A Cognitive Radio Multimedia Network Testbed for Multimedia Communication

Preliminary Design Review

Eric Pires

Jeffrey Kobza

Kevin Francois

Tobias Young

University of Massachusetts Dartmouth

7 December 2017

Sponsored by: Dr. Honggang Wang

Advised by Dr. Honggang Wang, Dr. Dayalan Kasilingam and Dr. Liudong Xing

Table of Contents

Updated Requirements

Customer Requirements

As discussed in the Concept Design Review, there are steps and procedures that must be followed for the implementation of this system. Upon further research and communication with the customer, these requirements were elaborated. The only change to the requirements from the customer was requirement 4, as shown in Table 1. This requirement simply elaborated on the focus and goal of the project. The goal being to create an expandable test bed for future research.

No.	Customer Requirement	Requirement Description
1	Setup network testbed for multimedia communication	Using the USRP N210 SDR device, design and implement a reliable duplex communication network capable of transmitting and receiving data from one device to another over a common but reliable frequency.
2	Use testbed to transmit and receive a raw data sample	Testbed must be able to send and receive raw data from one device (Client) into another (Server)
3	Expand the network testbed by configuring multiple devices to work synchronously	Improve on the base testbed by adding multiple transmission and receiving components that can communicate synchronously. Use a network switch to connect devices into a central network hub
4	Develop a clear understanding of how to develop a network platform	No particular file transmission is sought, just a learning experience for future improvement and development.

Table 1: Tabular Representation of Customer Requirements

Engineering Requirements

Overall the engineering requirements went largely unchanged. There were minor edits to requirements 1.c and 1.d on Table 2. It was felt that the requirement needed to be tailored to the selected modulation scheme. Requirement 2.c was also refined to better elaborate the goal of the communication testbed.

Customer Requirement No.	**Engineering Requirement No.**	**Engineering Requirement Description**	**Justification/ Comments**	**Test Method**
1	1.a	Map out a network in which one device transmits over a frequency within the S-band ranges of 2-2.4Ghz and another receives over the same frequency	Transmission errors expected over common frequency channel, reliability aspect covered in second semester.	Inspection
	1.b	Install the USRP Hardware Driver on a host PC to detect and update firmware within the USRP devices	The USRP N210 communicates to a host environment via ethernet, and must utilize the hardware drivers distributed by Ettus Research for proper PC communications on either a Windows or Linux environment.	Test
	1.c	Using GNU Radio, implement a Transmission Protocol that can encode a data input and broadcast a radio signal within the given frequency point through the antenna port of the USRP	In order for data to be wirelessly transmitted, a data source must be encoded and modulated to a carrier waveform to be broadcast over the frequency spectrum. DQPSK selected for its support of the encoder in use.	Demo/ Analysis
	1.d	Using GNU Radio, implement a Reception Protocol that can demodulate a radio signal received from the antenna port if the USRP and decode a raw data output.	In order to receive wireless data transmissions, a USRP data source must be filtered and demodulated to recover the original data stream, then decode it to its original format. DQPSK selected for its support of the decoder in use.	Demo/ Analysis

2	2.a	Run Full-Duplex benchmark to test working specifications for sample rate and frequency communication	Benchmark will provide information that will confirm the hardware's functionality given a sample rate for Tx and Rx along with a supported frequency. Test included in UHD installation	Test
	2.b	Run GUI receiver plot and waveform generator to further test and confirm radio communications.	GUI programs provided in GNU Radio installation will provide a testable input transmission and GUI spectrum plot to view the received input.	Test/ Analysis
	2.c	Demonstrate a working file exchange between both USRP devices	Following the system diagram, utilize GNURadio to map out a transmitter and receiver protocol that sends and receives a file successfully.	Test/ Demo
3	3.a	Add additional transmitters and receivers through MIMO configuration.	MIMO (Multiple Input Multiple Output) allows for optimization of network speed by synchronizing device computation between multiple units rather than depending on one.	Test/ Inspection
	3.b	Attach main transmitter and receiver to Ethernet switch to connect all devices within network testbed. Benchmark sampling results of transmitted and received packets	Number of added devices will increase throughout the year; therefor a network switch will be utilized to accommodate all device traffic within the network.	Demo/ Analysis

Table 2: Tabular Representation of Engineering Requirements

Updated System Description and Overview

The purpose of this project is to test and develop a working network testbed utilizing Software Defined Radio (SDR) technologies as the primary means of data communications. The testbed will be the underlying system for further research within the scope of radio communications. The primary goal specified by the customer is to achieve reliable data communications between two systems connected to this network. To do so, the testbed must contain a programmed transmitter (TX) and receiver (RX) that is capable of sending and receiving raw data reliably from one point to another. The range of this communications network must work within the confines of a single enclosed area.

The **Co**gnitive Radio **M**ultimedia N**e**twork **T**estbed for multimedia communication (COMET) comprises of a network of Universal Software Radio Peripherals (USRP) configured to transmit data within the testbed. This is done through the USRP hardware Driver that allows users on a Windows or Linux System to upload firmware onto the USRP's RISC softcore microprocessor. The device accommodates many solutions for application development, including open-source platforms such as GNU Radio. This software works with the UHD drivers to provide a high-level design interface for Radio Frequency (RF) applications. This software is easily available on Unix based systems, which provide many needed dependencies missing in the Windows installation.

The USRP N210 is one of the few Network-based USRP devices that utilizes an Ethernet interface. Though this is limited to 1Gbps speed, the Ethernet interface can be utilized within a network switch for use within a central network testbed. The device also contains a Multiple Input Multiple Output (MIMO) port that can be utilized for increased spectral efficiency and linkage reliability [1]. Figure 1 demonstrates the proposed system diagram for COMET:

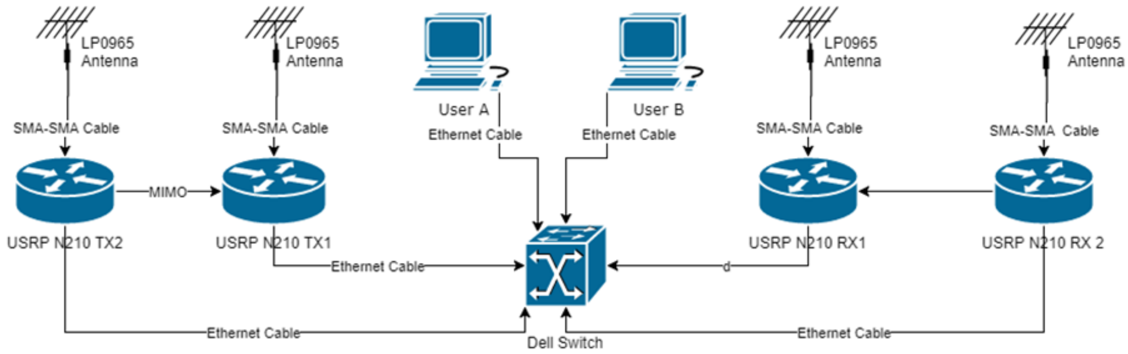

Figure 1: Network Map Version 2.0

Research on Differential Quadrature Phase Shift Keying (DQPSK) was done upon arriving at the selection of PSK modulation in the Concept Design Review [1]. This modulation type was ultimately chosen for its improved transmission reliability. Not only is it an excellent RF communication scheme, it's conceptually straightforward and it transmits and receives two bits at a time instead of one. One of the advantages of this scheme is that its information is conveyed through relative phase. In other words, every phase is dependent on the previous phase [2].

A DPQSK transmitter designed in GNU Radio in which we implemented the sending of a text file is shown below in figure 2a. In the first block, the File Source is initialized as a saved text file on one computer and sent to a Packet Encoder. This block will convert the transmitted file into packets of two to be sent smoothly through the interface. At the DPSK modulator block, the type is set to Quadrature to make it a DQPSK modulator, converted to a bit stream, and multiplied by a constant in order to keep its gain up. Finally, the UHD: USRP Sink represents the USRP device that this transmitter just sent the text file to. The QT GUI Frequency Sink block simply displays the transmitted frequency graph in a pop-up window.

The corresponding DPQSK receiver is depicted in Figure 2b. In this case, the program was run on a second computer where the USRP Source block represents a second USRP device.

Following this, another Frequency Sink is implemented to see the initial received frequency. At the same time, the USRP Source branches out to the DQPSK Demodulator block to break down the transmitted bit stream. At the packet decoder, it separates the sent packets of two and sends them to the File Sink block. This block takes the demodulated and decoded transmitted text file and places it into a new text file on this second computer. The final result can be seen much later in figure 5b satisfying the second customer requirement. In the table below are the currently satisfied customer requirements.

Rqmt ID	Requirement	Comments	Test Method
1	Setup network testbed for multimedia communication	Successfully configured a transmitter and a receiver in GNU Radio over 2.45 GHz through the UHD Drivers.	A/I/D/T
2	Use testbed to transmit and receive a raw data sample	A text file was successfully transmitted from one computer through a USRP and received by a second computer through a second USRP.	A/D/T

Table 3: Tabular Representation of Satisfied Customer Requirements

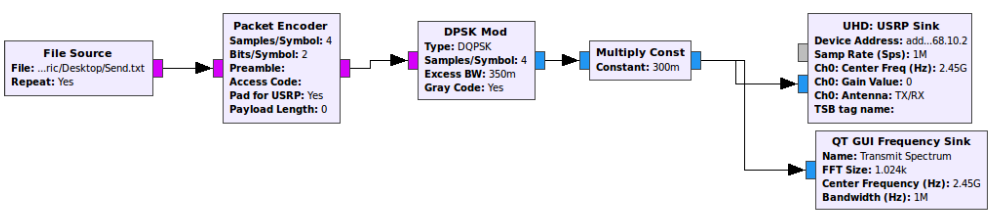

Figure 2a: GNU Radio TX System Flowchart

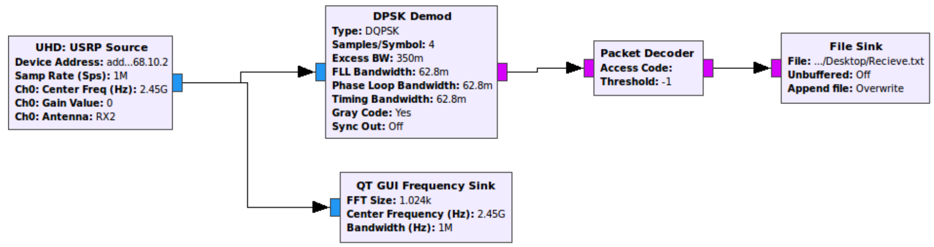

Figure 2b: GNU Radio RX System Flowchart

Constraints and Applicable Standards

The previous review reports discussed the frequency range restrictions imposed by the Federal Communications Commission (FCC). As further research and testing was done, a few more constraints were observed. The LP0965 antenna frequency range varies from 850 – 6500 MHz. This broad range falls within both the Ultra High Frequency (UHF) band and Super High Frequency (UHF) band. This must also consider the specifications of the SBX Daughterboard, which has a range from 400 to 4400 MHz Therefore, the system will be able to cover any frequency ranging from 850 to 4.4GHz. Although it limits the full capabilities of the antenna, this range is more than enough for suitable modern data communications.

Another noted constraint was the support of the development software. GNU Radio works best with a Fedora or Debian Linux distribution as discovered during through trial and error. The official installation documentation offers a simple build script that installs all necessary dependencies and packages for GNU Radio as well as the latest compatible UHD drivers [3]. The binary installers provided for Windows systems is not sufficient compared to its Linux counterpart. Therefore, the Ubuntu Operating System is currently being used to configure and program the USRP devices.

In addition, the transmitters and receivers also require Packet Encoder and Packet Decoder blocks respectively in GNU Radio. The Packet Encoder is used to send packets of a specified number of bytes at a time to the receiver. The receiver in turn will then “unpack” these bytes back to their original form using the Packet Decoder block. Without the use of these blocks in GNU Radio, the received file will be full of unintelligible garbage.

There is also a bandwidth constraint on the USRP N210 devices. Because the project uses a 1 Gigabyte Ethernet interface and an SBX Daughterboard with a bandwidth of 40 MHz, it only allows for an allotted 20 MHz of bandwidth, or 25 mega-samples per second (MS/s) that can be streamed by the host interface [4].

Analysis of Preliminary Design

Summary

The goal of the preliminary design was to experiment and analyze the functionality of the USRP hardware. Through careful research and experimentation, the design process called for a introductory understanding of radio frequency engineering through the GNU radio software. The prototype testbed can be seen in Figure 3 below.

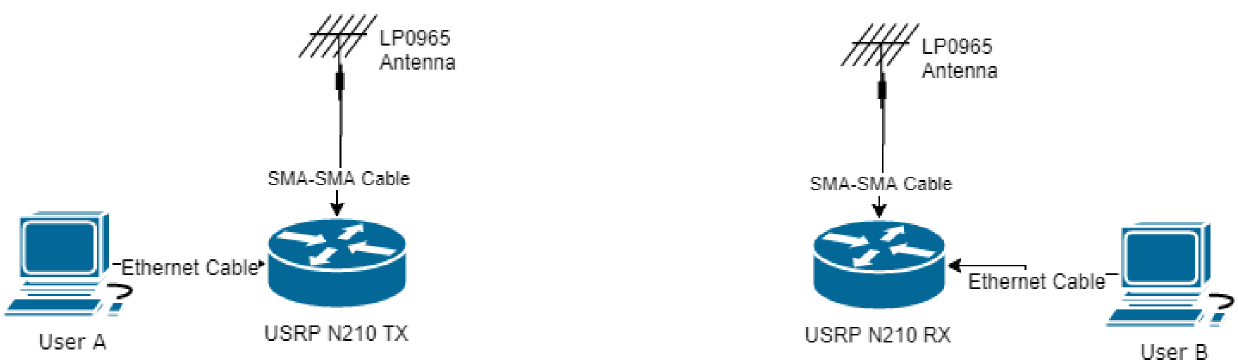

Figure 3: Preliminary Design Prototype

The design above was the result of a semesters work of wireless transmission research and testing through the GNU Radio software. This design includes two USRP devices, each responsible for Transmission and Reception respectively. They are both connected to host devices via ethernet for script execution and file exchange. User A contains the sample text file that is transmitted through the system specified in Figure 2a. This data is encapsulated and broadcast through the connected antenna, then decapsulated and received by the system specified in Figure 2b. Upon testing a sample text file, the File Sink and Source blocks were modified to include a PNG file to test image transmission. Much of this design can be accredited to GNU radios open-source repository, which provided many examples of similarly structured systems [5].

Testing Results

When first tested, the GNU radio code successfully compiled and was burned to the USRP's memory for execution in the testbed. The scripts for both the transmitter and receiver included a graph of the relative gain against the frequency spectrum. The activity observed in the se graphs demonstrated a clear response from one end the other. When observing the resulting file on the receiver's end, however, the data was not readable. Upon further analyzation, it was noted that the modulated data was never encoded. Without encoding the data, the received file was unable to properly encapsulate the TXT file in its proper format. To resolve this issue, the encoder block was incorporated to send two bits per sample (which the documentation clarified as a requirement for the given modulation scheme). The test was performed again and resulted in a success. The sample file was transmitted repeatedly and captured by the receiver and stored in a new TXT file on a new computer. The only possible form of communication between the devices was the protocol designed and executed in lab. This proved the testbed's functionality viable for data communications. The observed testing table and waveforms can be seen below in Table 3 and Figure 4.

Test	Test Description	Expected Result	Result/Comments
Verify USRP connection	Connect USRP to PC via Ethernet then run uhd_usrp_probe on Terminal to get response from device	Probe returns device IP and specifications of hardware.	Probe is successful, returns device specifications
Run and verify TX system	Generate DQPSK TX Code and upload on USRP	TX GUI should appear, TX LED on USRP should turn on to show that the device is in transmission mode	Success, the USRP is transmitting over the frequency spectrum (See Figure 5a)
Run and verify RX system	Generate DQPSK RX Code and upload on USRP	RX GUI should appear, output file should be generated in specified directory, RX LED on USRP should turn on to show that device is in receiving mode.	Failure, the USRP is properly receiving over the frequency spectrum and shows repeated spectrum activity. However, the generated file contains garbage values and does not contain the contents of the original transmitted file.
	Add encoder block and repeat second step	RX GUI should appear, output file should be generated in specified directory, RX LED on USRP should turn on to show that device is in receiving mode.	Success, the USRP is receiving the transmitted data over the frequency spectrum. File is generated holding the contents of the original transmitted file. (See Figure 5b)

Table 4: Prototype Test Method

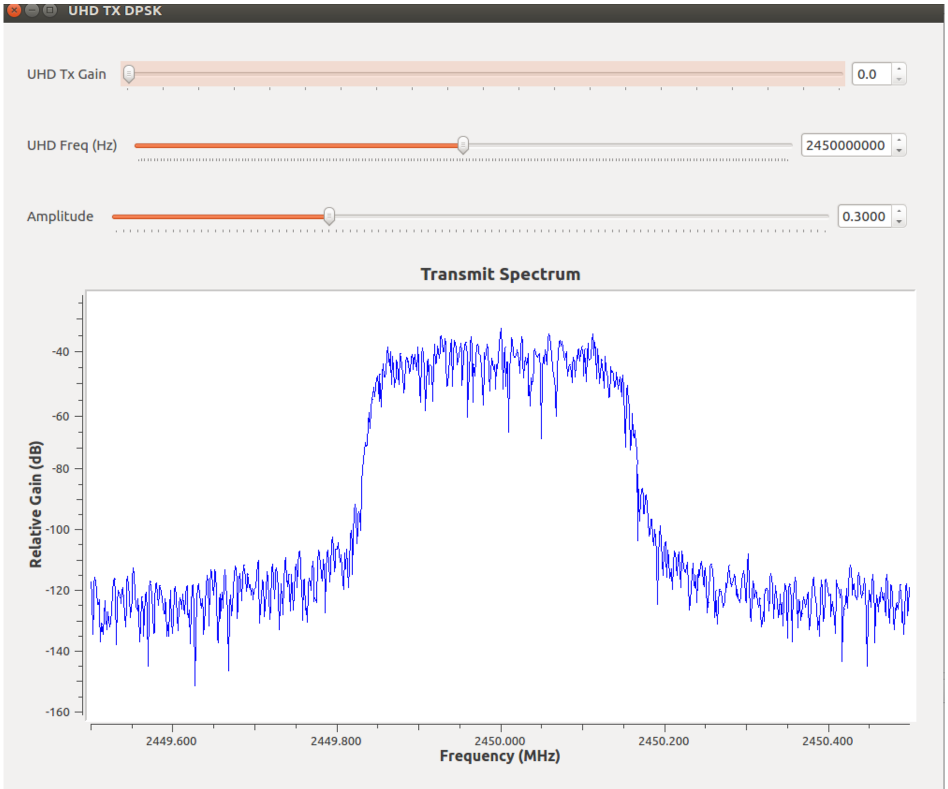

Figure 5a: TX Waveform

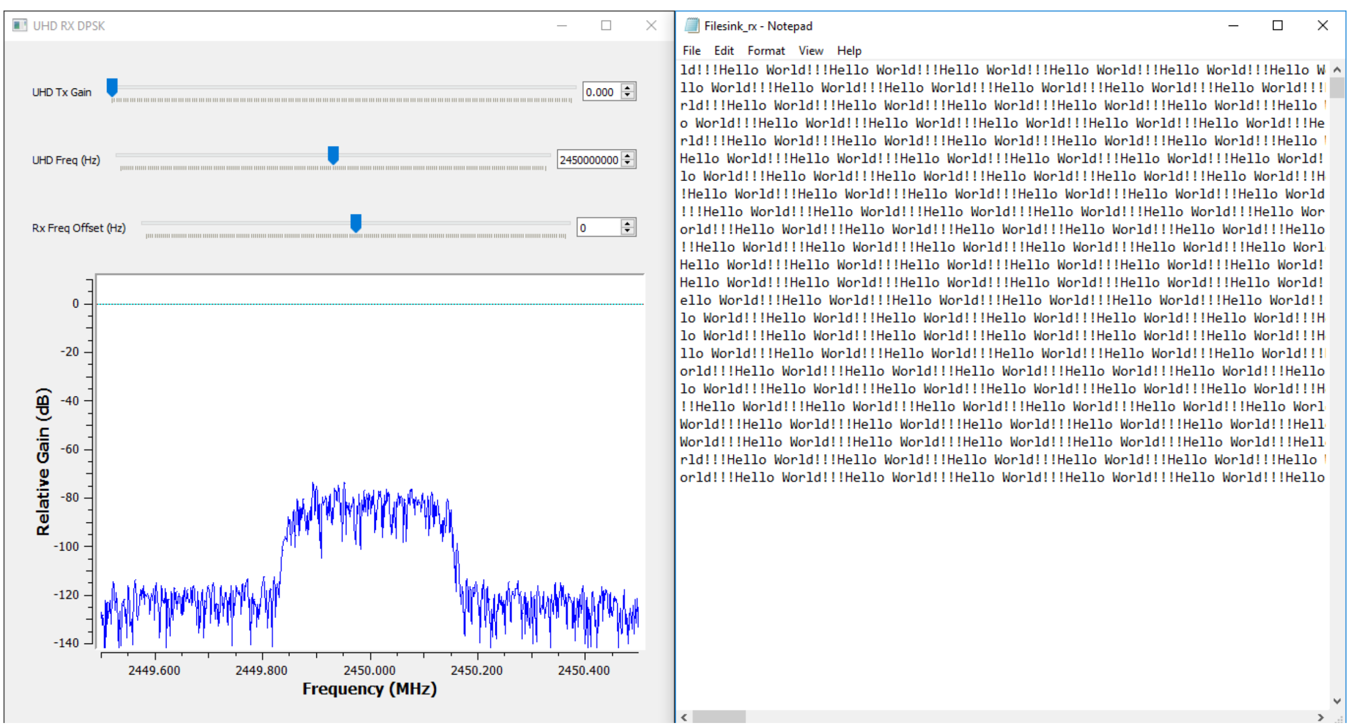

Figure 5b: RX Waveform with received file output

Outlook

Overall, the first semester design touched upon two of the three major engineering requirements. The completed tasks were arguably the most difficult to accomplish, given our group's limited background in RF engineering. Going forward, the next major requirement will require the utilization of the MIMO option and connected the devices in a network switch. MIMO is a plug and play protocol that simply requires minor changes to the UHD Source and Sink block in GNU Radio [6]. The switch installation will require a new piece of hardware that must be configured and wired to accommodate multiple ethernet connections to both users and USRP devices.

Problems and Risks

Problem: Image Transmission

The testing phase encountered some issues with image transmission. Unlike the text file exchange, the PNG image transmission proved to be unsuccessful. Despite not being a requirement, image transmission is something that can increase the appeal of the design. Therefore, new encoding methods will be researched to improve the functionality of the system.

Problem: Device Addressing

One problem that must be taken account for is the USRP IP addressing. As of now, each USRP has the default IP address (192.168.10.2). Since the devices are only required to communicate to one PC directly, there was no need to differentiate the devices within the testbed. However, as the network switch is incorporated, the USRPS will need to be uniquely addressed with new IP addresses. This will allow for the end user to pinpoint one of the several connected devices through connecting to the switch rather than directly to the USRP.

Risk: Bandwidth Allocation

One risk to observe moving forwards is the bandwidth allocation of the testbed. As more USRP devices are used, the amount of allocated bandwidth will increase. As noted in the constraints section, the ethernet connection can support up to 25 MS/s. Therefore, it is necessary to allocate a reasonable amount of shared bandwidth between the transmitting and receiving devices to ensure this constraint is met appropriately.

Updated Plan and Schedule

The progress timeline originally conceived in the previous report has matched well with our actual progress. At the moment, the USRP devices have been configured and tested. Despite this success, the estimations were very off for different steps of the project. The biggest discrepancy was the time needed for configuration. The previous schedule had this step completed within an hour. Unfortunately, this took much more time to accomplish. The issue lies within GNU Radio and its Windows compatibility. The binary installer does not include all required dependencies, requiring more individual installations. The build scrip provided in the GNU Radio installation documentation provides all required dependencies for Linux systems. Once the installation was completed, the software design took less time than expected. This is due to the vast open-source library offered by GNU Radio to provide a better starting point for RF system programming. Table 3 demonstrates the final schedule vs the originally conceived schedule

Task	Team Member(s)	Time	13-Oct	20-Oct	27-Oct	3-Nov	10-Nov	17-Nov	24-Nov	1-Dec	8-Dec
Draft Final Report for System Requirements	Eric	2 - 4hrs									
Configure USRP drivers on host PC	Eric	1 hr									
Configuration Documentation	Eric	2 hr									
Write-up (Tx Research)	Kevin	2 hr									
Write-up (Rx Research)	Jeff	2 hr									
Write-up (MIMO hardware testbed Research)	Toby	2 hr									
Research and Design a Transmission schematic on GNU Radio	Kevin, Eric	10-20 hrs									
Research and Design a Receiver schematic on GNU Radio	Jeff, Toby	10-20 hrs									
Concept Design Review	Eric	10-20hrs									
Integrate Transmitter/Receiver Phase 1 Testbed	Eric, Toby	5-10 hrs									
Test and Document Transmitter/Receiver Phase 1 Testbed	Kevin, Jeff	5-10 hrs									
Preliminary Design Review	Eric	10-20 hrs									

Table 3a: Original COMET Project Gantt Chart for Phase 1

Task	Team Member(s)	Time Spent	13-Oct	20-Oct	27-Oct	3-Nov	10-Nov	17-Nov	24-Nov	1-Dec	8-Dec
Draft Final Report for System Requirements	Eric	3hrs									
Configure USRP drivers on host PC	Eric, Toby	10hrs									
Configuration documentation	Eric	2hrs									
Write-Up (Tx Research)	Kevin	2hrs									
Write-Up (Rx Research)	Jeff	2hrs									
Write-Up (MIMO hardware tesbed research)	Toby	2hrs									
Research and design transmission schematic on GNURadio	Kevin, Eric	15hrs									
Research and design a receiver shematic on GNURadio	Jeff, Toby	15hrs									
Concept design Review	Eric	5hrs									
Integrate phase 1 Transmitter/Receiver testbed	Eric, Toby	7hrs									
Test and document phase 1 testbed	Kevin, Jeff	7hrs									
Preliminary Design Review	Eric	7hrs									

Table 3b: Final COMET Gannt Chart for Phase 1

Materials List

The following equipment has been used throughout the semester:

- 2x USRP N210
- 2x SBX 400-4400 MHz Rx/Tx Daughterboards
- 2x LP0965 PCB Antenna
- 2x USB 3.0 to Ethernet Adapter

- 2x Ethernet Cat 5E cables
- Windows PC
- Ubuntu PC

The equipment list will expand for the next semester to include the following components: -

- 2x MIMO cable
- 2x Dell Networking 5524 Gigabit Ethernet Switch
- 2x USRP N210
- 2x SBX 400-4400 MHz RX/TX Daughterboards
- 2x LP0965 PCB Antenna

In addition to the specified hardware, the project utilized the GNU Radio Open-Source software on both PCs for coding the USRP devices.

References

[1]"gnuradio/gnuradio", *GitHub*, 2017. [Online]. Available: https://github.com/gnuradio/gnuradio. [Accessed: 02- Dec- 2017].

[2] P. Krishnan, "What are the advantages of differential phase shift keying (DPSK) over phase shift keying (PSK)?", *Quora*, 2017. [Online]. Available: https://www.quora.com/What-are-the-advantages-of-differential-phase-shift-keying-DPSK-over-phase-shift-keying-PSK. [Accessed: 16- Nov- 2017].

[3] “InstallingGRFromSource", *GNU Radio Wiki*, 2017. [Online]. Available: https://wiki.GNU Radio.org/index.php/InstallingGRFromSource. [Accessed: 16- Nov- 2017].

[4] “About USRP Bandwidths and Sampling Rates” *Ettus Research*, 2016. [Online]. Available: https://kb.ettus.com/About_USRP_Bandwidths_and_Sampling_Rates. [Accessed: 6- Dec- 2017].

[5] "gnuradio/gnuradio", *GitHub*, 2017. [Online]. Available: https://github.com/gnuradio/gnuradio. [Accessed: 06- Dec- 2017].

[6] “Synchronization and MIMO Capability with USRP Devices” *Ettus Research*, 2016. [Online]. Available: https://kb.ettus.com/Synchronization_and_MIMO_Capability_with_USRP_Devices [Accessed: 6- Dec- 2017].