance, noise and interference, and number of nodes sharing the same communication medium.

Even with these similarities, we observe the huge difference in data throughput between 802.11 Wireless (both UDP and TCP) and Bluetooth. At close range, UDP and TCP reach almost 20000 Kbps. At a closer distance of 10 feet, even DH5 (Bluetooth packet type with considerably high data rate) only hits 260 kB/s, or about a tenth of what UDP and TCP can attain at 30 feet. The same is true in other parts of the experiment. Even at its lowest throughput due to high microwave interference, TCP maintains a data rate of about 3000 Kbps. However, this makes sense, since WiFi is known for its higher bandwidth (at a higher cost in power), whereas Bluetooth is cheaper and has much less bandwidth. In addition, recall that we are using the newer and more efficient 802.11n wireless card (as opposed to the older generation 802.11b, which the lab manual specifies), increasing the gap between Bluetooth and WiFi speeds here. Bluetooth is also extremely susceptible to increasing numbers of slave nodes in a piconet. In our experiment, we found that adding even one or two more slave nodes (with a max of 7 slaves in a piconet) resulted in a noticeable and not-insignificant drop in data throughput. While devices on the same Wireless LAN must also share a channel, since the bandwidth is significantly higher, perhaps more devices can be added without as noticeable of a loss in data throughput.

Before addressing electromagnetic interference, we first note that WiFi and Bluetooth both utilize the 2.4 GHz ISM band frequency, along with other devices including microwaves, wireless phones, car alarms, etc. When many sources of interference are active, noise is generated that can result in a decrease of wireless range, mitigated wireless data throughput, or even loss of connection. In Lab 2 Part C, we looked at the fairness of multiple Bluetooth transmissions done at the same time and across the same physical path and found the data throughput to drop (logically) but in a fair way. Although we didn't do this test for WiFi devices, we can theorize that the interference would likewise cause the WiFi throughput to drop, and the data rates to be shared relatively fairly with respect to other WiFi devices.

The curious thing is what happens when Bluetooth and WiFi devices interact together given interference. As it turns out, WiFi (using TCP) takes a much larger share of data rate (931 kB/s) than does Bluetooth (174.67 kB/s). Bluetooth also fluctuates quite a bit during this interference while the single TCP connection really doesn't have room to fluctuate. This isn't surprising, since WiFi is known to be faster in normal conditions. Even considering the attenuation of signals due to interference, WiFi should still have higher data rates than Bluetooth.

Discussion

In Lab 2 Part A, the data throughput of TCP dropped much lower than expected at a distance of 90 feet. Whereas UDP dropped to around 1600 Kbps, TCP dropped to a measly 56 Kbps! (This seems even worse when compared to the almost 20000 Kbps data rate UDP and TCP were getting at 30 feet.) However, we have to remember that, with a distance of 90 feet, not only is the signal supposed to attenuate, but there is also much more room for interference from other objects, whether from signals of close frequency or from building materials. It is possible that many people were using WiFi and/or Bluetooth devices when we decided to transmit the packets, or the laptop moved and the network card was blocked by, say, a piece of metal. As Apple's website explains, barriers like metal are very high in interference potential, with concrete and plaster next in line; all of these things can be found in or around Boelter Hall.

Similarly, the spectrum analyzer in Lab 2 Part B displayed noise and the occasional one or two spikes even when the microwave was off. This signal traffic in Boelter Hall, combined with the building materials, almost certainly made the Data Throughput vs. Microwave Power Level data less than accurate. For example, setting the microwave to "medium" had almost no interference effect with TCP throughput. UDP throughput also recovered a bit on the "high" setting when it should have dropped even lower. We can say that the microwave perhaps was not outputting a constant amount of waves at the ISM frequency, or was at that moment resetting its microwave cycle. Movable objects nearby the microwave could also have reflected waves closer to or away from the laptops, affecting transmission.

Most of our values in Lab 3 seem to be on point, with the exception of some minor discrepancies. In Lab 3 Part A, our DH5 data rate should have dropped between 10 feet and 15 feet, but instead it increased by 5 kB/s. Since we did not take the average of many experimental values, it's possible that the initial data rate at 10 feet should actually have been higher due to extra interference at the beginning, and then the data rate at 15 feet would then make sense. Since each packet here is sent with such high variance and fluctuation, it would perhaps be most beneficial to take the average of these values as well. (We did not average all the data rates from each individual packet due to time constraints.)

We cannot know for certain whether results in Lab 3 Part C and D are inconsistent, since there really is no baseline and no way to measure the amount of interference (other than what we are currently experimenting with). We simply know that throughput drops in the event of interference and that WiFi trumps Bluetooth in throughput fairness. Overall, our experimental results indeed followed our expectations and the theory behind IEEE 802.11 and Bluetooth, and they demonstrated clearly the factors that affect data throughput of wireless technology.

Reference

Apple, 2014, Wi-Fi and Bluetooth: potential sources of wireless interference: Apple, Inc.

Elgargouri, A., Medium Access Control (MAC) for Bluetooth: Academia.edu.

Tanenbaum, A. and Wetherall, D., *Computer Networks*, 5th ed. Upper Saddle River, NJ: Pearson Education, Inc. 2011.

Wikipedia, 2014, Bluetooth: Wikimedia Foundation, Inc.

Wikipedia, 2014, IEEE 802.11: Wikimedia Foundation, Inc.