Latency of Cell Phone Communication

Shunyue Yuan, Zhuozhen Cai, Chongwen Lu
University of Illinois at Urbana-Champaign
PHYS 398 DLP, Fall 2020

Abstract

In this project, we investigate the relation between latency and the speed of cellular networks (dominant factor), as well as the stability of latency in both long and short time scales. We design a hand-held device and write programs to measure the latency in cell phone communications. Our approach is to make a phone call to collect data: the caller plays a piece of audio and records, the answerer records the audio delivered by phone call. By analyzing the time difference between the sender and the receiver's signals, we are able to calculate the latency. The hand-held device includes an external power supply, an Arduino Mega 2560 microcontroller, and multiple sensors. The data acquisition program will enable us to record a piece of audio, and will provide us information to calculate the precise start time of a recording. The data analysis program reads a binary file and analyzes the latency by minimizing the chi-square between two sets of data, it will also calculate correlation between two sets of data to minimize the error.

Introduction & Background

1. Cellular Latency

Cellular latency is defined as the time it takes for a source to send a packet of data to a receiver. It might be influenced by the speed of the network and the distance between caller and receiver.

2. Application

Cosmic ray showers are cascades initiated by cosmic rays interacting in the atmosphere. Atmospheric muons are an essential component of cosmic ray showers and we can observe the cosmic ray by detecting and counting the number of charged particles in the acceptance of our apparatus. However the air shower can be extensive, which means that we need to put multiple sensors at different locations. In this case, an alert sensor will tell other sensors to work when cosmic ray showers arrive, and the central computer of the array will collect the data from sensors periodically (typically every 30 sec). The signals and data are usually transmitted by cables. In our project, we will investigate the stability of cell phone networks, and the feasibility of employing cell phone networks for signal cross-timing in extended sensor arrays.

3. Arduino

Arduino is an open-source hardware and software company that provides single-board microcontrollers and microcontroller kits for building digital devices. In our project, we will wire various sensors to the Mega 2560 microcontroller. We will use the Arduino Mega 2560 microcontroller to execute our data acquisition program and to control the sensors.

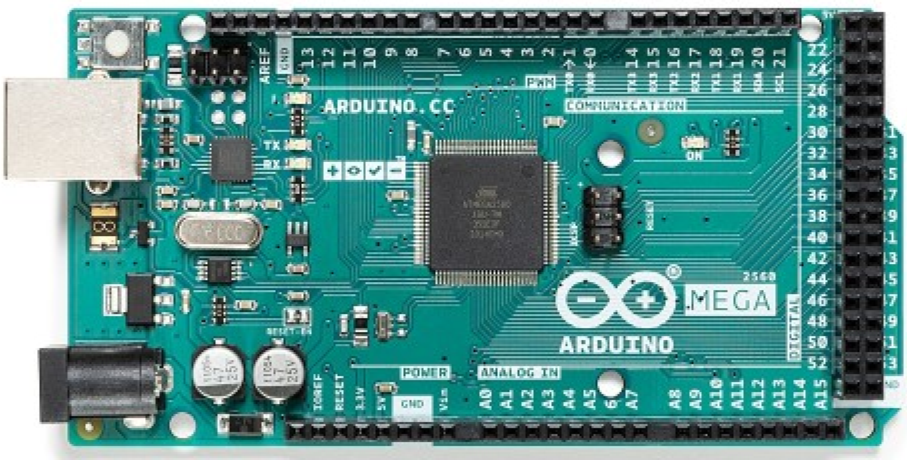


Fig 1. Arduino Mega 2560 Microcontroller.

4. Arduino IDE

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. The Arduino Integrated Development Environment is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards.

5. Sensors:Real Time Clock (RTC) & GPS

A real-time clock is a clock that keeps track of the current time and can be used to program actions at a certain time. Most RTCs use a crystal oscillator whose frequency is 32.768 kHz. The RTC has a time accuracy of $\pm$ 2parts per million (PPM) from 0°C to +40°C *[10]*, which is a drift of $\pm$ 2 microseconds per second. In our experiment, we will also use GPS to calibrate the RTC to maintain a better accuracy. For calibration details refer to pages 5 & 9.

6. Types of Networks

- LTE (Long Term Evolution) is a mobile Internet technology standard. It is the fourth generation wireless mobile telecommunications technology. It is also called 4G or 4G LTE.
- Voice Over LTE (VoLTE) is an advanced technology so that VoLTE network supports both voice and data simultaneously, without interrupting each other. The traditional LTE networks do not support data and voice together or can affect the quality of data or call.
- 3G is the third generation of wireless mobile telecommunications technology.

Methodology

To measure the latency, we designed a method (as shown in Fig 2): when making a phone call, the caller plays a piece of audio via a speaker, then the caller records the audio directly, and the answerer records the audio delivered by phone call. By comparing the time difference between the signals in two recordings, we are able to calculate the latency.

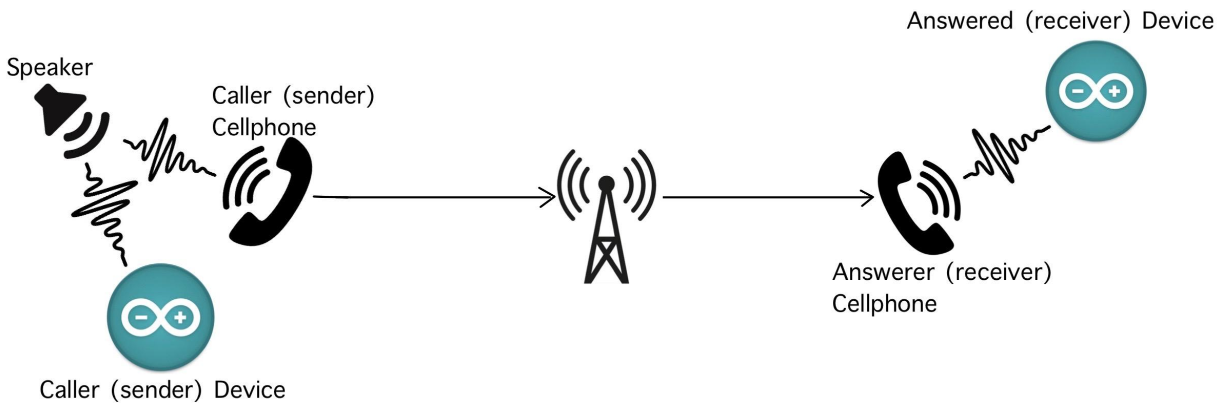


Fig 2. Layout of latency measurement process.

1. Hardware

At the first stage, we assembled the Arduino microcontroller and various sensors onto our breadboards (refer to Fig. 3). They were wired following the schematic of the circuit design and primarily used to test the functionalities of Arduino and sensors. With the preparation work done by Professor Gollin, we built our printed circuit boards (PCBs) based on the circuit from our breadboards for further data collection and kept our breadboards as backups.

The PCB works as a more compact system that integrates sensors and components that were specifically required for our project. These components include

I. the Arduino Mega2560,

II. the MAX4466 Electret Microphone, which records the sound signal with a 40kHz sampling rate into binary files. It is a type of electrostatic capacitor-based microphone, with a frequency range from 20Hz to 20KHz. The microphone sends data to Arduino at the rate 40kHz, and the Arduino then changes the voltage data to ADC (Analog-to-Digital Converter) values.

III. the DS3231 I2C real-time clock (RTC),

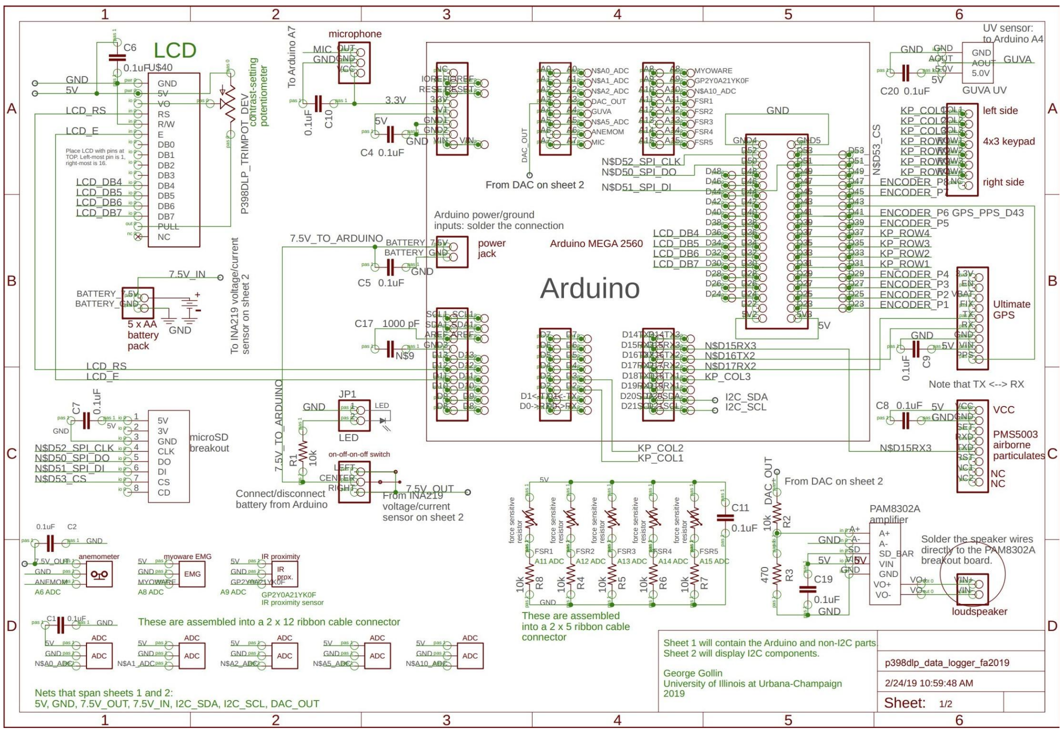
IV. the ultimate GPS, which generates a pulse per second (PPS) signal and is used to gather location information. It sets RTC based on coordinated universal time (UTC), which is the time used in Greenwich Mean Time zone and is six hours ahead of the central standard time.

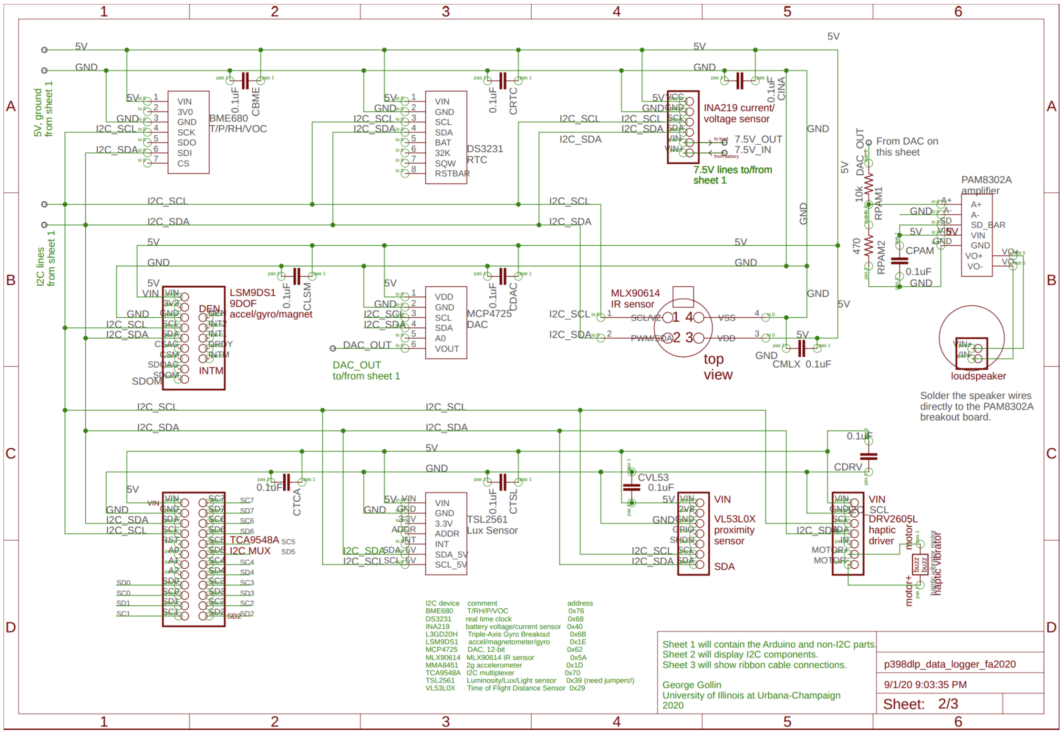
V. the liquid crystal display (LCD), with a 10kohm trimmer potentiometer, which displays information from data acquisition code,

VI. the 4 × 3 keypad, which allows users to make instructions on Arduino such as collecting data, starting and stopping recording,

VII. the Micro-SD breakout board, which stores recorded binary files and text files containing data from GPS and RTC.

Fig 3: Data logger schematic. Sheet1 contains Arduino and non-I2C parts, and sheet2 contains I2C components. Sheet 3 shows ribbon cable connections.




5V ground from sheet 1
I2C lines from sheet 1
BME680 T/P/RH/VOC
DS3231 RTC
INA219 current/voltage sensor
7.5V lines to/from sheet 1
From DAC on this sheet
PAM8302A amplifier
LSM9DS1 9DOF accel/gyro/magnet
MCP4725 DAC
MLX90614 IR sensor
top view
DAC_OUT to/from sheet 1
loudspeaker
Solder the speaker wires directly to the PAM8302A breakout board.
TCA9548A I2C MUX
TSL2561 Lux Sensor
VL53L0X proximity sensor
DRV2605L haptic driver
haptic vibrator
I2C device comment address
BME680 T/RH/P/VOC 0x76
DS3231 real time clock 0x68
INA219 battery voltage/current sensor 0x40
L3GD20H Triple-Axis Gyro Breakout 0x6B
LSM9DS1 accel/magnetometer/gyro 0x1E
MCP4725 DAC, 12-bit 0x62
MLX90614 MLX90614 IR sensor 0x5A
MMA8451 2g accelerometer 0x1D
TCA9548A I2C multiplexer 0x70
TSL2561 Luminosity/Lux/Light sensor 0x39 (need jumpers!)
VL53L0X Time of Flight Distance Sensor 0x29
Sheet 1 will contain the Arduino and non-I2C parts.
Sheet 2 will display I2C components.
Sheet 3 will show ribbon cable connections.
George Gollin
University of Illinois at Urbana-Champaign
2020
p398dlp_data_logger_fa2020
9/1/20 9:03:35 PM
Sheet: 2/3

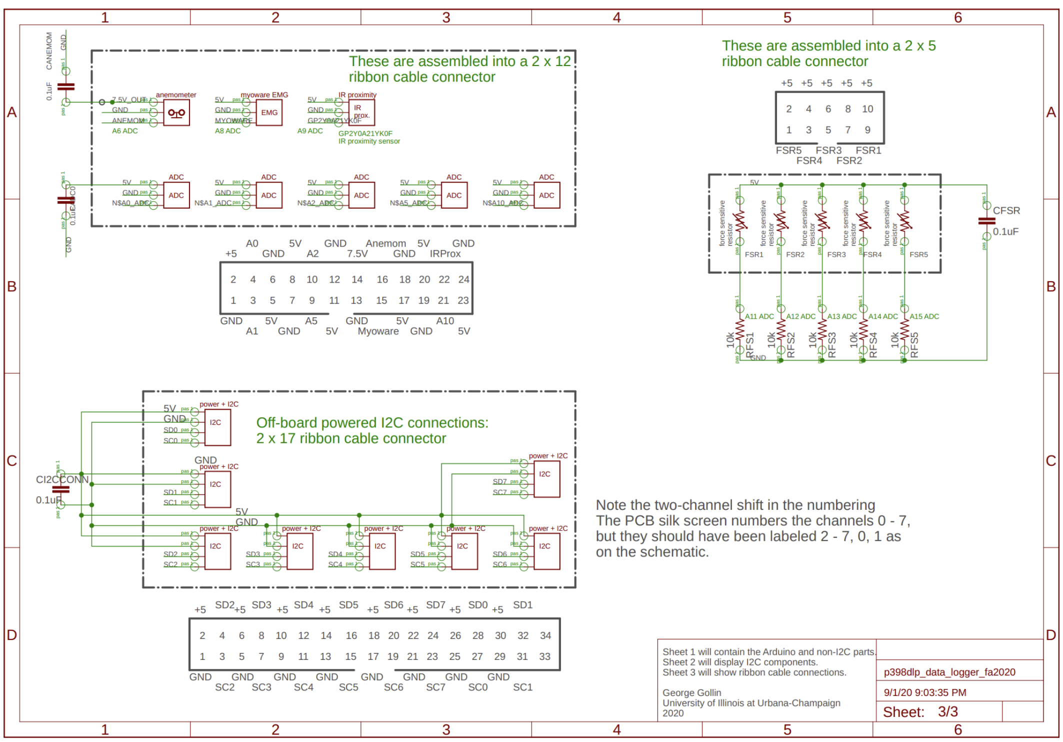

These are assembled into a 2 x 12 ribbon cable connector
anemometer
myoware EMG
IR proximity
GP2Y0A21YK0F IR proximity sensor
These are assembled into a 2 x 5 ribbon cable connector
FSR5 FSR3 FSR1
FSR4 FSR2
CFSR
0.1uF
Off-board powered I2C connections: 2 x 17 ribbon cable connector
CI2CCONN
Note the two-channel shift in the numbering
The PCB silk screen numbers the channels 0 - 7,
but they should have been labeled 2 - 7, 0, 1 as
on the schematic.
Sheet 1 will contain the Arduino and non-I2C parts.
Sheet 2 will display I2C components.
Sheet 3 will show ribbon cable connections.
George Gollin
University of Illinois at Urbana-Champaign
2020
p398dlp_data_logger_fa2020
9/1/20 9:03:35 PM
Sheet: 3/3

Fig 4: PCB layout for the cell phone group

2. Software

2.1 Data Acquisition Program

Data acquisition program (DAQ) was written in the Arduino IDE software, and compiled and uploaded to Arduino Mega 2560. Based on resources from the course website, we organized our DAQ that can read inputs from the keypad as different commands, and respond to run corresponding functions.

Our DAQ works as follows:

- After receiving the command (#8) from the keyboard, our DAQ will write the time it receives the command into a file. It will also collect the location from GPS, and write them into files.
- After writing the time into a file, it will take a while before the microphone starts recording, thus our DAQ will record that duration (tCPU). The duration is measured by the CPU internal clock. Because the CPU ticks are not very stable, the DAQ will also measure the duration by GPS (tGPS), which gives us a more accurate time. The ratio of tGPS and tCPU can be used to calibrate the length of recording, which is also measured by CPU internal clock.
- The microphone will then start recording and stops recording when we press "*" on the keyboard.

2.2 Data Analysis Code

In addition to DAQ, we had an additional analysis code file that used python as the primary programming language and implemented functionalities from Scipy, Numpy, and Matplotlib libraries. The analysis code tries to find the time shift between the time when the sender hears the message and the time when the receiver hears the message. In order to obtain the precise latency between the sender and the receiver, it's important to analyze the recording start time and the exact shift from the data we acquire. Therefore, besides storing the actual recording from the microphone, we also save the timing information into separate meta datas when running the DAQ code on the Arduino board.

In the analysis code file, we have

- two functions that calculate the exact starting time of the recordings of the sender and the receiver. They are
  - getAudioMicros - parses a file that contains the GPS time information and returns the starting time of the recording, and
  - adjustMicros - parses an additional file and returns the needed time adjustment due to tiny inconsistencies between the GPS and two different Arduino boards.
- The plot_shift function - which is the core function that calculates the chi-square between the sender and the receiver. It takes the minimum and maximum bin shift as input and

outputs the averaged chi-square value for each bin shift in that range. Note that we have to make sure the sample we need does not exceed the boundary of the inputs. This function returns the averaged chi-square value at each bin shift. The concept “bin” and “bin shift” will be explained later in 4.1 and 4.2.

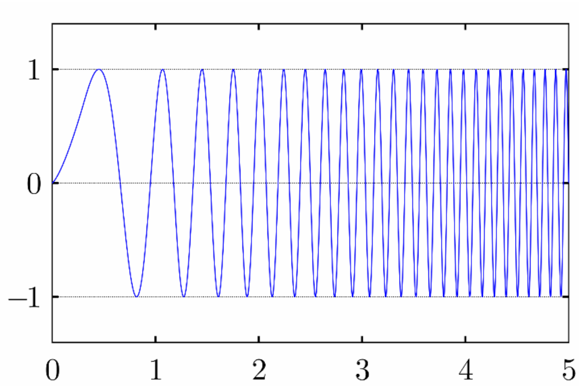
- The chi_square function - which is the main function we call on each test. It takes sender and receiver sample data and relevant timing information, and it outputs the calculated time latency between the sender and receiver. It first tries to align the input and output to the same starting point with reasonable alignment, and then it recursively searches for bin shifts from large ranges to small ranges. By doing this search recursively, we can easily locate bin shift value where the chi-square value is the lowest. In order to achieve the precision of magnitude less than one bin shift, the analysis does a polynomial regression fitting at the last step where we have already located the minimum bin shift within a relatively small range. It then tries to find the minimum point on the approximated curve and mark the lowest point as the actual bin shift.

- The s_drift and r_drift values - represent the drifting values of the CPUs on the sender and the receiver Arduino boards. This information is needed for precise measurements since different Arduino boards can have different cpu clock frequencies.

- The microStart function - takes all the relevant bin shift and timing information and outputs the latency between the sender and receiver in milliseconds.

3. Choice of Sound Source - Chirp Signal

After some trials and explorations, we decided to use the chirp signals as our final sound source because of its unique characteristics. Our selected chirp was a sinusoidal wave that linearly increased in frequency over time, which had the waveform as



$$x(t) = sin\left[\theta_0 + 2\pi\left(\frac{\gamma}{2}t^2 + f_0 t\right)\right] \quad (1)$$

$$\gamma = \frac{f_1 - f_0}{T} \quad (2)$$

where θ_0 is the initial phase at $t = 0$, f_0 and f_1 are the start and stop frequency, $T$is the sweeping time, therefore γ is the chirpiness constant.

Fig 5 (1). Chirp Signal: Amplitude vs time.

As the frequency of a chirp signal kept changing, we could synchronize two recorded signals by comparing the frequency patterns and locating points with minimal differences by chi-square functions. Further details are shown in the analysis part of this report.

We generated chirp signals as .WAV files from an online website named Sweep Tone Generator [8], where we could adjust parameters of the signal including the start and stop frequency, dBFS level, duration time, and sample rate. Here dBFS is Decibels relative to full scale, which means the number of Decibels below the maximum possible digital amplitude level.

When choosing sound, we found that if the frequency range is too large or the frequency is too high or too low, signals the speaker generates will have an obvious change in amplitude. Therefore we choose the frequency range from 800 to 900Hz.

Table 1. Characteristics of signal samples for data collection

Source	Start Frequency (Hz)	Stop Frequency (Hz)	Sample Rate (kHz)	Duration (s)	Level (dBFS)
Chirp	800	900	44.1k	2	0

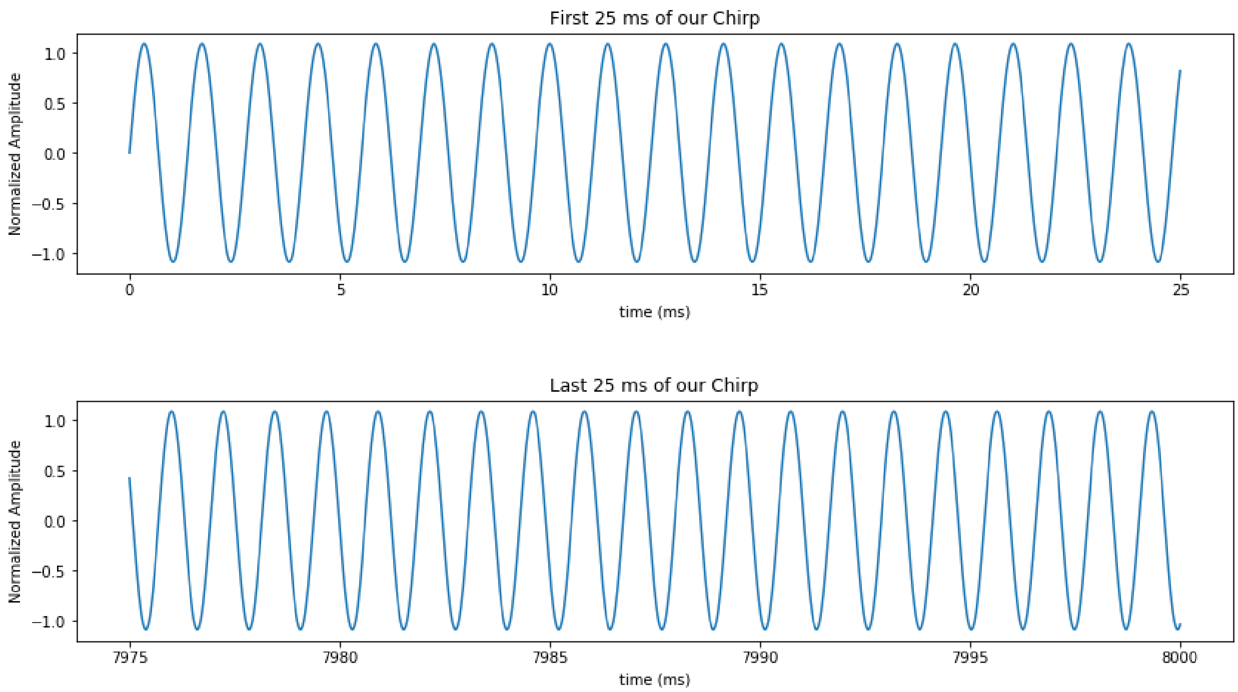


Fig 5 (2). The first 25 millisecond and last 25 millisecond of our Chirp signal.

4. Some Definitions & Mathematics

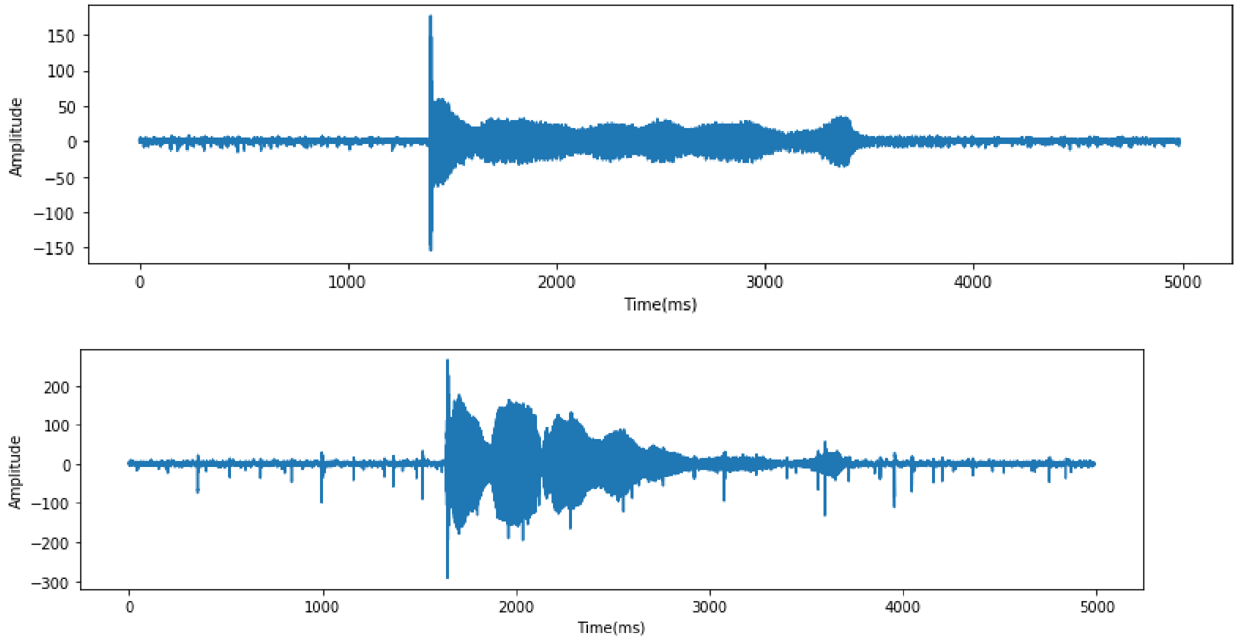
4.1 Sample Rate & Bin

Sample rate refers to the number of samples we take in a certain time. For example, if the sample rate of our recording is 40KHz, it means that the microphone takes a sample every 25 microseconds. Each sample is a number that describes the amplitude of sound, we will call a sample a "bin" later. Our recording will be stored in a binary file (bin file), when analyzing, we will convert the bin file to a (Python) numpy array. The length of the array is thus the total number of samples the microphone took when recording, each element in that array is the amplitude of sound at the time the sample was taken.

4.2 Minimize Chi Square & Convert Bins Shifted to Corresponding Time

The direct plot of arrays look like the following

Fig 6. Example: Direct plot of sender envelope (top), direct plot of receiver envelope (bottom).



We will use the chi-squared test (will be mentioned in 4.3) to determine how well the sender and receiver's envelopes match. To minimize the chi square, we first need to crop the sender and receiver data so that only the area with signals will be taken. For the plot above we choose bins from 50000 (Sender Start Bin) to 150000, for the plot below we choose bins from 60000 (Receiver Start Bin) to 160000, then we will get a plot that looks like the following.

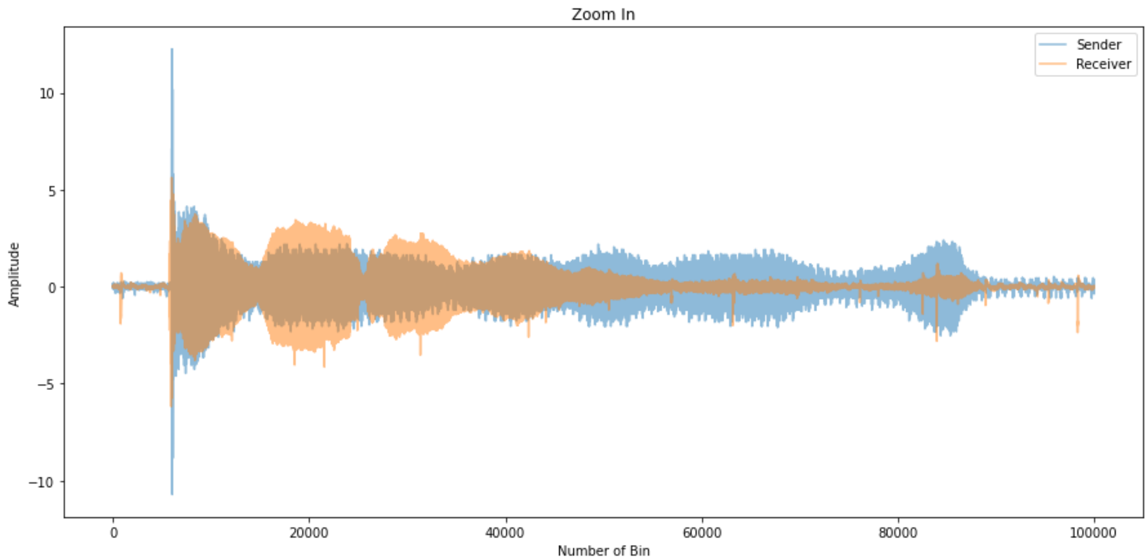


Fig 7. Example: Put sender and receiver data together after cropping.

To further minimize the chi square, we will move the "receiver plot" around along the x-axis, and the number of bins shifted corresponding to the smallest chi square is the "Bin Shift" we desired.

To convert "Sender Start Bin", "Receiver Start Bin", and "Bin Shift" to corresponding time, we wrote a python program. The time duration between bins is controlled by the CPU clock, as CPU ticks are not uniform enough, we need to calibrate it. As the tCPU and tGPS are recorded in each test (refer to page7 for more details), we can also calculate the time calibration index for sender (sCalibrationIndex = tGPS_Sender/tCPU_Sender), and time calibration index for receiver (rCalibrationIndex = tGPS_Receiver/tCPU_Receiver).

Assume our recordings start at the same time, then the corresponding time (latency); given "Sender Start Bin (SSB)", "Receiver Start Bin (RSB)", "Bin Shift (BS)", "sCalibrationIndex (SCI)", "rCalibrationIndex (RCI)", and the Sample Rate; is

$$Time = (RSB \times RCI - SSB \times SCI + BS \times RCI)/Sample\ Rate. \tag{3}$$

4.3 Chi Squared Test

We utilized a chi-squared test to compare the binary data from sender's and receiver's recordings. The latency will be calculated by locating the decimal bin number with minimal chi-square value. A chi-squared test, also written as χ2 test, is a statistical hypothesis test that is used to measure the match of two statistical distributions. Our formula of chi square is

$$\chi 2 = \frac{1}{N} \sum_{i=1}^{N} (S_i - R_i)^2 \tag{4}$$

Here R_i is the normalized receiver's recording; S_i is the normalized sender's recording, N is the number of bins in our cropped area. N in the denominator is used to normalize the chi-square value.

The precision of our chi square test is the number of bins we need to shift around "Bin Shift" (the same as "BS" in formula (3)) in order to increase the minimum chi square value by one. Assume the minimum chi square is M, "Bin Shift" that corresponding to the minimum chi square is x_0,the quadratic fitting near minimum chi square is $f(x) = ax^2 + bx + c$. We have the precision (in second) of chi square test

$$\sigma = \pm \left| \frac{-b+\sqrt{b^2-4a(c-M-1)}}{2a} - x_0 \right| \div Sample\ Rate \ . \qquad (5)$$

Assume the name of our numpy array is "Data". In our analysis, we will first subtract every element in the array by "mean of the array" (Data - np.mean(Data)), this will recenter the data to have zero mean value and simplify further calculation. Then we will normalize our data by dividing every element in the array by the "standard deviation of the array" (Data/np.std(Data)). Finally we will calculate the chi square by formula (4), get the function between the chi-square value and number of bins the receiver's envelope was shifted along the x-axis, and find the minimum point of this function.

4.4 Correlation

The correlation test is a measure of similarity of two series data, and it's also known as the dot product between two signals. The formula of correlation can be written as:

$$\text{correlation} = \frac{\sum_{i=1}^{N} S_i * R_i}{N}. \qquad (6)$$

Here R_i is the normalized receiver's recording, S_i is the normalized sender's recording, and N is the total number of data points in both sender's signal and receiver's signal. Similar to the chi-square test, we first subtract every element in the signals by its mean and then divide every element by its standard deviation to normalize the sender and receiver signals. This normalization step is necessary, otherwise the difference of the voice output volumes on sender's and receiver's sides will cause the correlation test to fail. We used the correlation test to check the results from the chi-square test.

Experimental Procedure

1. Calibration

The methodology of calibration is to place two PCBs a certain distance away, then use our code to record and calculate the distance between the two PCBs. By comparing the calculated L with actual L, we can know how accurate our DAQ and python analysis code are.

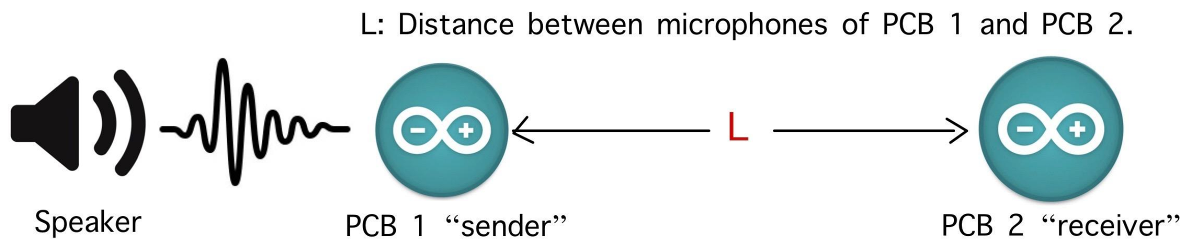


Fig 8. Instrumental Layout of Calibration

1.1 Time Accuracy of DAQ

When measuring the time accuracy of DAQ, to avoid the error from chi square analysis, we used a sound whose envelope looks like a delta function (the best sound source we could find was hand-clapping). Then we compared the time difference between the first peaks of the two envelopes. Here is an example:

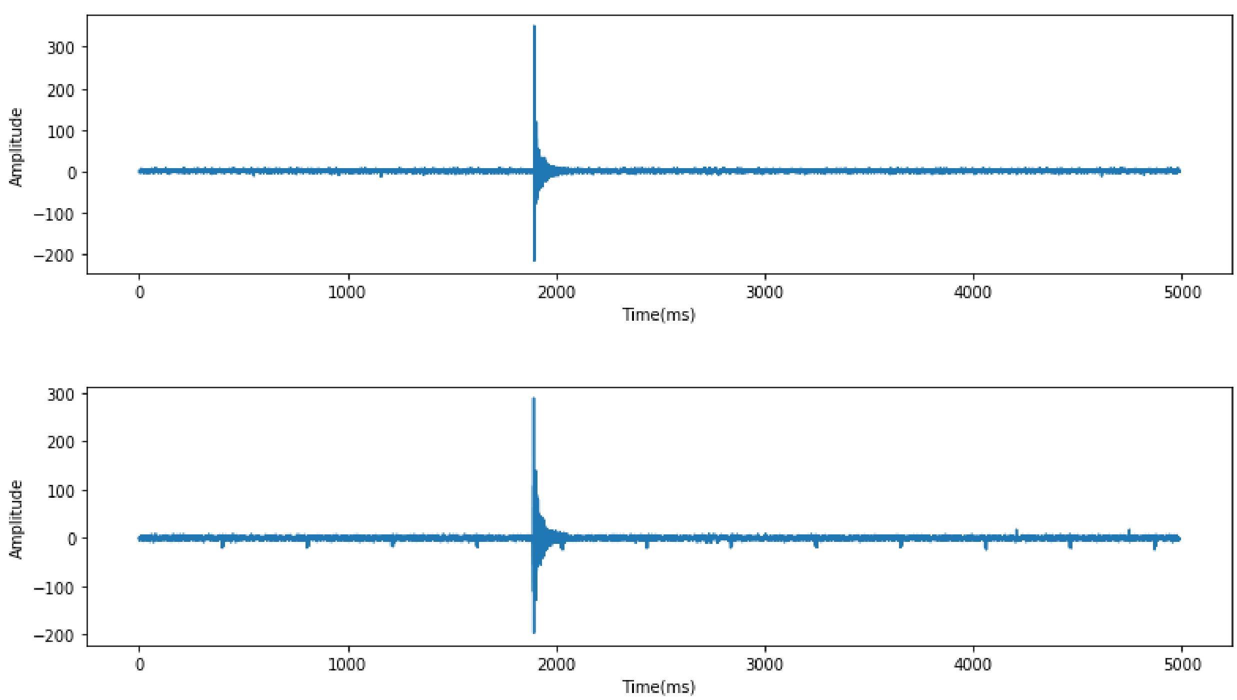


Fig 9: Overall envelopes of recorded binary files from PCB1 (top) and PCB2 (bottom) separately. The sound source is hand clapping.

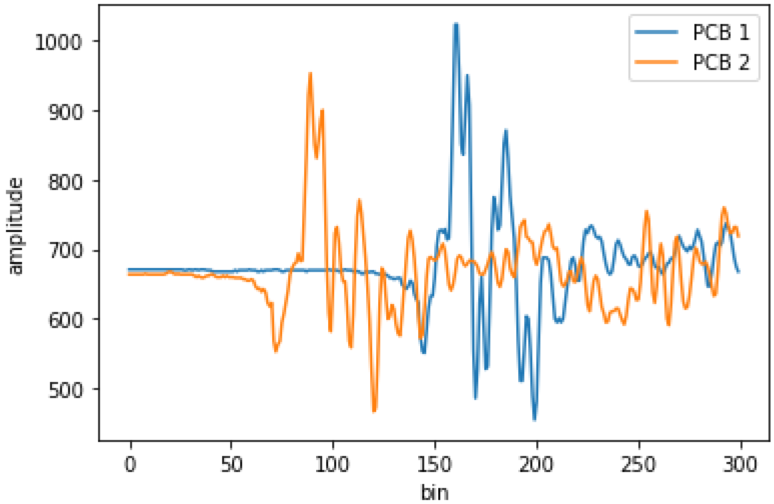


Fig 10. Zoomed in plot of PCB 1's and PCB 2's recordings. Both are cropped so that start from bin = 75700 and have a length = 300 bins.

The two PCBs were put apart so that L = 15.00 centimeters (see fig. 8). We measured three times and yielded an average calculated distance of 14.80 cm, and a range of only 0.02 cm. Which suggested that the time accuracy of our DAQ is about 6 microseconds.

Table 2. Calibration of DAQ, Actual Distance = 15.00 cm

	Test #1	Test #2	Test #3	Mean (cm)	Range (cm)
Calculated Distance (cm)	14.80	14.80	14.78	14.80	0.02
Error (%)	1.3%	1.4%	1.5%	1.4%	0.1%

1.2 Accuracy of DAQ & Python Analysis Code

To determine the accuracy of all our codes, we did five sets of measurement using our Chirp signals. In these tests, we arrange the two PCBs so that L = 6.6cm, 9.8cm, 10.0cm, 13.0cm, and 13.1 cm (refer to Fig. 8).

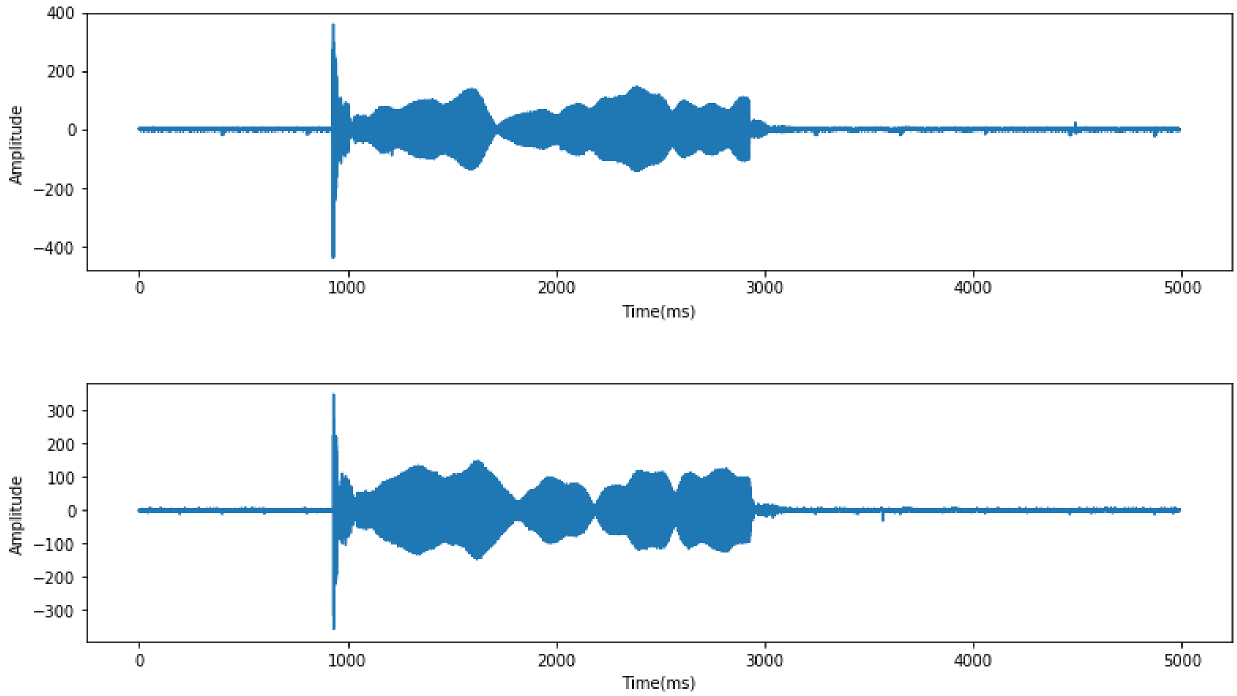


Fig 11. Overall envelopes of recorded binary files from PCB1 (top) and PCB2 (bottom).

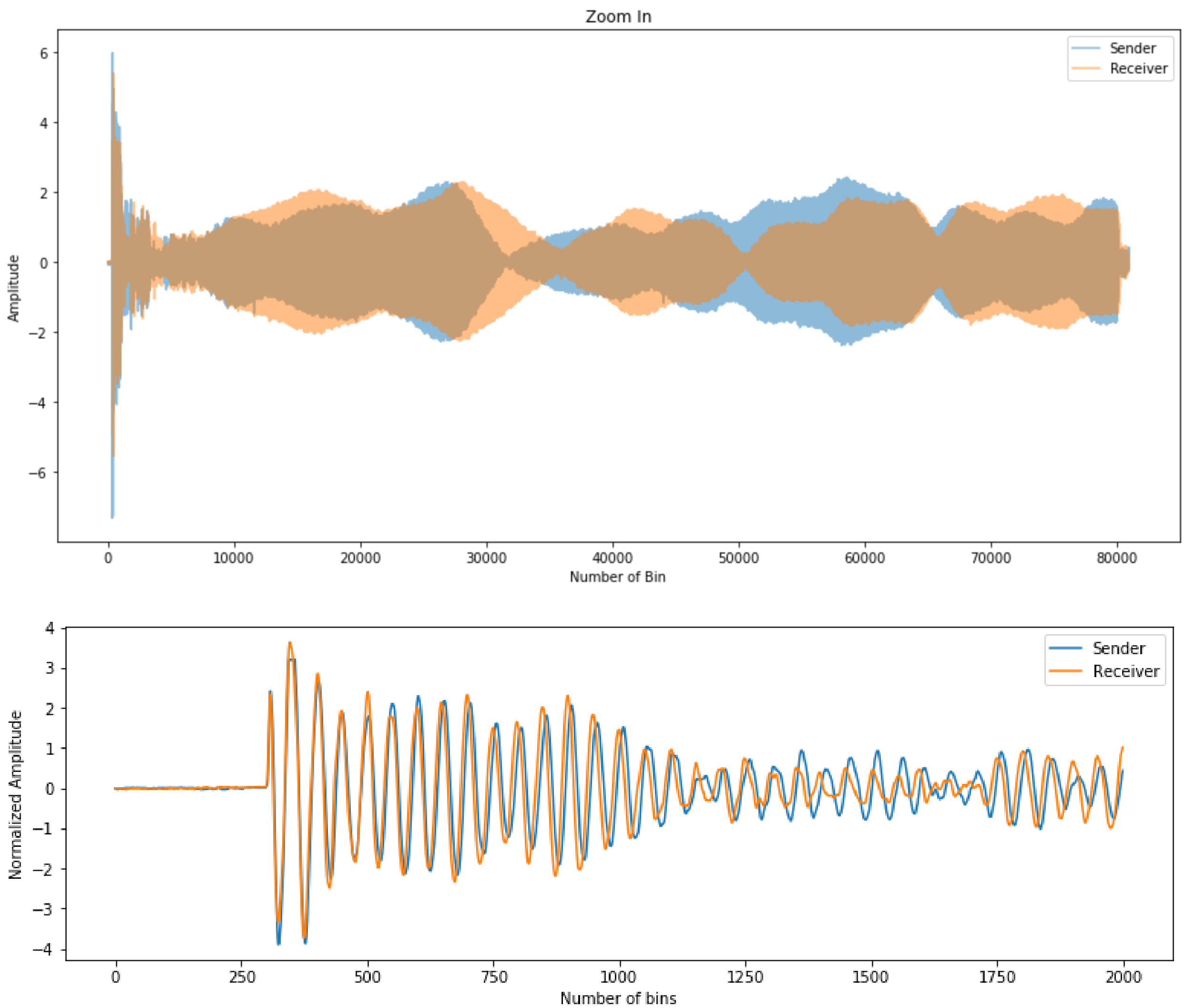


Fig 12. (Top) Comparing the zoomed in signal patterns (after cropping) between PCB1's recording (blue) and PCB2's recording (orange). Both are cropped so that start from bin = 37000 and have a length = 81000 bins. (Bottom) Zoomed in plot of PCB1's and PCB2's envelopes after chi-square and correlation match. PCB1's envelope starts from bin number = 37000, PCB2's envelope starts from bin number = 37052, both have a length of 2000 bins.

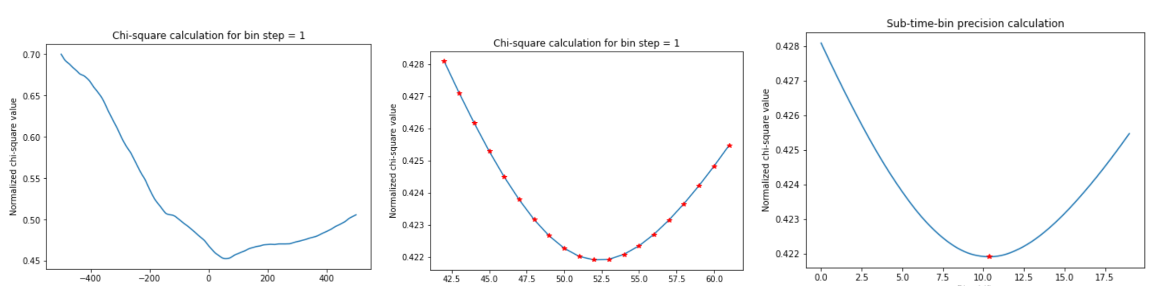


Fig 13: Move the PCB2's cropped envelope from -500 bins to 500 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left).

Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For x-axes, bin number = 42 in the middle plot corresponds to bin number = 0 in the right plot. For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

For the first measurement, the decimal bin shift corresponding to minimum chi square is 52.37. And the corresponding time delay is 228.58 microseconds. Thus, the calculated distance is 7.84cm.

Here are all the results we get for five sets of tests.

Table 3. Calibration of DAQ & Analysis Code

	Test #1	Test #2	Test #3	Test #4	Test #5	Mean
Actual Distance (cm)	6.60	9.80	10.00	13.00	13.10	/
Calculated Distance (cm)	7.84	10.67	10.45	12.61	14.36	/
Absolute Error (cm)	1.24	0.87	0.45	0.39	1.26	0.84
Percentage Error (%)	18.8	8.88	4.50	3.00	9.62	8.96

Correlation (Actual, Calculated) = 0.969

Here we can see that the relative error of the first data set is significantly larger than the other two. But consider its actual distance, which is 6.6 cm, the percentage error is acceptable. And the average of absolute errors is about 0.84 cm, which suggests that our DAQ, along with our analysis code, has an precision of 2.5×10^{-5}second, which is good. The correlation mentioned below also suggests a good accuracy .

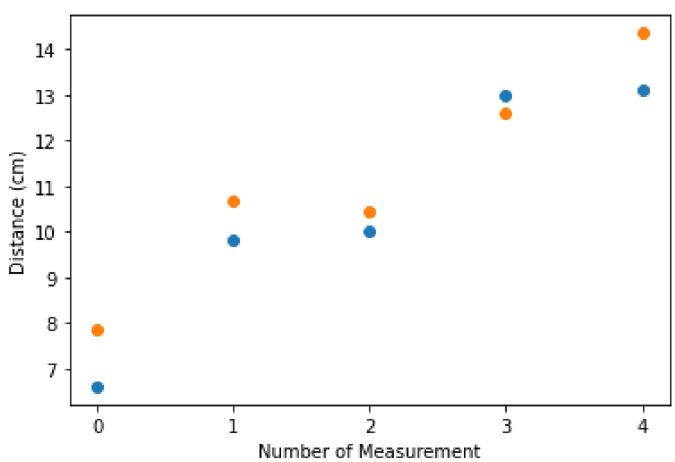


Fig. 14. Plot of actual distances and calculated distances. The correlation is very close to 1, which suggests that our calculations agree with the actual values. More straightforward, as we can see from the plot above, the trends of calculated and actual distances are similar.

2. Data collection

We did some preparations to synchronize the real time clocks on individual PCBs. The Arduino code provided by professor George Gollin set RTCs using GPS data from satellites, and we also checked the time displayed on the LCD to be the same referring to an assigned google clock in order to ensure the GPS worked well.

The first step was to make phone calls between two of the group members. Each group member placed their phone and PCB in front of the microphone of the computer in a controlled short distance.

1. At a scheduled time, we pressed the keypad to start recording. Though we could not guarantee the two devices would start recording at exactly the same time, this step could reduce the time differences within a short range and simplify further analysis work.
2. The selected chirp signal would be played by the computer on the side of the sender, and we would stop recording after the sound stops. There was no need to ensure the same stopping time.
3. And immediately after recording stopped, the data acquisition code automatically allowed Arduino to gather GPS and MET data and to write files into SD cards.

3. Analysis

The chi square is sensitive to irrelevant information. When doing analysis, we “crop” the recordings so that only the part with chirp signals is counted to calculate chi square.

Results and Discussion

1. Calls Under Different Types of Network

Here we present the latency study results for calls under different types of network. We performed three measurements under 4G network with VoLTE on at the midnight of Nov 11, 2020, three measurements under 4G network with VoLTE off at the midnight of Nov 11, 2020, three measurements under 3G network at the midnight of Nov 11, 2020, and three measurements under 3G at the midnight of Nov 10, 2020. The Chi-square test and correlation are also calculated with the data.

1.1. Calls Under 4G Network (VoLTE On) (At the Midnight of Nov 11, 2020)

Test 1.

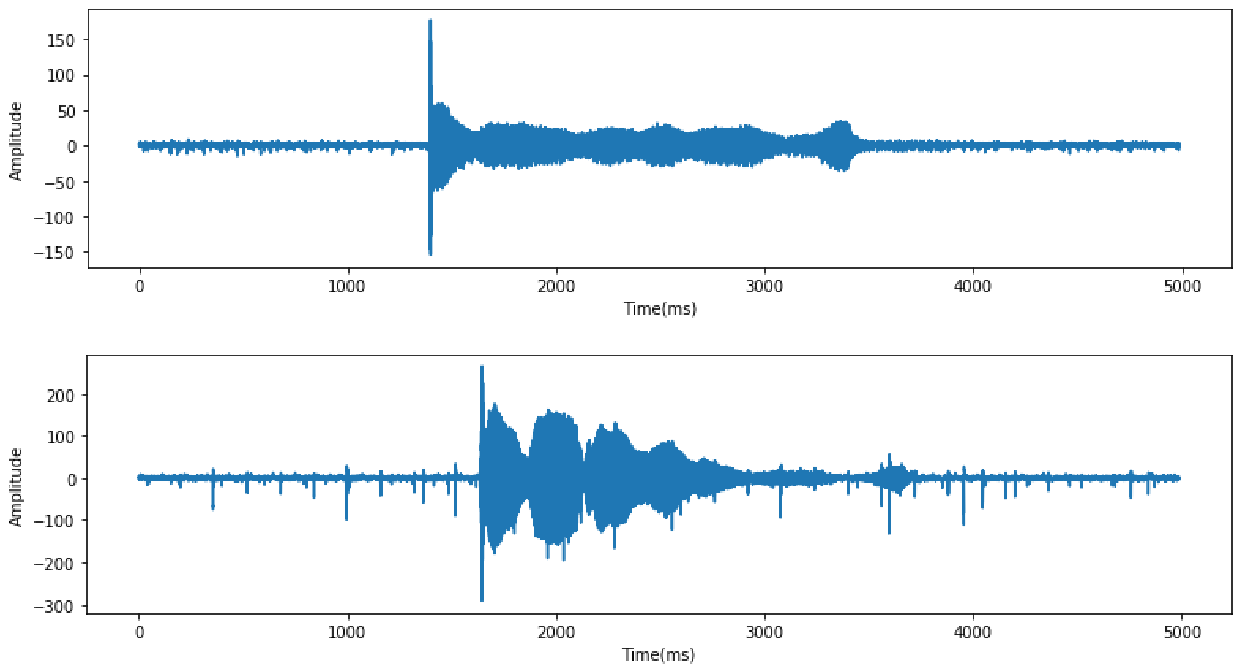


Fig 15: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

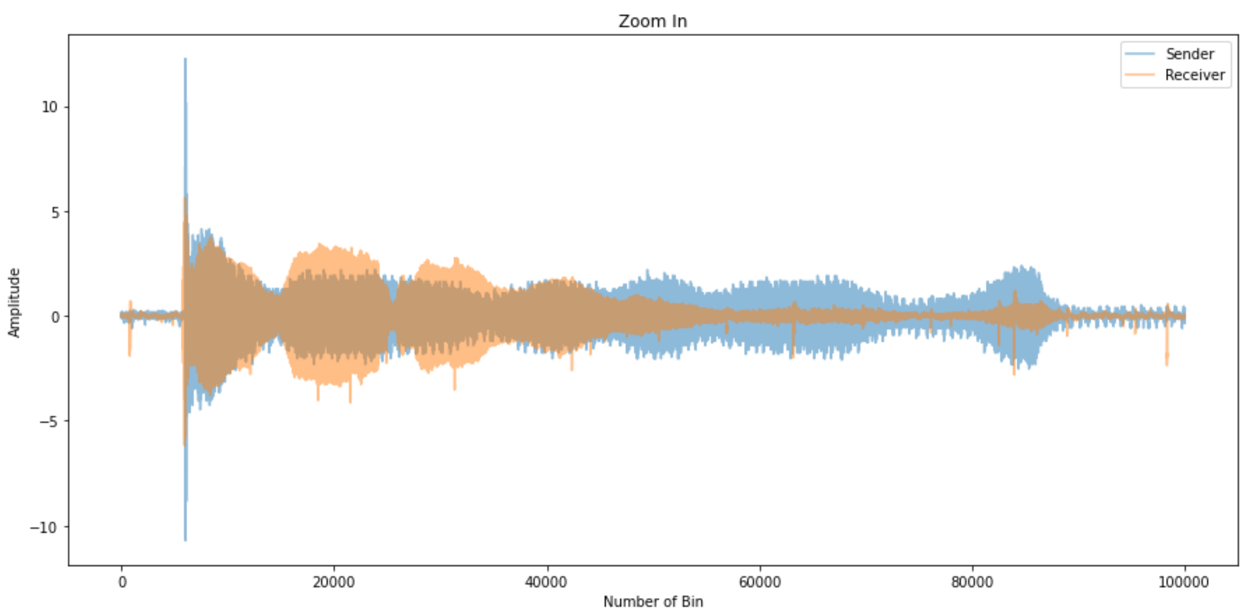


Fig 16: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 50000; the receiver's envelope shown starts from bin = 60000, both have a length of 100000 bins.

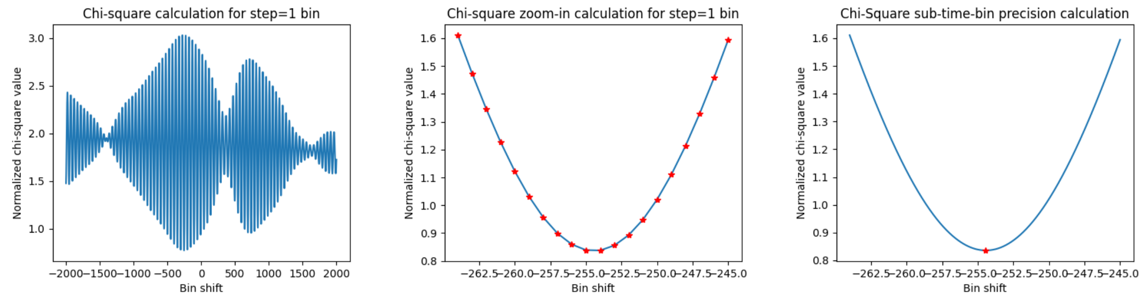


Fig 17: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

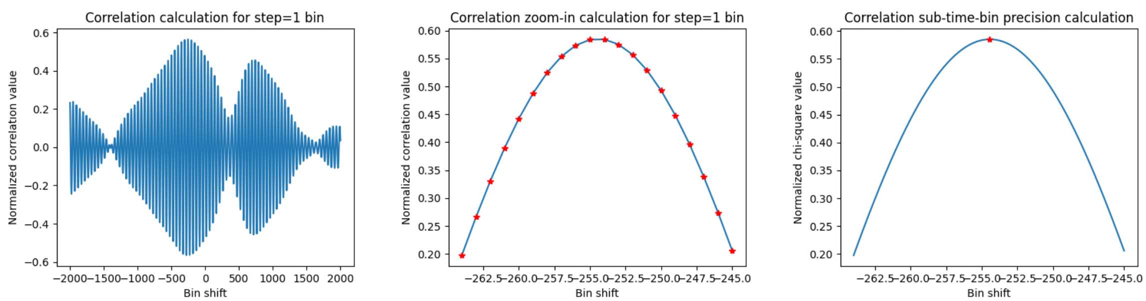


Fig 18: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of -254.44 with error ±0.0355. The corresponding latency is 243.77 milliseconds. The maximum correlation corresponds to a decimal bin shift of -254.44 with error ±0.0503. The corresponding latency is 243.77 milliseconds.

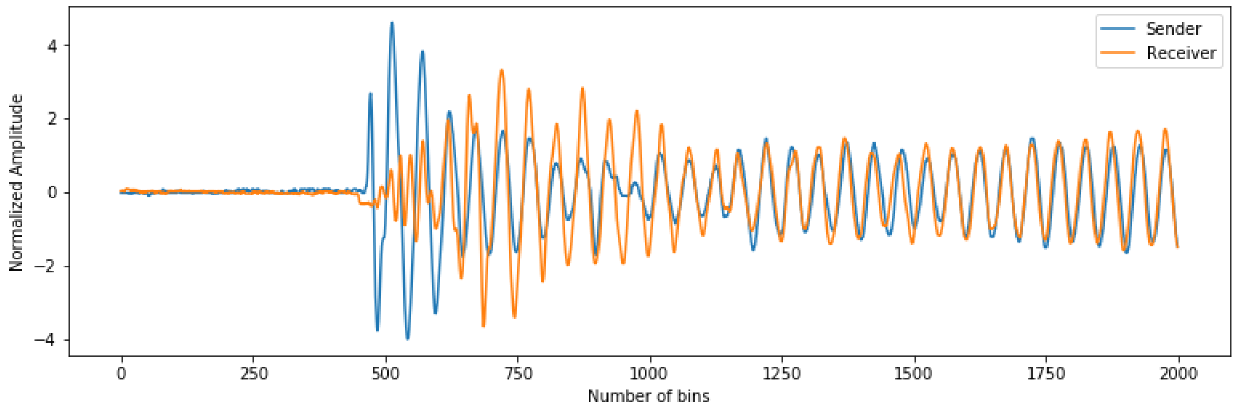


Fig 19: Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 55500, receiver envelope starts from bin number = 65246, both have a length of 2000 bins.

Test 2.

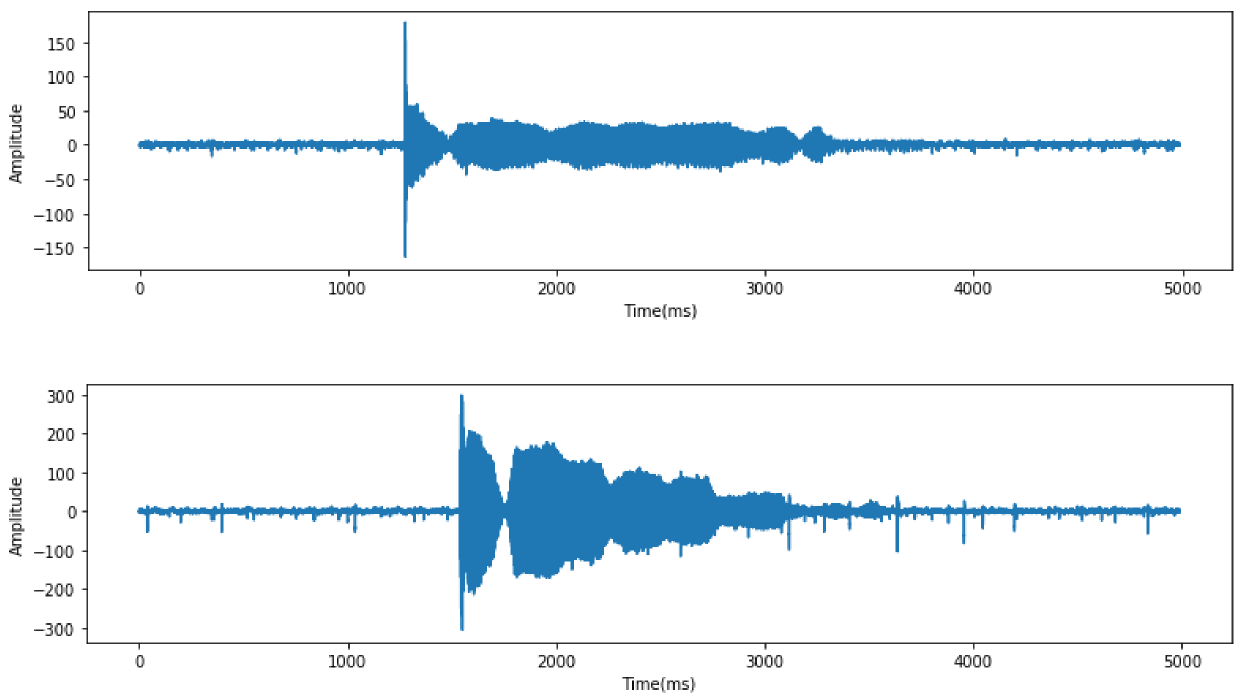


Fig 20: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

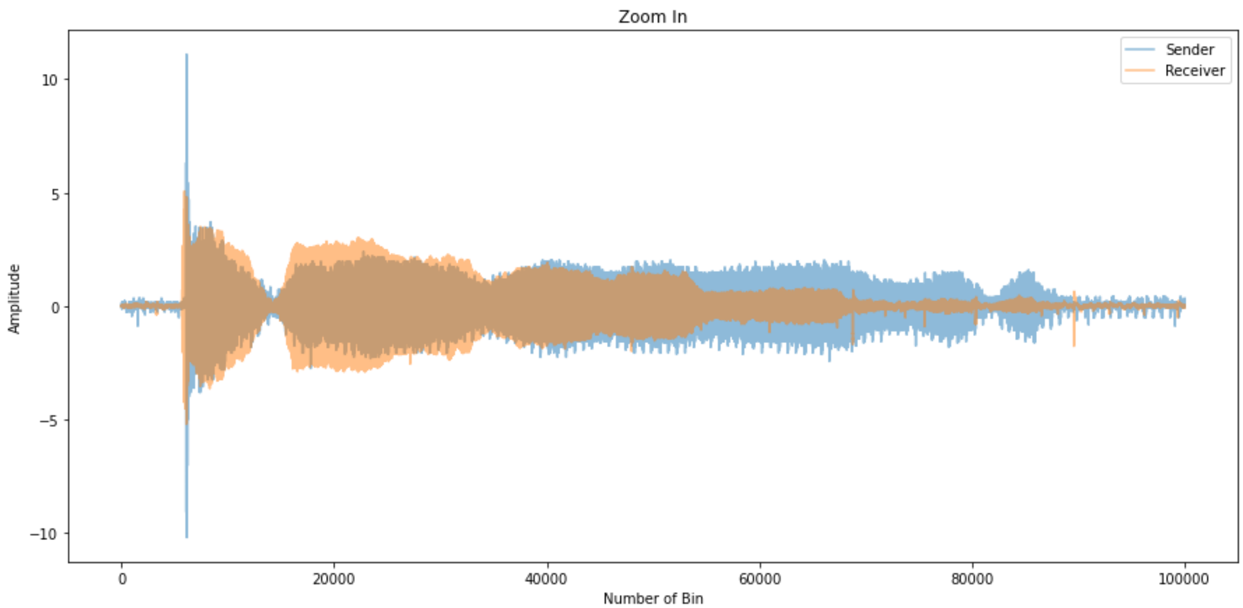


Fig 21: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 45000; the receiver's envelope shown starts from bin = 56000, both have a length of 100000 bins.

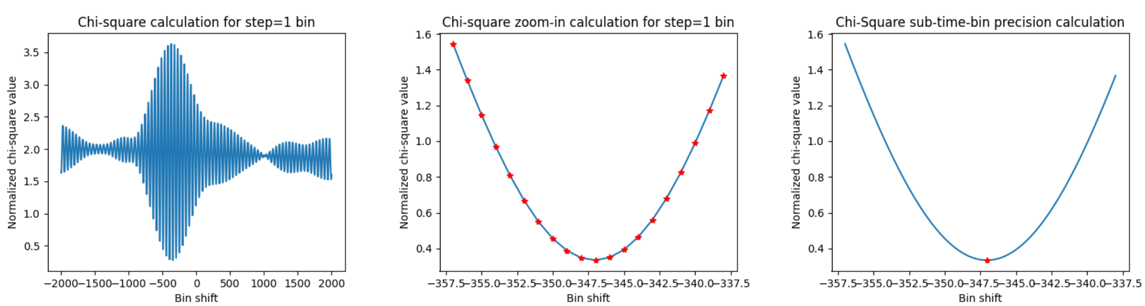


Fig 22: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

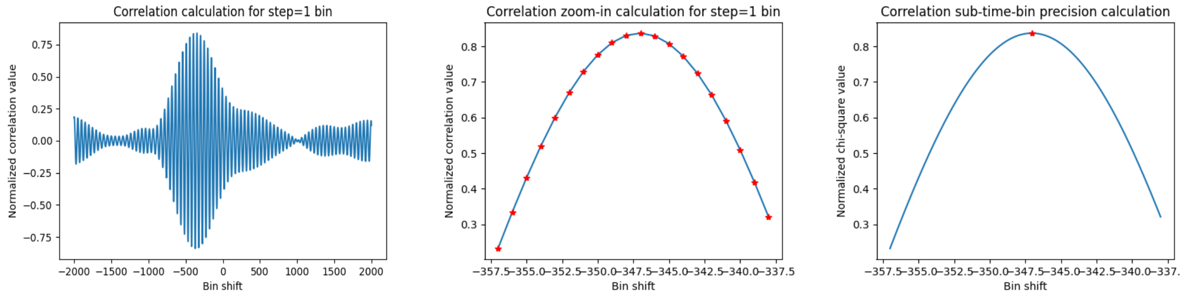


Fig 23: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of -347.05 with error ±0.0294. The corresponding latency is **266.05 milliseconds**. The maximum correlation corresponds to a decimal bin shift of -347.05 with error ±0.0416. The corresponding latency is **266.05 milliseconds**.

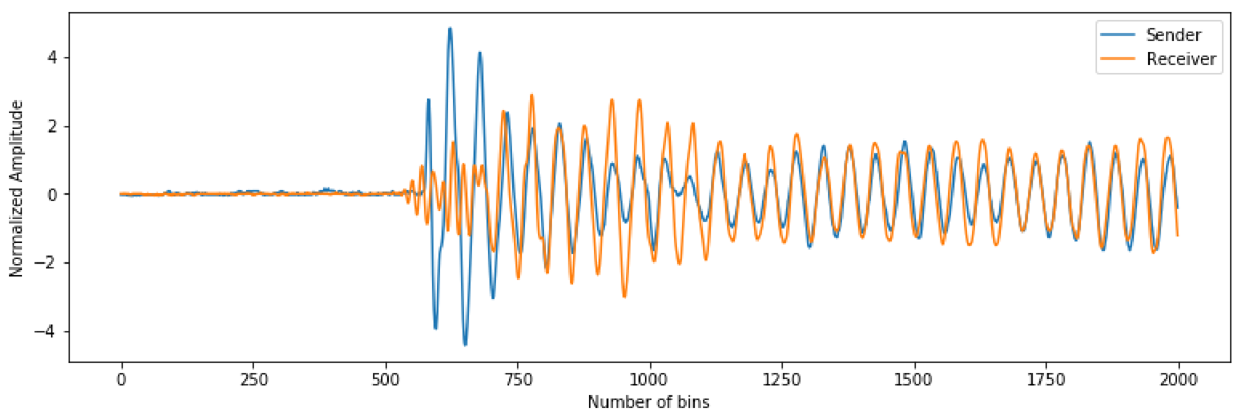


Fig 24. Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 50500, receiver envelope starts from bin number = 61153, both have a length of 2000 bins.

Test 3.

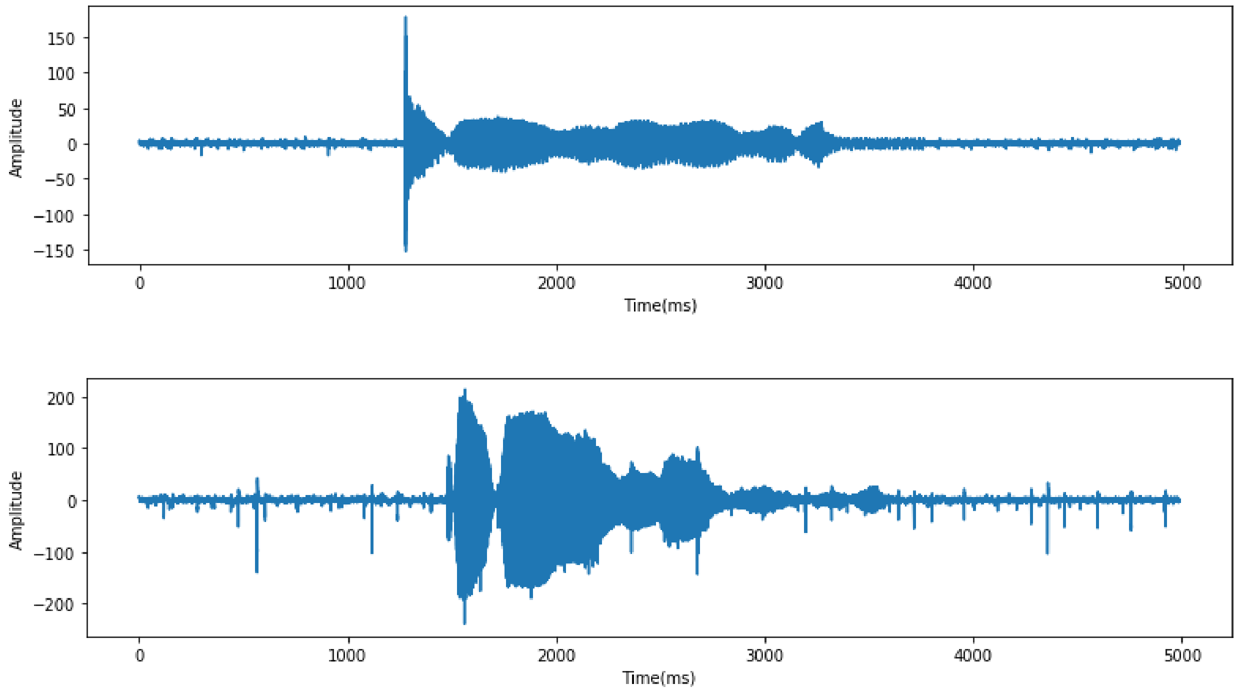


Fig 25: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

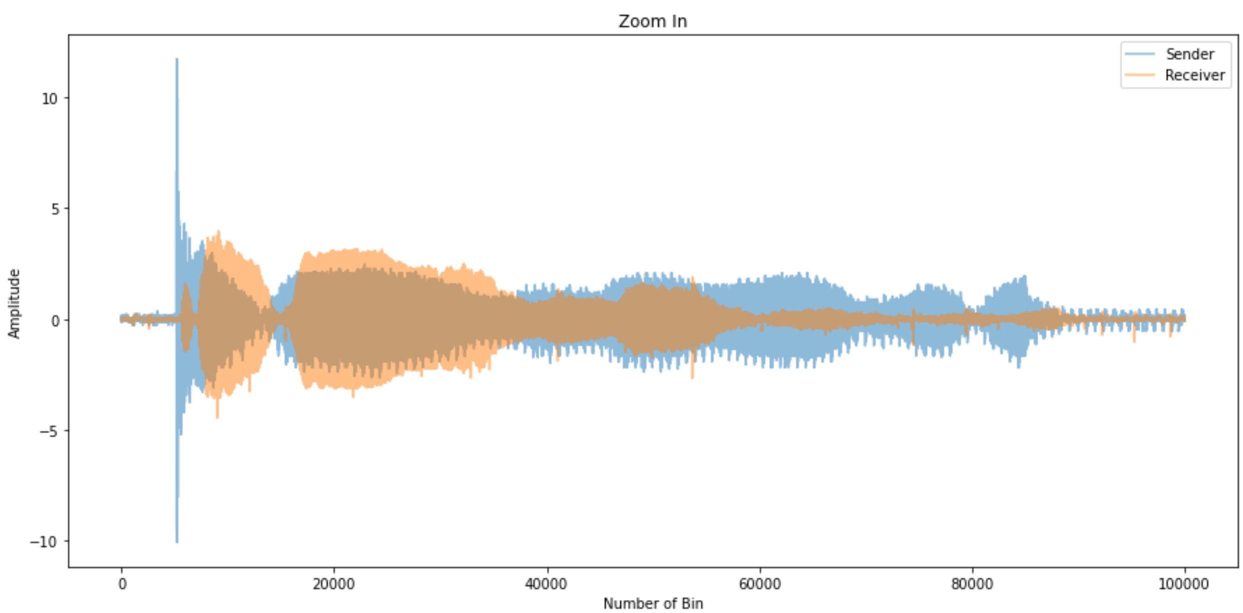


Fig 26: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 46000; the receiver's envelope shown starts from bin = 53500, both have a length of 100000 bins.

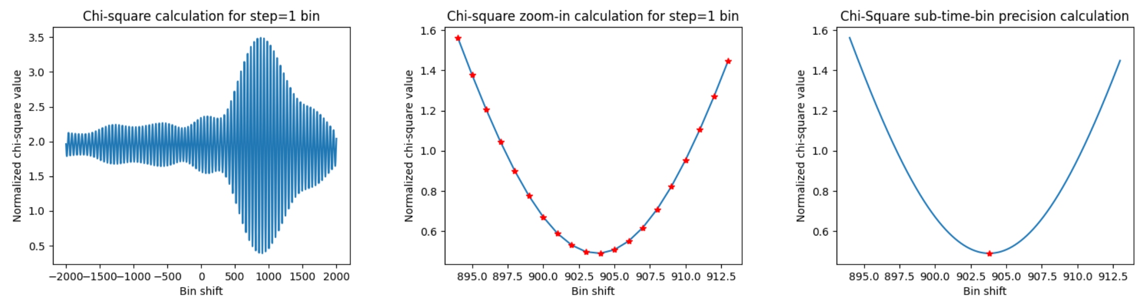


Fig 27: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

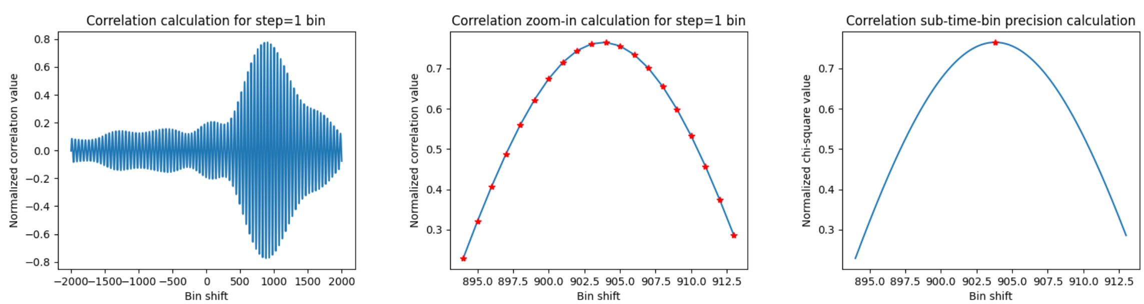


Fig 28: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of 903.8 with error ±0.0308. The corresponding latency is **222.68 milliseconds**. The maximum correlation corresponds to a decimal bin shift of 903.8 with error ±0.0436. The corresponding latency is **222.68 milliseconds**.

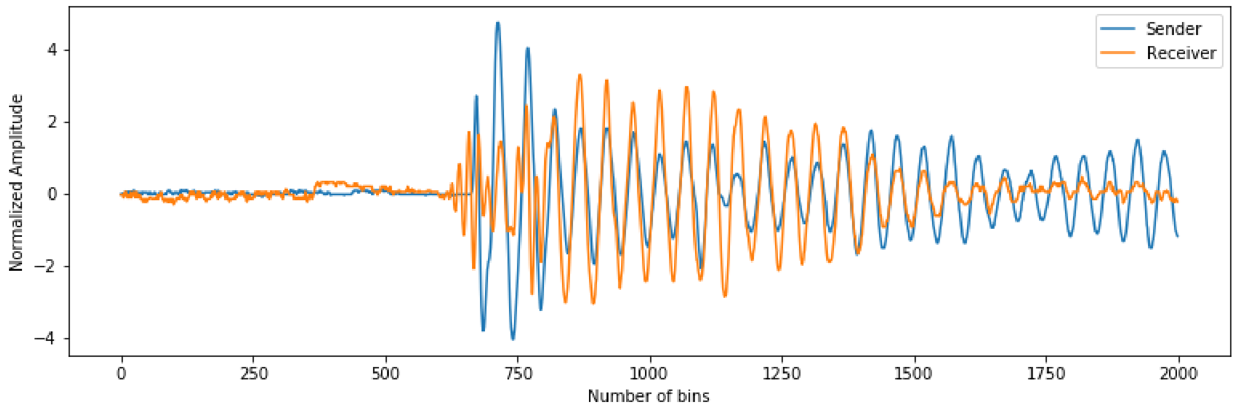


Fig 29. Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 50500, receiver envelope starts from bin number = 58904, both have a length of 2000 bins (bottom).

Table 4. Results of Latency Test 1, 2, 3 (4G, VoLTE On)

	Sender Calibration Index (SCI)	Receiver Calibration Index (RCI)	Sender Crop Range (1000 Bins)	Receiver Crop Range (1000 Bins)	Bin Shift At Min $\chi 2$/ Max correlation	Latency (ms)
Test 1.	1.00034	1.00021	50-150	60-160	-254.44	243.77
Test 2.	1.00033	1.00021	45-145	56-156	-347.05	266.45
Test 3.	1.00031	1.00021	46-146	53.5-153.5	903.8	222.68

Average precision of Chi square tests: 0.0319 bins = 0.798 microseconds.

1.2. Calls Under 4G Network (VoLTE Off) (At the Midnight of Nov 11, 2020)

Test 4.

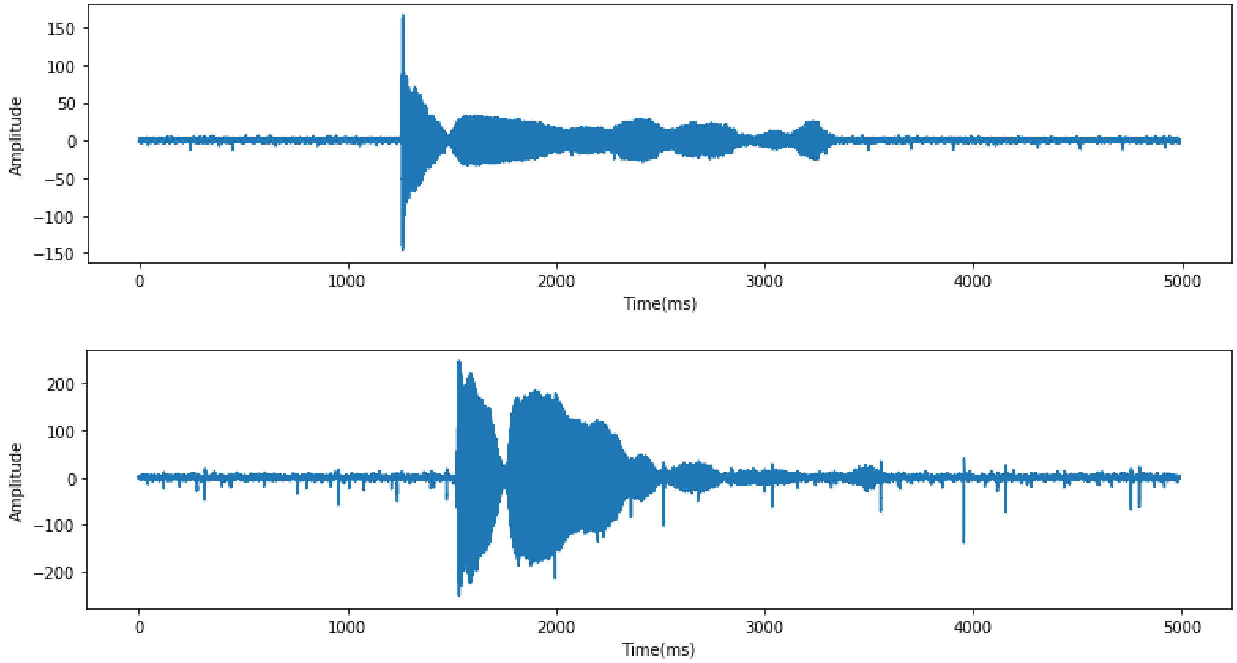


Fig 30: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

Fig 31: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 45000; the receiver's envelope shown starts from bin = 55000, both have a length of 100000 bins.

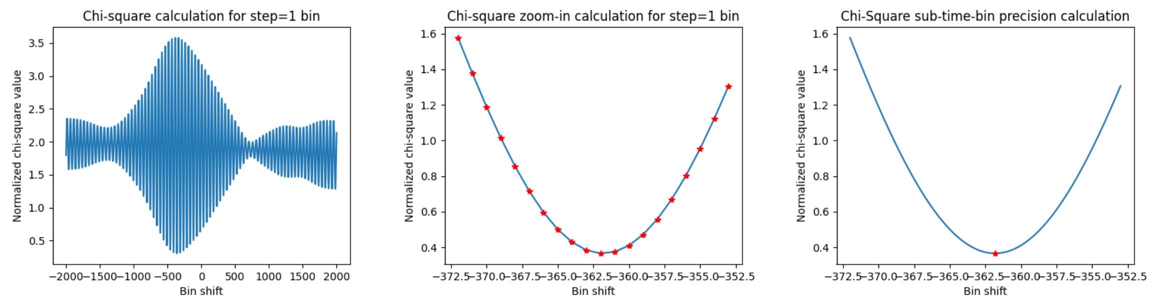


Fig 32: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

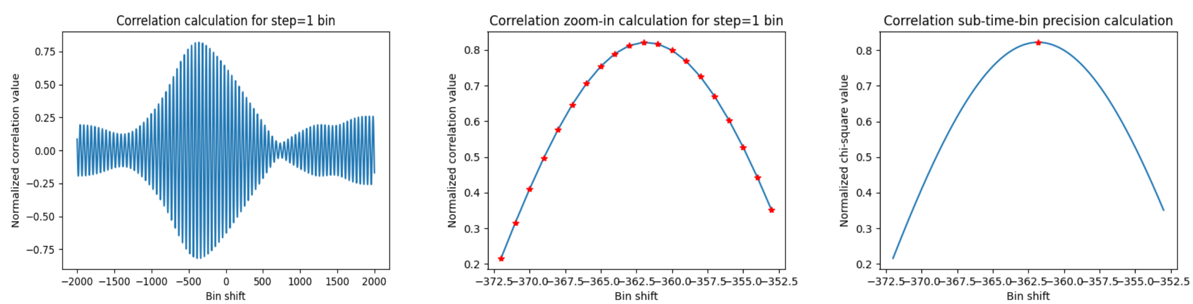


Fig 33: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of -361.82 with error ±0.0301. The corresponding latency is **266.01 milliseconds**. The maximum correlation corresponds to a decimal bin shift of -361.82 with error ±0.0425. The corresponding latency is **266.01 milliseconds**.

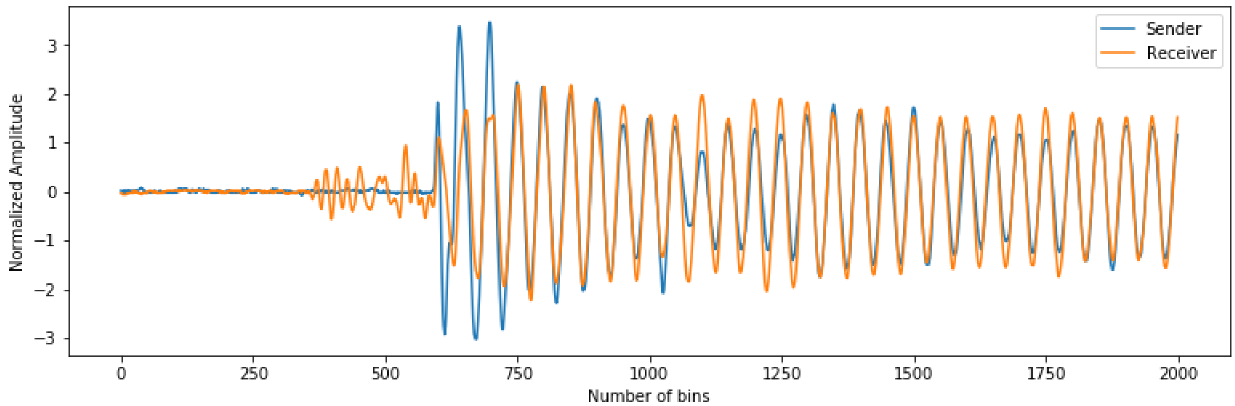
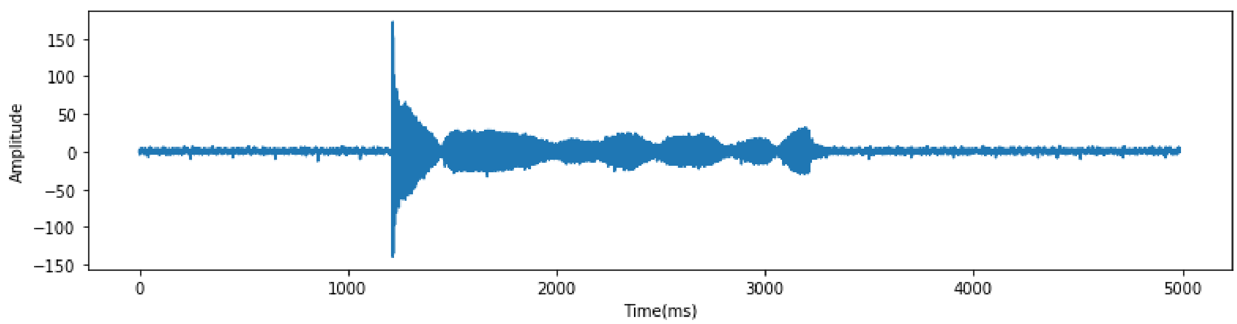


Fig 34. Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 50000, receiver envelope starts from bin number = 59638, both have a length of 2000 bins.

Test 5.



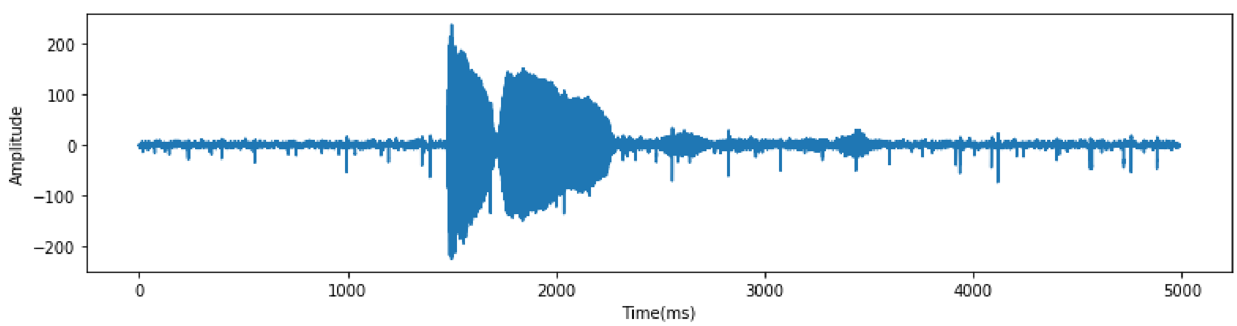


Fig 35: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

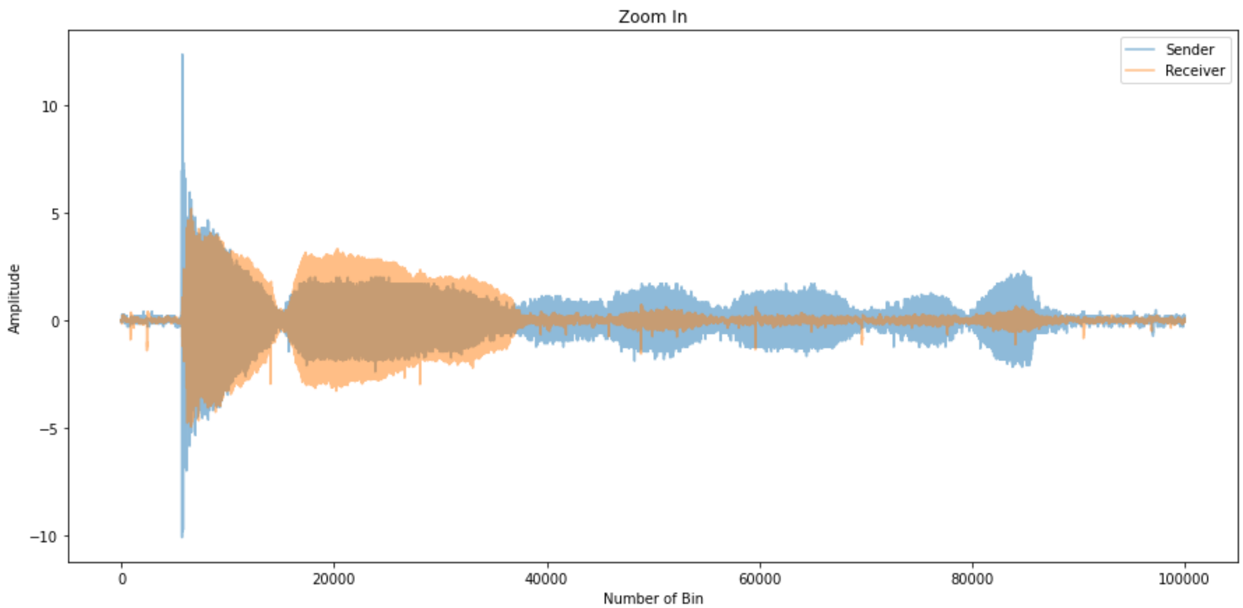


Fig 36: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 43000; the receiver's envelope shown starts from bin = 53500, both have a length of 100000 bins.

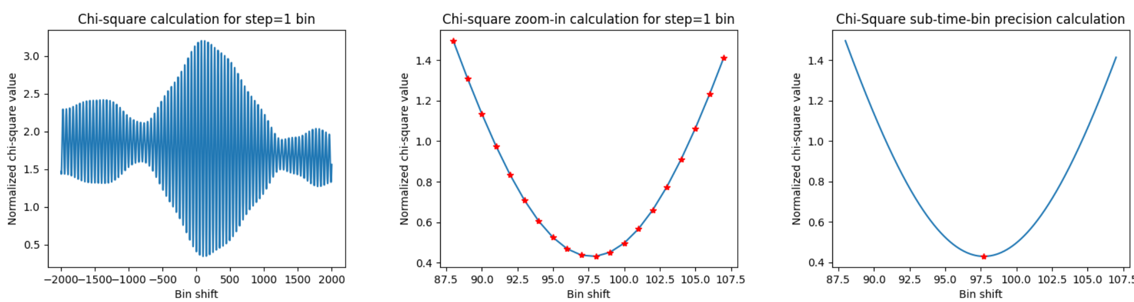


Fig 37: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

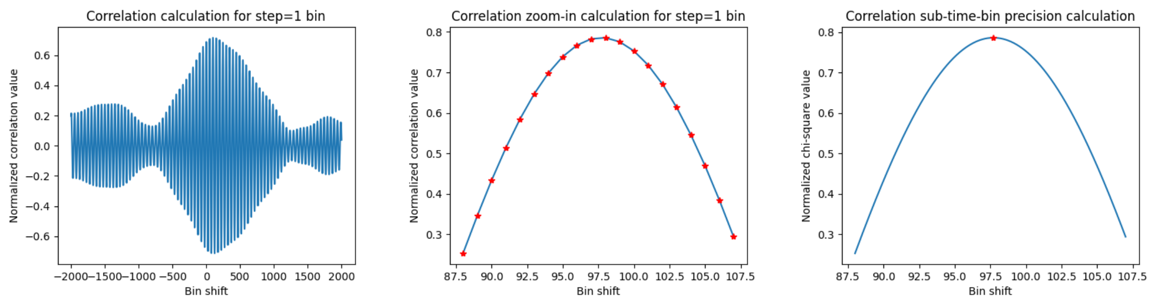


Fig 38: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of 97.72 with error ±0.0308. The corresponding latency is **264.99 milliseconds**. The maximum correlation corresponds to a decimal bin shift of 97.72 with error ±0.0435. The corresponding latency is **264.99 milliseconds**.

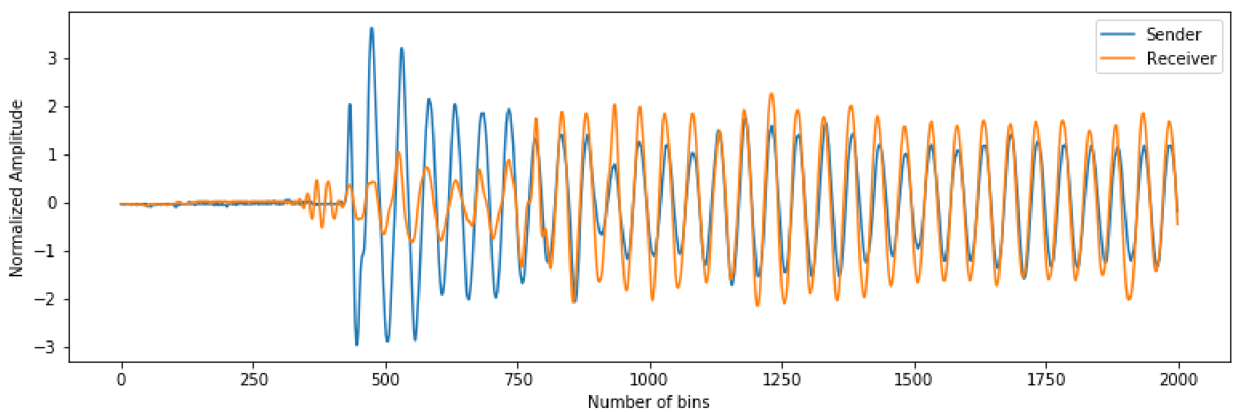


Fig 39. Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 48250, receiver envelope starts from bin number = 58848, both have a length of 2000 bins.

Test 6.

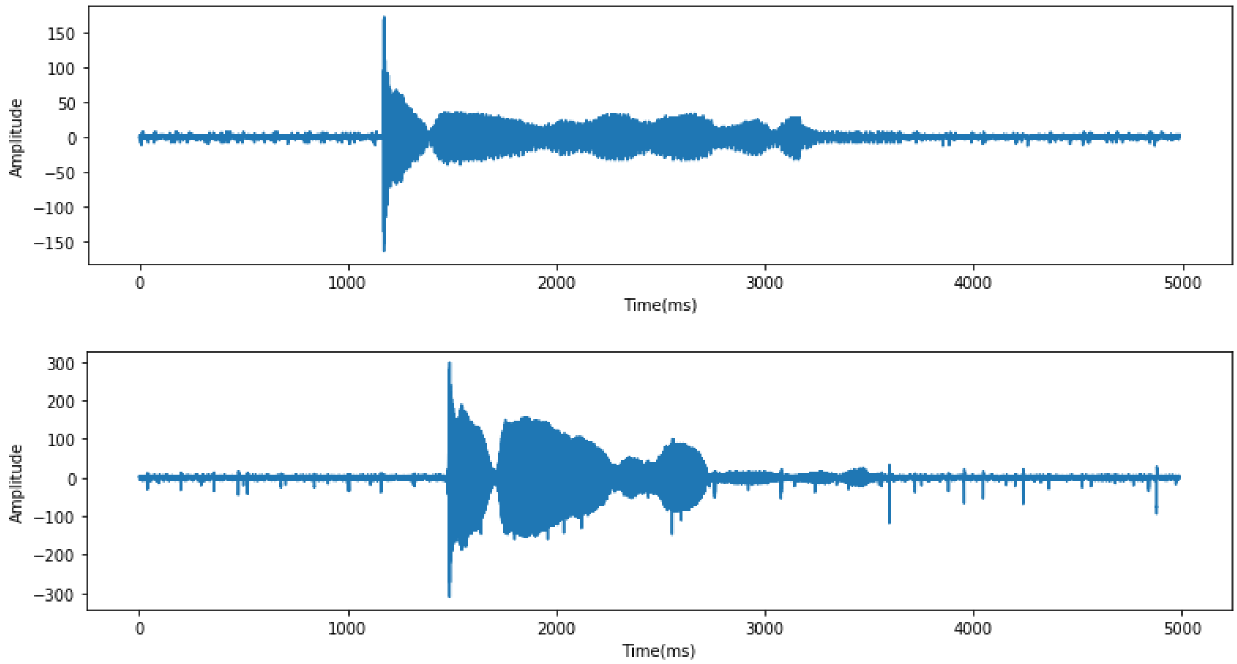


Fig 40: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

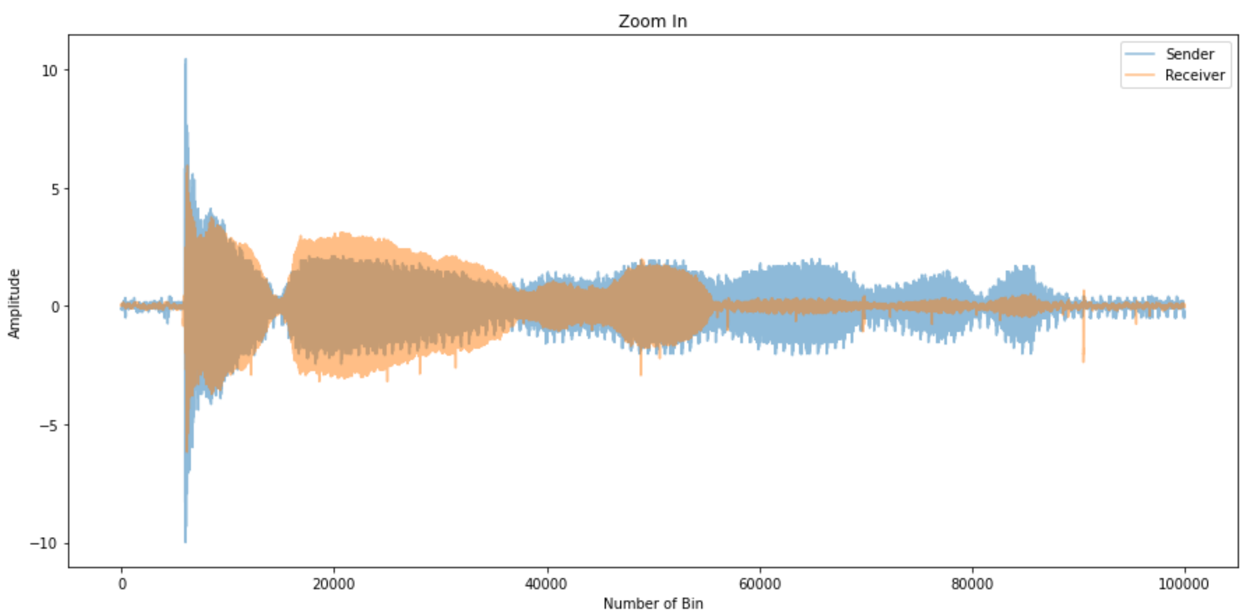


Fig 41: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 41000; the receiver's envelope shown starts from bin = 53500, both have a length of 100000 bins.

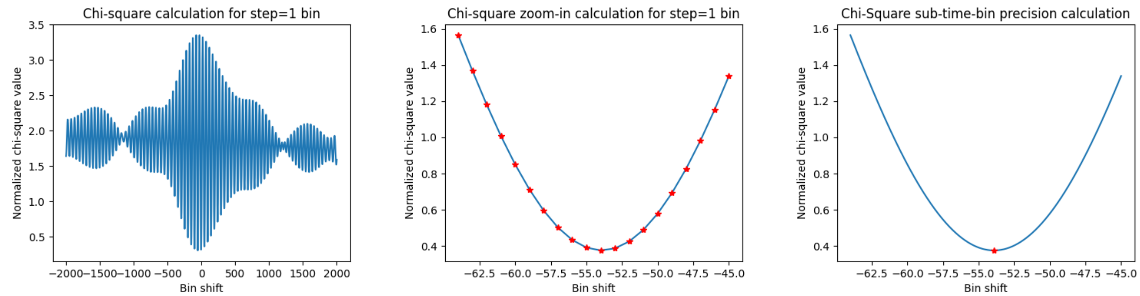


Fig 42: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

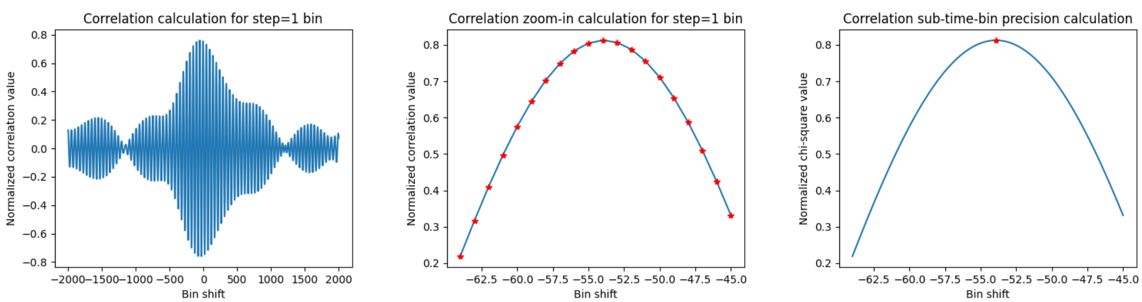


Fig 43: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of -53.92 with error ±0.0301. The corresponding latency is **309.81 milliseconds**. The maximum correlation corresponds to a decimal bin shift of -53.92 with error ±0.0425. The corresponding latency is **309.81 milliseconds**.

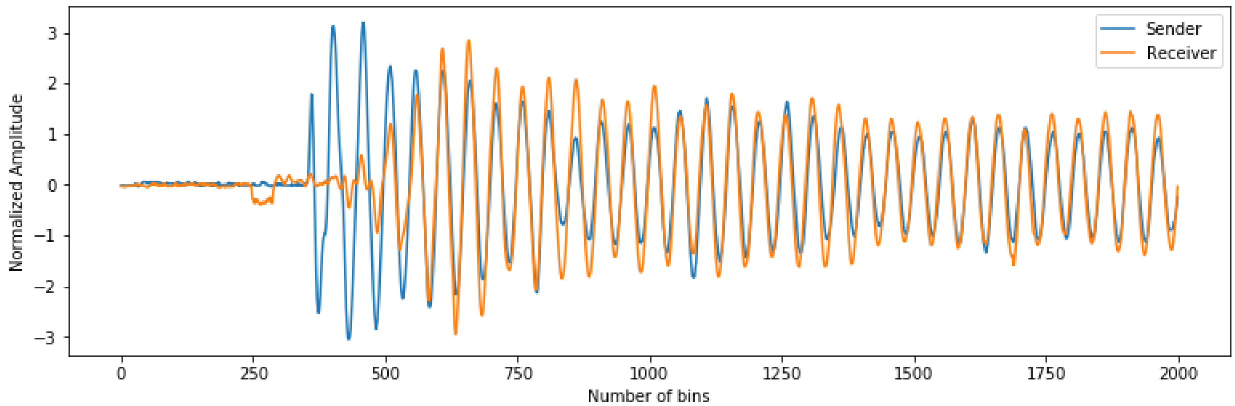


Fig 44. Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 46600, receiver envelope starts from bin number = 59046, both have a length of 2000 bins.

Table 5. Results of Latency Test 4, 5, 6 (4G, VoLTE Off)

	Sender Calibration Index (SCI)	Receiver Calibration Index (RCI)	Sender Crop Range (1000 Bins)	Receiver Crop Range (1000 Bins)	Bin Shift At Min $\chi 2$/ Max correlation	Latency (ms)
Test 4.	1.00030	1.00021	45-145	55-155	-361.82	266.01
Test 5.	1.00030	1.00021	43-143	53.5-153.5	97.72	264.99
Test 6.	0.999098	1.00021	41-141	53.5-153.5	-53.92	309.81

Average precision of Chi square tests: 0.0303 bins = 0.758 microseconds.

1.3. Calls Under 3G Network (At the Midnights of Nov 10 & Nov 11, 2020)

Test 7.

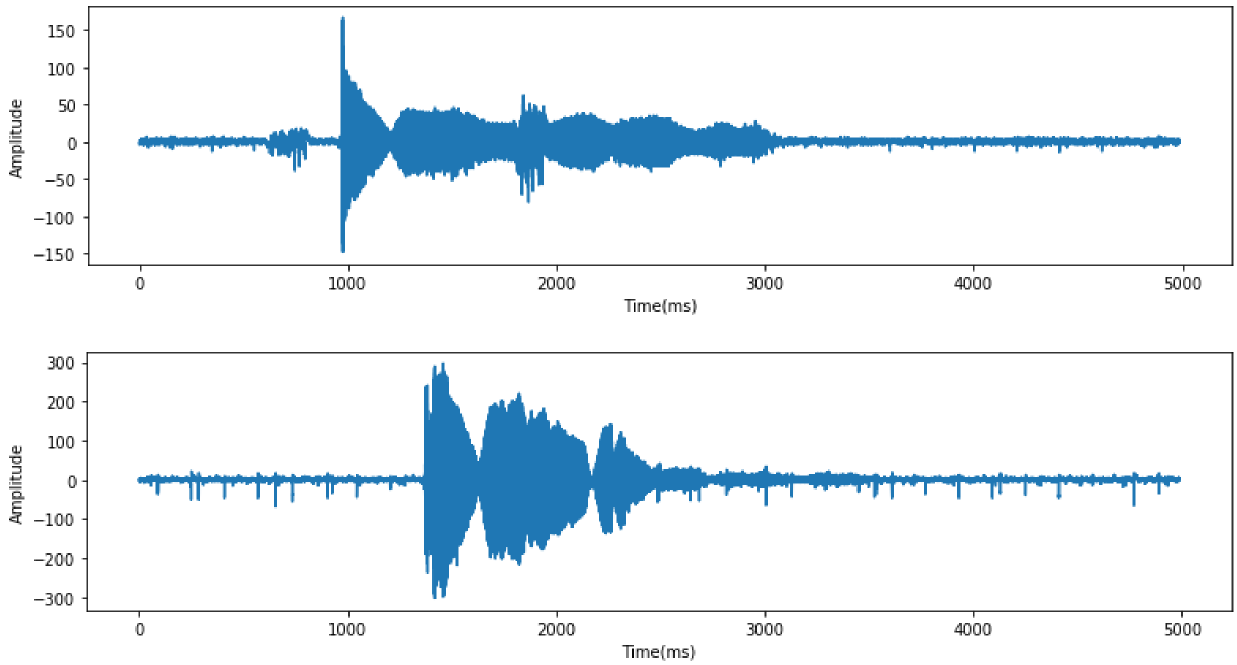


Fig 45: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

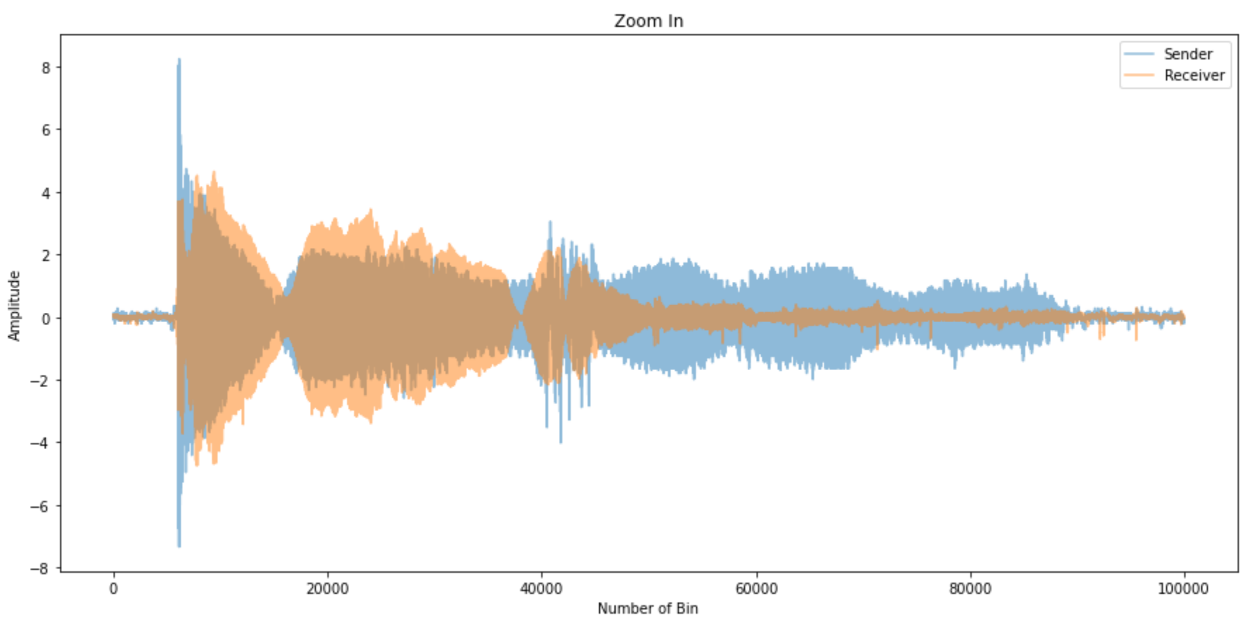


Fig 46: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 33000; the receiver's envelope shown starts from bin = 49000, both have a length of 100000 bins.

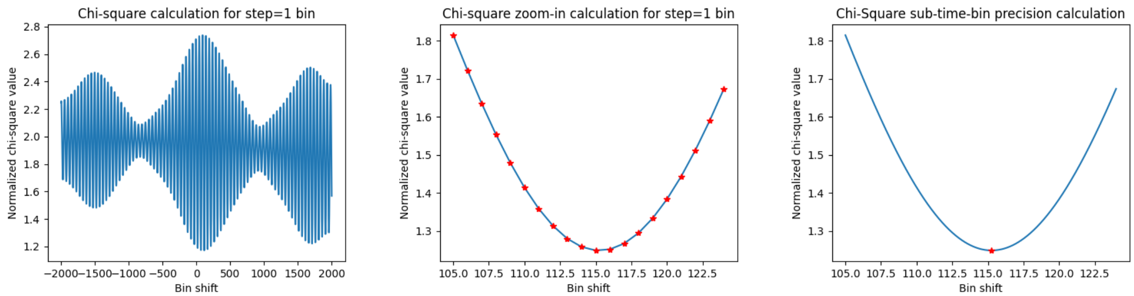


Fig 47: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

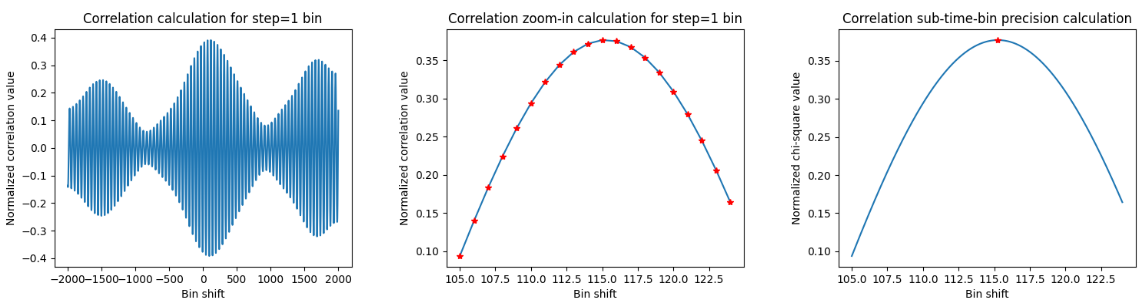


Fig 48: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of 115.26 with error ±0.0443. The corresponding latency is **401.74 milliseconds**. The maximum correlation corresponds to a decimal bin shift of 115.26 with error ±0.0626. The corresponding latency is **401.74 milliseconds**.

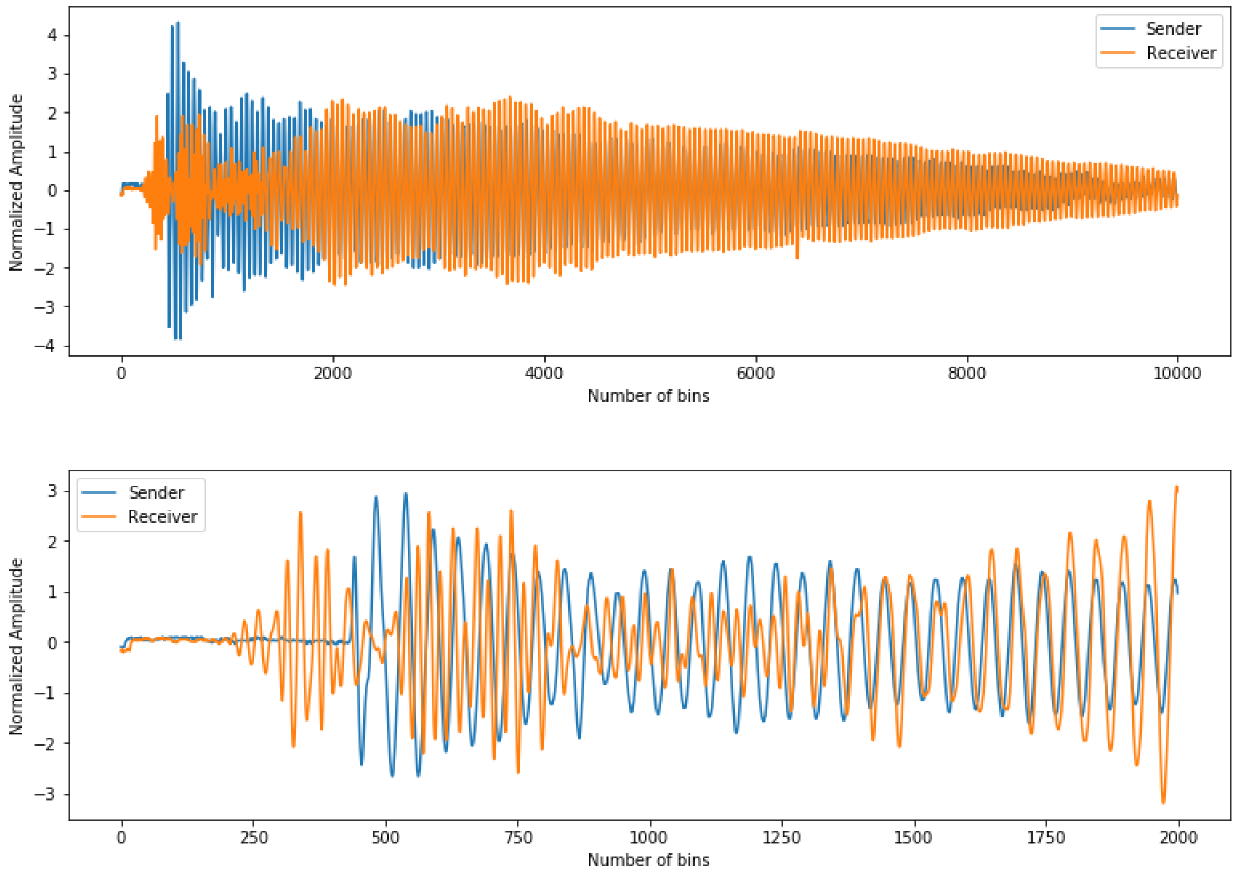
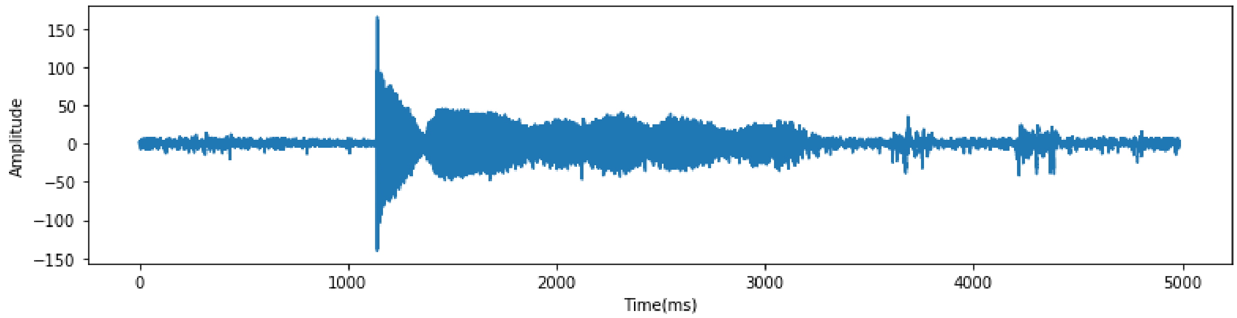


Fig 49. Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 38600, receiver envelope starts from bin number = 54715, both have a length of 10000 bins (top). Sender envelope starts from bin number = 38600, receiver envelope starts from bin number = 54715, both have a length of 2000 bins (bottom)

Test 8.



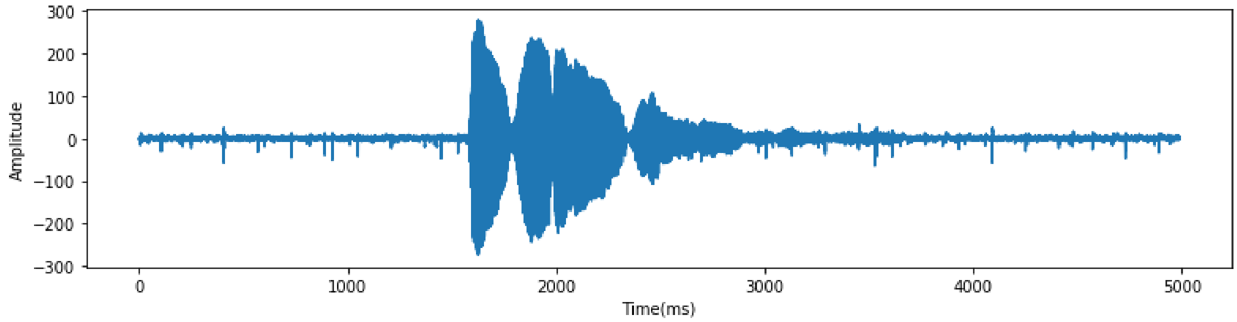


Fig 50: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

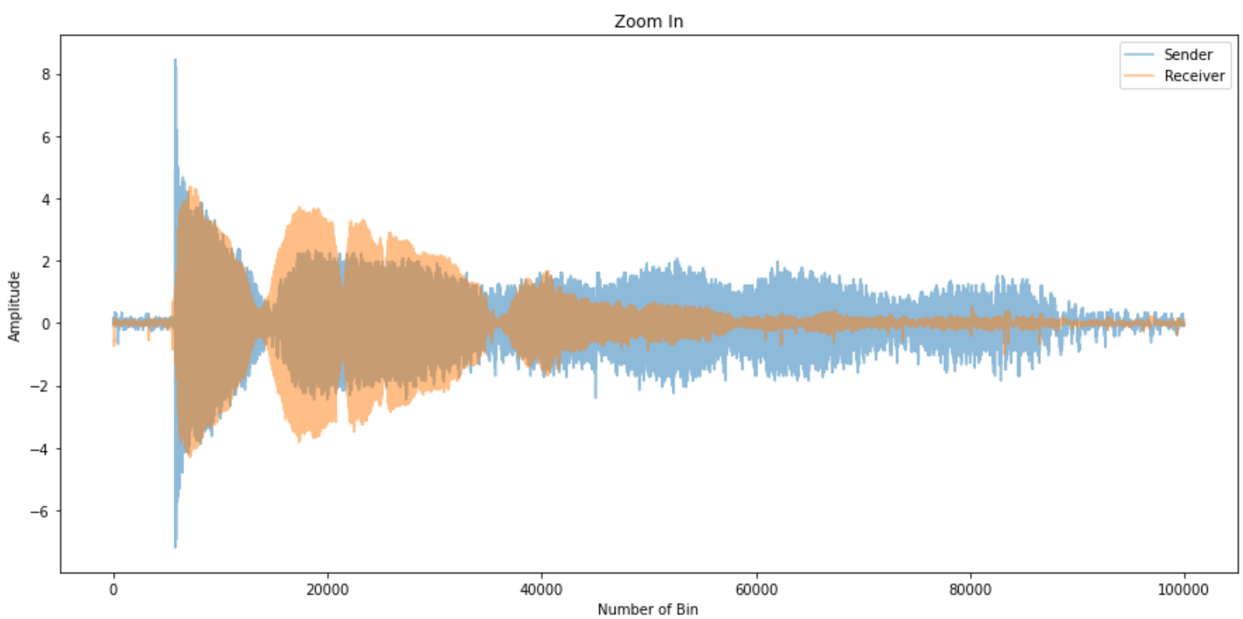


Fig 51: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 40000; the receiver's envelope shown starts from bin = 58000, both have a length of 100000 bins.

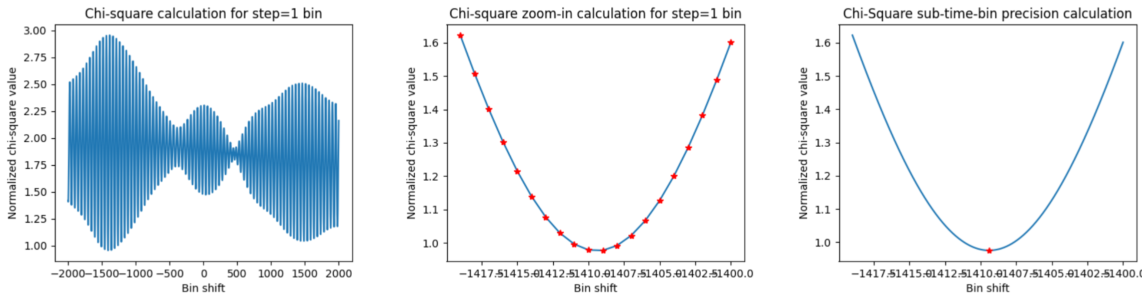


Fig 52: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

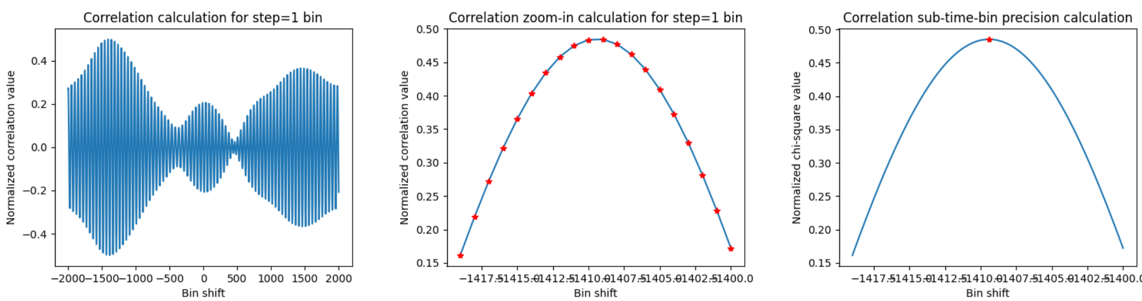


Fig 53: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of -1409.41 with error ±0.039. The corresponding latency is **413.43 milliseconds**.The maximum correlation corresponds to a decimal bin shift of -1409.41 with error ±0.0552. The corresponding latency is **413.43 milliseconds**.

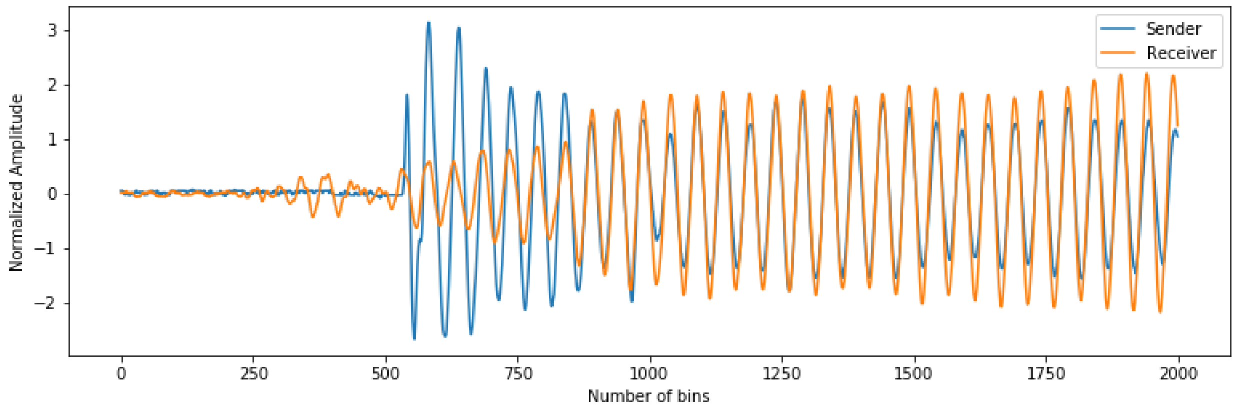


Fig 54. Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 45200, receiver envelope starts from bin number = 61780, both have a length of 2000 bins (top).

Test 9.

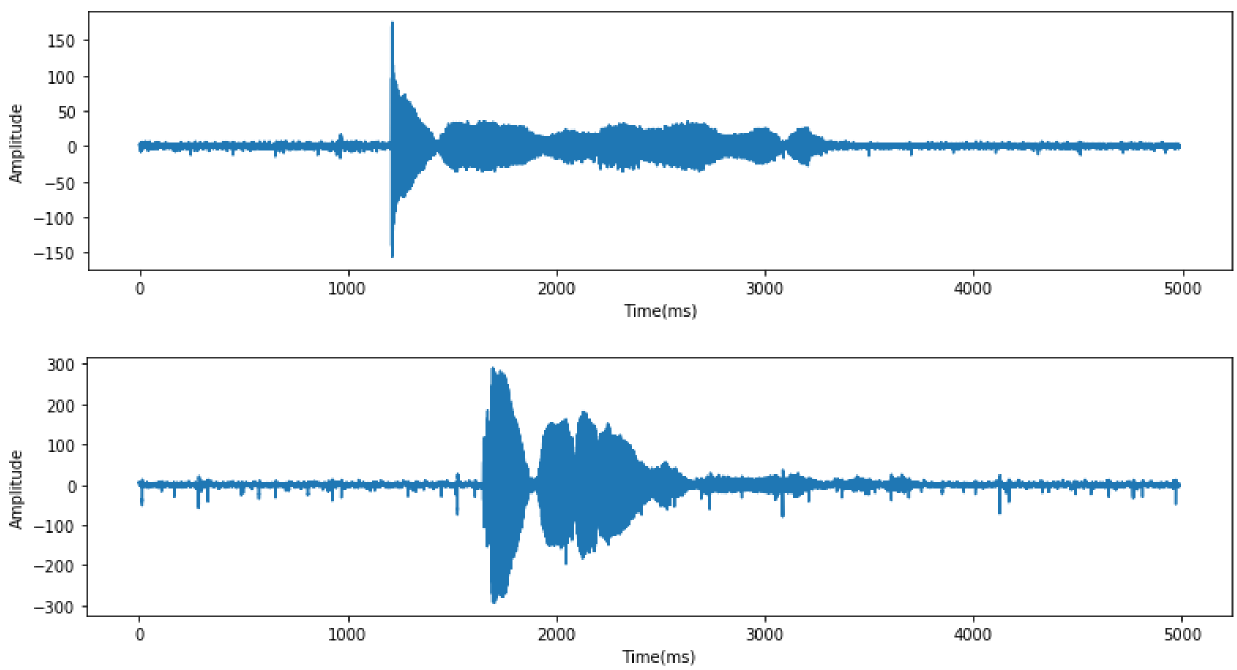


Fig 55: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

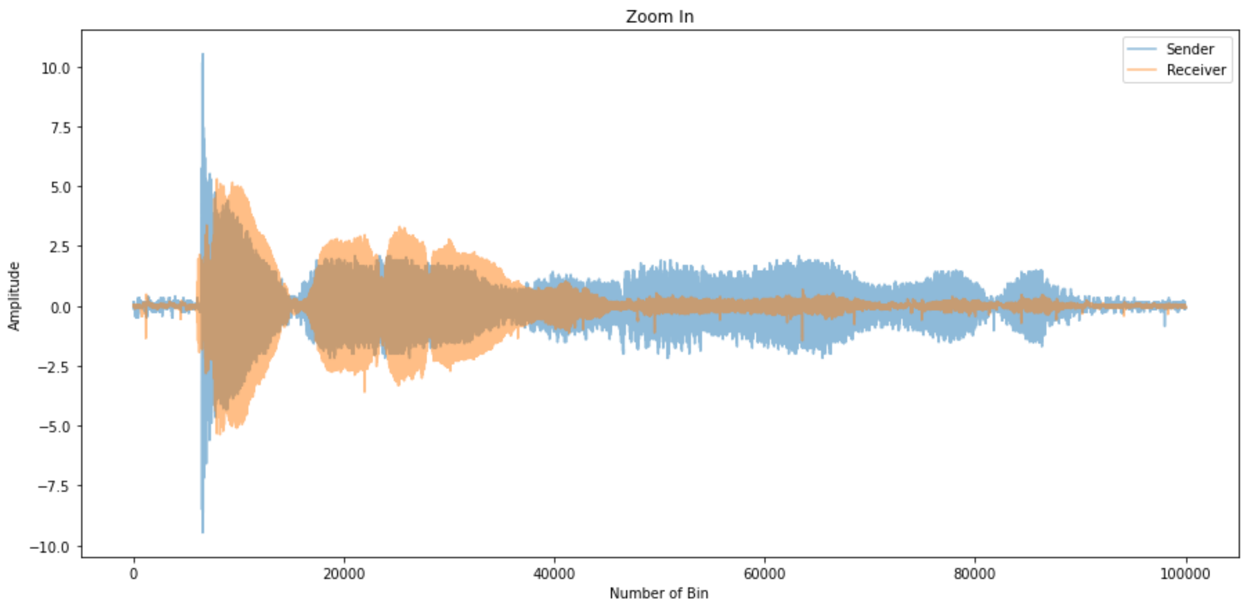


Fig 56: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 42000; the receiver's envelope shown starts from bin = 60000, both have a length of 100000 bins.

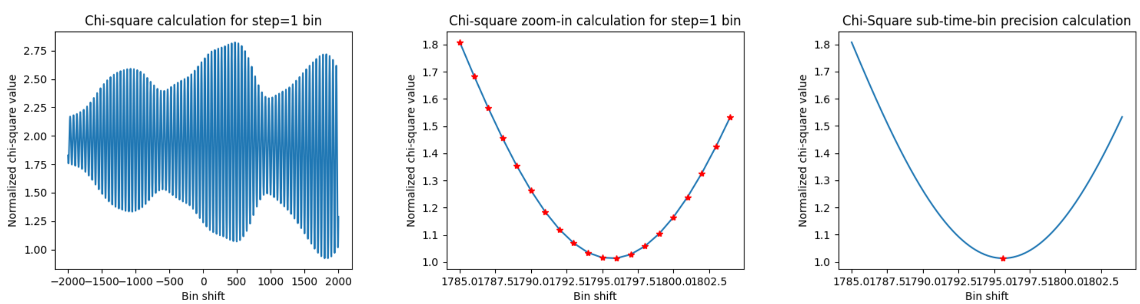


Fig 57: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

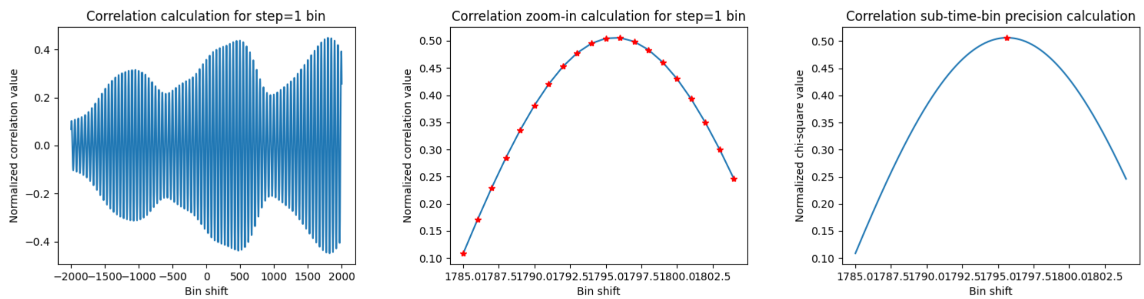


Fig 58: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of 1795.64 with error ±0.0385. The corresponding latency is **493.48 milliseconds**. The maximum correlation corresponds to a decimal bin shift of 1795.64 with error ±0.0545. The corresponding latency is **493.48 milliseconds**.

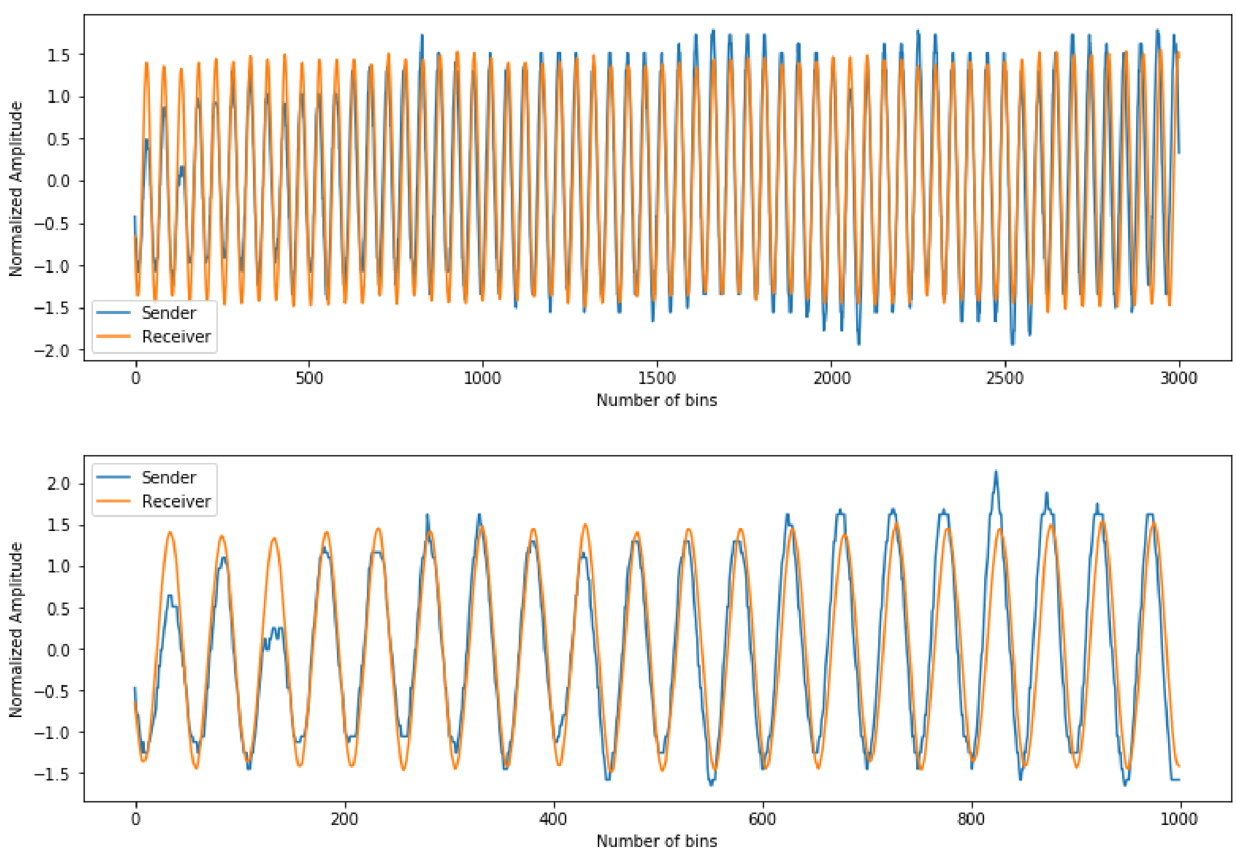


Fig 59. Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 58600, receiver envelope starts from bin number = 78396, both have a length of 3000 bins (top). Sender envelope starts from bin number = 58600, receiver envelope starts from bin number = 78396, both have a length of 1000 bins (bottom).

Table 6. Results of Latency Test 7, 8, 9 (3G)

	Sender Calibration Index (SCI)	Receiver Calibration Index (RCI)	Sender Crop Range (1000 Bins)	Receiver Crop Range (1000 Bins)	Bin Shift At Min χ2/ Max correlation	Latency (ms)
Test 7.	0.999096	1.00021	33-133	49-149	115.26	401.74
Test 8.	0.999100	1.00021	40-140	58-158	-1409.41	413.43
Test 9.	0.999099	1.00021	42-142	60-160	1795.64	493.48

Average precision of Chi square tests: 0.0406 bins = 1.02 microseconds.

Test 10. (3G Network, Midnight of Nov 10, 2020)

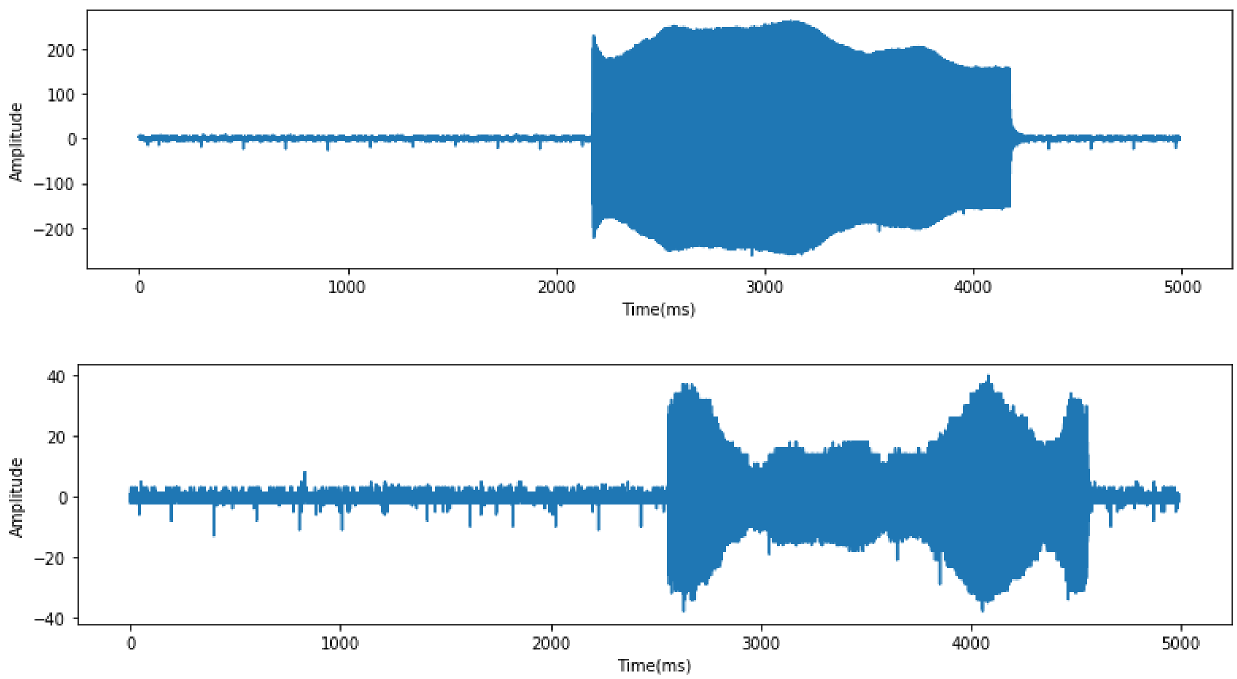


Fig 60 : Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

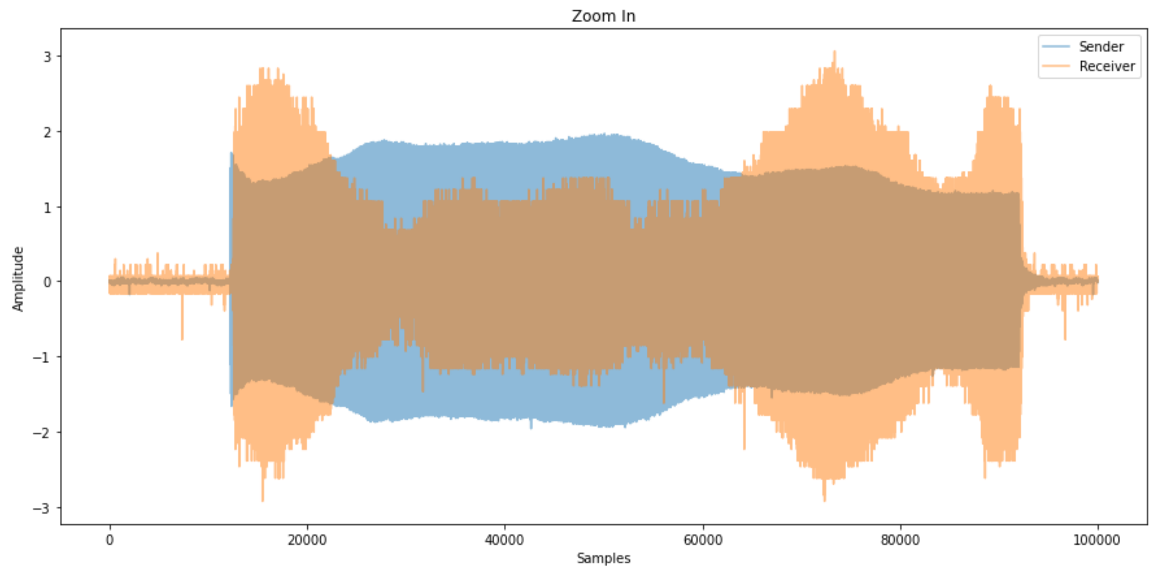


Fig 61: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 50000; the receiver's envelope shown starts from bin = 60000, both have a length of 100000 bins.

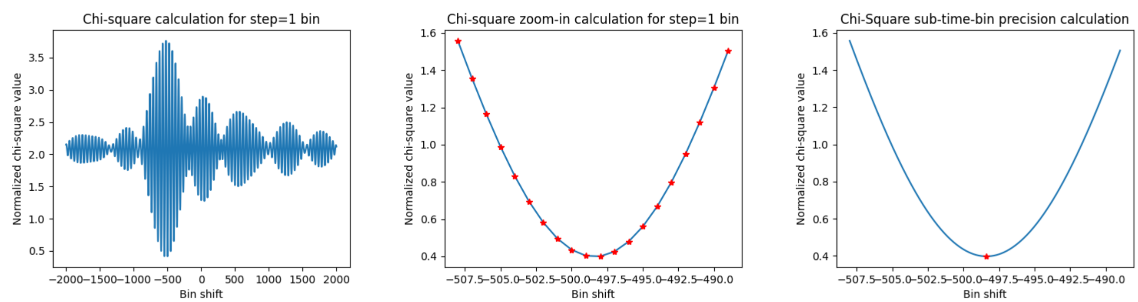


Fig 53: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

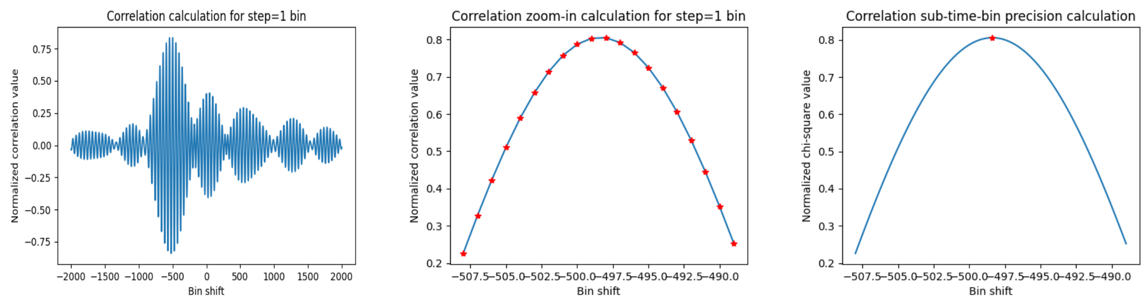


Fig 62: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of -498.39 with error ±0.0264. The corresponding latency is **360.68 milliseconds**. The maximum correlation corresponds to a decimal bin shift of -498.39 with error ±0.0373. The corresponding latency is **360.68 milliseconds**.

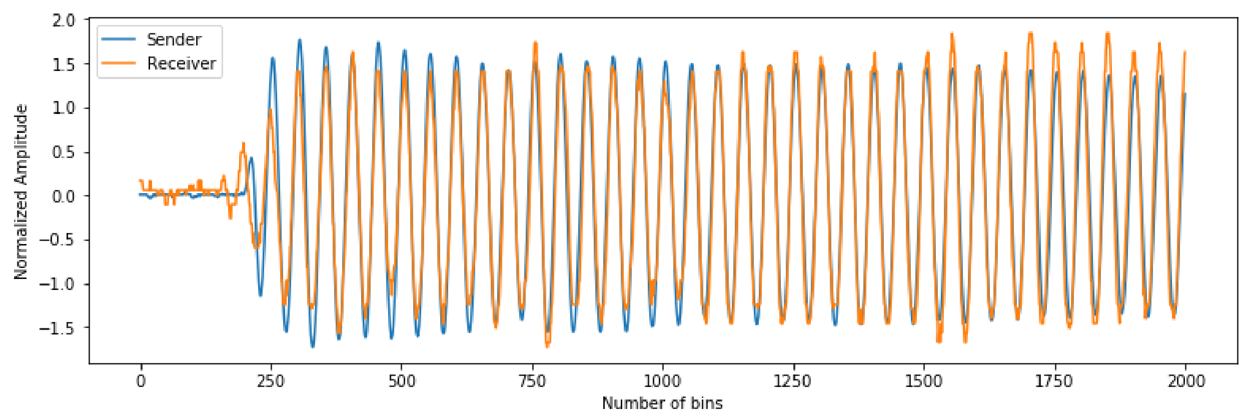


Fig 63. Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 87000, receiver envelope starts from bin number = 101502, both have a length of 2000 bins.

Test 11. (3G Network, Midnight of Nov 10, 2020)

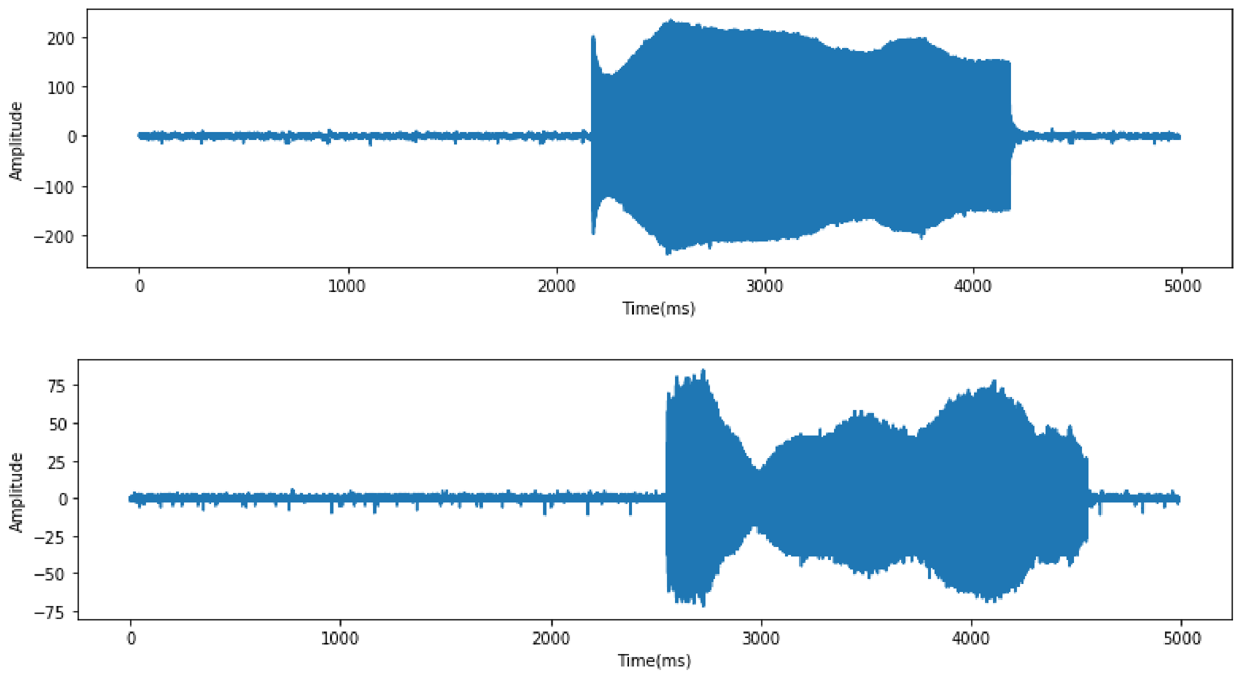


Fig 64: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

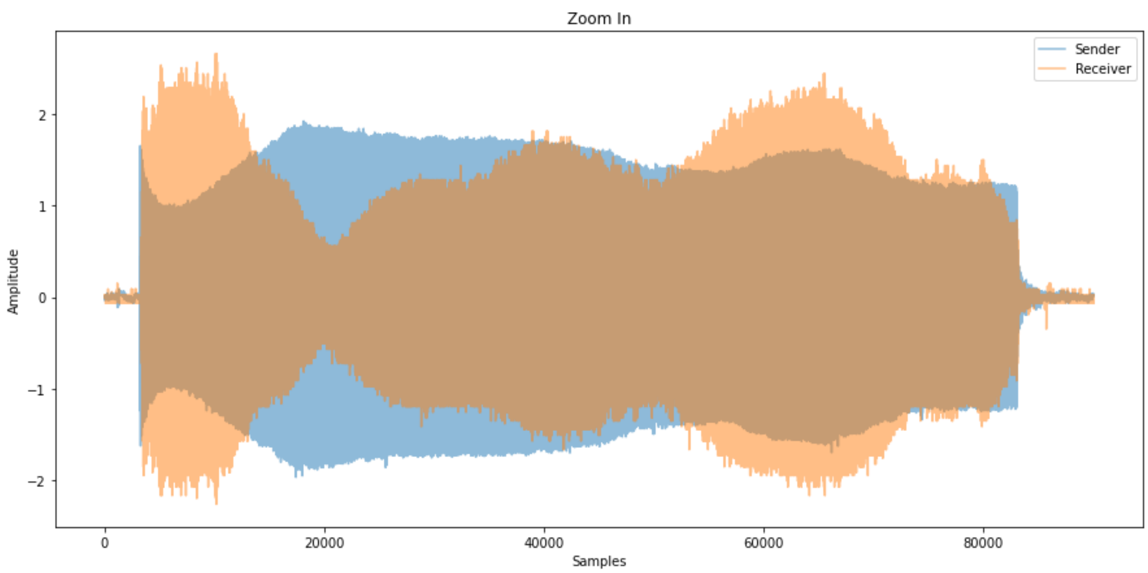


Fig 65: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 45000; the receiver's envelope shown starts from bin = 56000, both have a length of 100000 bins.

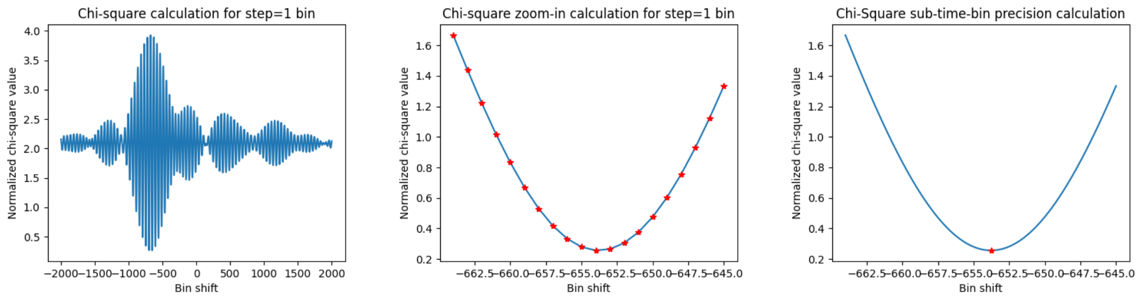


Fig 66: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

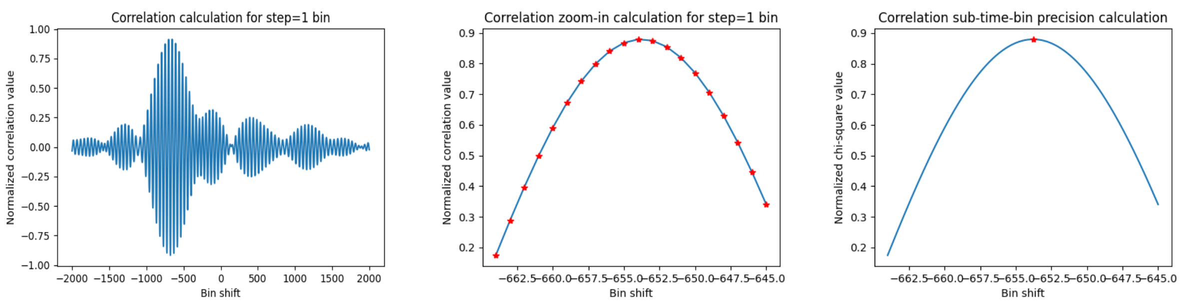


Fig 67: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi square corresponds to a decimal bin shift of -653.78 with error ±0.0266. The corresponding latency is **356.58 milliseconds**. The maximum correlation corresponds to a decimal bin shift of -653.78 with error ±0.0375. The corresponding latency is **356.58 milliseconds**.

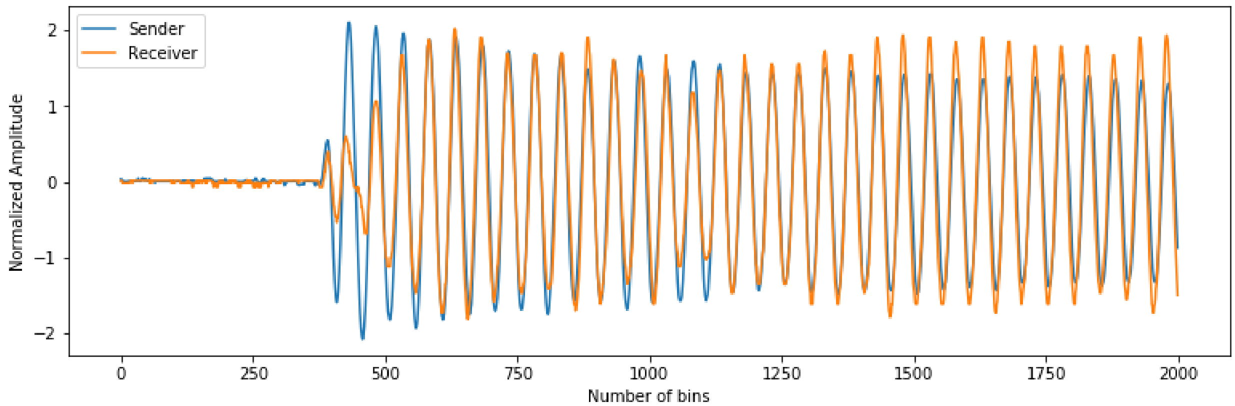


Fig 68:Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 86800, receiver envelope starts from bin number = 101146, both have a length of 2000 bins.

Test 12. (3G Network, Midnight of Nov 10, 2020)

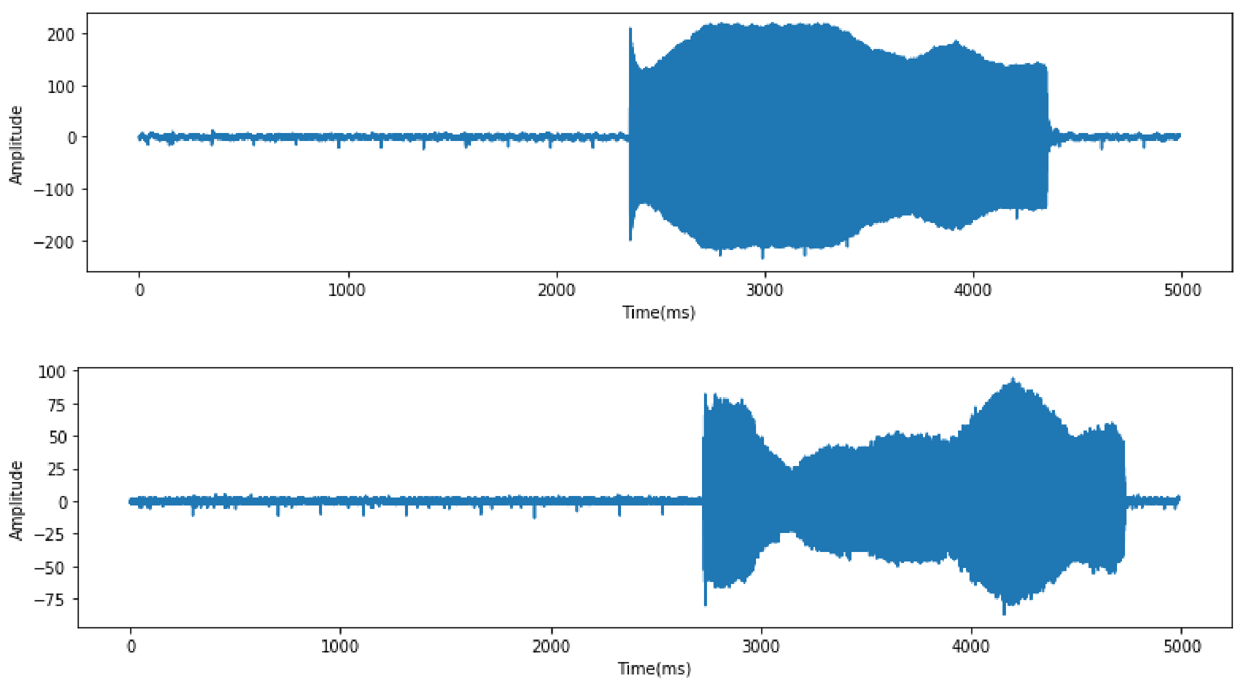


Fig 69: Overall envelopes of recorded binary files from sender (top) and receiver (bottom) separately.

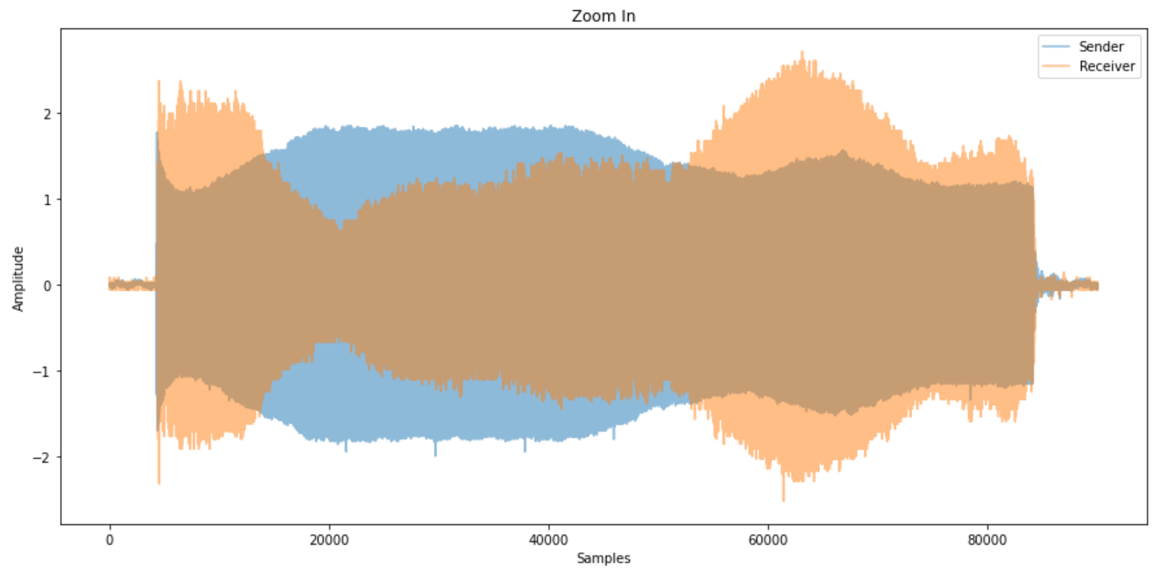


Fig 70: Comparing the zoomed in signal patterns (after cropping) between the sender side (blue) and the receiver side (orange). Both envelopes are cropped, the sender's envelope shown starts from bin = 46000; the receiver's envelope shown starts from bin = 53500, both have a length of 100000 bins.

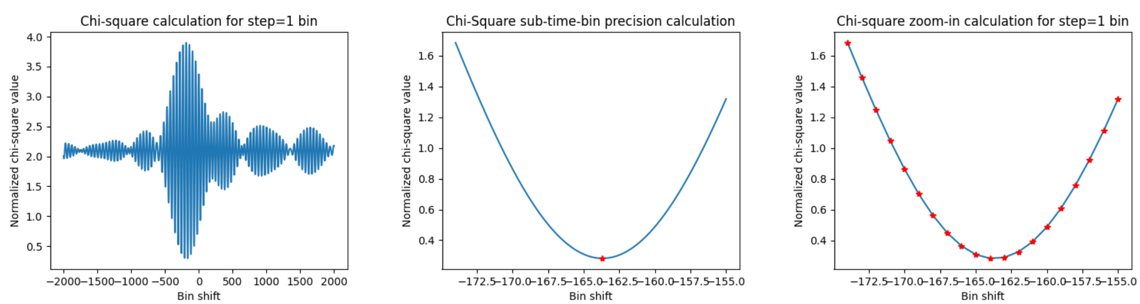


Fig 71: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized chi-square values to find the difference between two patterns (left). Zoomed in plot around the minimum of chi-square (middle). Locating the decimal bin number corresponding to the minimum chi square value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized chi-square value.

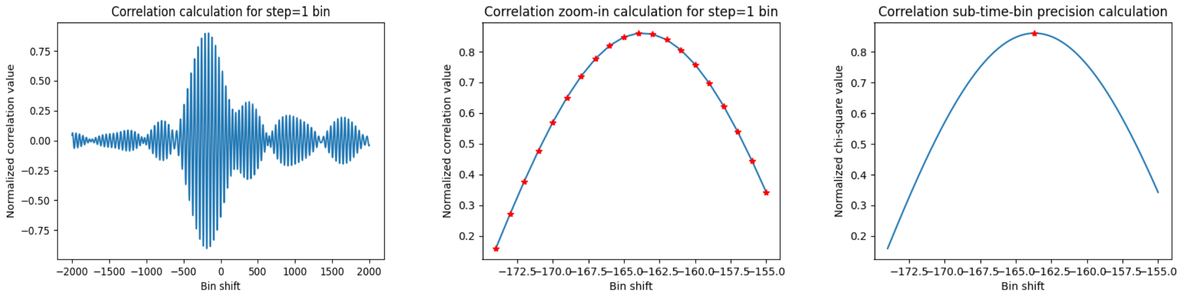


Fig 72: Move the receiver's cropped envelope from -2000 bins to 2000 bins with step = 1 bin, calculate the corresponding normalized correlation values to find the difference between two patterns (left). Locating the decimal bin number corresponding to the maximum correlation value using polynomial curve fitting (right). For all three plots, x axis is the number of bins shifted, y axis is the normalized correlation value.

Result: The minimum chi-square corresponds to a decimal bin shift of -163.69 with error ±0.0268. The corresponding latency is **356.18 milliseconds**. The maximum correlation corresponds to a decimal bin shift of -163.69 with error ±0.0379. The corresponding latency is **356.18 milliseconds**.

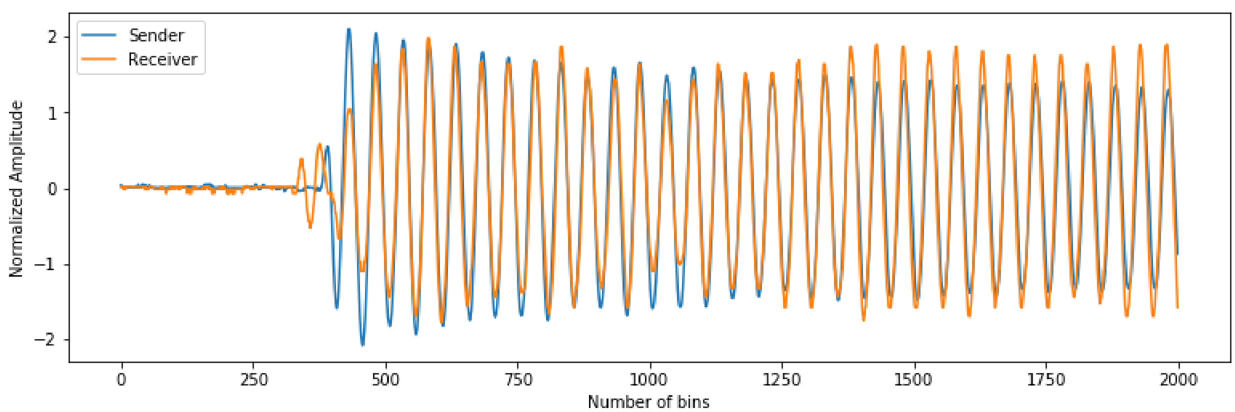


Fig 73: Zoomed in plot of sender and receiver envelopes after chi-square and correlation match. Sender envelope starts from bin number = 86800, receiver envelope starts from bin number = 101636, both have a length of 2000 bins.

Table 7. Results of Test 10, 11, 12 (3G, Midnight of Nov 10)

	Sender Calibration Index (SCI)	Receiver Calibration Index (RCI)	Sender Crop Range (1000 Bins)	Receiver Crop Range (1000 Bins)	Bin Shift At Min χ2/ Max correlation	Latency (ms)
Test 10.	0.999184	1.00021	50-150	60-160	-498.39	360.68
Test 11.	0.999182	1.00021	45-145	56-156	-653.78	356.58

Test 12.	10.999181	1.00021	46-146	53.5-153.5	-163.69	356.18

Average precision of Chi square tests: 0.0266 bins = 0.665 microseconds.

2. Stability of Latency

2.1 Stability of Latency During Daytime (Long Term Behavior)

Test 13. (4G, Duration = 8 hours, 6 times per hour, Nov 14 9:00AM to 5:00PM)

On November 14, we did 6*8 = 48 sets of measurements to determine the stability of latency during a day. And here are the results. (See 4.3 for definition of Calibration Indices)

Table 8. Long Term Latency of 4G Network, Nov 14 from 9:00 to 17:00

Time	Sender Calibration Index	Receiver Calibration Index	Bin Shift at Min $\chi2$	Bin Shift at Max Correlation	Latency by $\chi2$ (ms)	Latency by Correlation (ms)	Chi square Test Precision ($10^{-3} ms$)
9:00	1.00032	1.00021	623.19	623.19	217.48	217.48	0.763
9:10	1.00034	1.00021	118.36	118.36	213.22	213.22	0.770
9:20	1.00031	1.00021	-249.37	-249.37	215.42	215.42	0.748
9:30	1.00032	1.00021	338.10	338.10	238.85	238.85	0.775
9:40	1.00030	1.00021	-1096.54	-1096.54	242.86	242.86	0.820
9:50	1.00032	1.00021	234.68	234.68	236.92	236.92	0.770
10:00	1.00033	1.00021	85.05	85.05	247.84	247.84	0.825
10:10	1.00034	1.00020	-1128.84	-1128.84	251.37	251.37	1.01
10:20	1.00032	1.00021	747.93	747.93	246.78	246.78	0.963
10:30	1.00033	1.00021	-915.33	-915.33	238.81	238.81	0.772
10:40	1.00029	1.00021	731.94	731.94	271.46	271.46	0.875
10:50	0.999097	1.00021	773.55	773.55	273.03	273.03	0.833
11:00	1.00034	1.00021	920.14	920.14	323.10	323.10	0.783
11:10	1.00034	1.00022	-838.91	-838.91	309.45	309.45	0.793

11:30	1.00030	1.00021	-1576.02	-1576.02	316.49	316.49	0.935
11:40	1.00034	1.00021	837.64	837.64	289.28	289.28	0.795
11:50	1.00031	1.00021	390.83	390.83	271.96	271.96	0.918
12:00	0.999098	1.00021	-384.19	-384.19	276.45	276.45	0.852
12:10	1.00033	1.00021	1894.74	1894.74	272.31	272.31	0.813
12:20	1.00031	1.00021	201.94	201.94	256.38	256.38	0.735
12:30	1.00034	1.00021	-219.22	-219.22	251.96	251.96	0.967
12:40	1.00032	1.00021	1628.01	1628.01	262.34	262.34	0.858
12:50	1.00034	1.00021	987.68	987.68	248.57	248.57	1.03
13:00	1.00030	1.00021	217.88	217.88	253.85	253.85	0.751
13:10	1.00031	1.00021	-866.30	-866.30	266.70	266.70	0.737
13:20	1.00035	1.00021	82.96	82.96	272.92	272.92	0.858
13:30	1.00033	1.00021	597.39	597.39	269.63	269.63	0.792
13:40	1.00029	1.00021	-189.77	-189.77	266.84	266.84	0.765
13:50	1.00030	1.00022	-745.71	-745.71	273.05	273.05	0.846
14:00	1.00031	1.00021	-422.32	-422.32	285.16	285.16	0.869
14:10	1.00033	1.00021	645.28	645.28	272.74	272.74	0.882
14:20	1.00030	1.00021	-160.77	-160.77	278.66	278.66	0.890
14:30	1.00034	1.00021	234.93	234.93	283.59	283.59	0.850
14:40	1.00031	1.00022	-1392.81	-1392.81	273.21	273.21	0.831
14:50	0.999096	1.00021	-348.31	-348.31	270.05	270.05	1.00
15:00	1.00035	1.00021	101.29	101.29	288.68	288.68	0.927
15:10	1.00034	1.00020	-805.13	-805.13	286.95	286.95	0.879
15:20	1.00035	1.00021	1218.16	1218.16	265.32	265.32	0.827
15:30	1.00030	1.00021	1067.31	1067.31	276.36	276.36	0.892

15:40	1.00032	1.00021	-1383.45	-1383.45	258.46	258.46	0.788
15:50	1.00034	1.00021	309.72	309.72	261.91	261.91	0.765
16:00	1.00033	1.00021	-391.62	-391.62	270.45	270.45	0.810
16:10	1.00030	1.00021	201.19	201.19	245.25	245.25	0.755
16:20	1.00028	1.00021	1399.59	1399.59	233.20	233.20	0.744
16:30	1.00035	1.00020	-1840.33	-1840.33	249.66	249.66	0.825
16:40	1.00034	1.00021	591.14	591.14	252.537	252.537	0.746
16:50	1.00035	1.00021	882.71	882.71	226.36	226.36	0.732
17:00	1.00032	1.00021	-793.55	-793.55	243.41	243.41	0.922

Mean = 263.94 ms; Standard Deviation = 22.74 ms; Maximum Latency = 323.10 ms; Minimum Latency = 213.22 ms, Mean Chi Square Test Precision: 0.835 us.

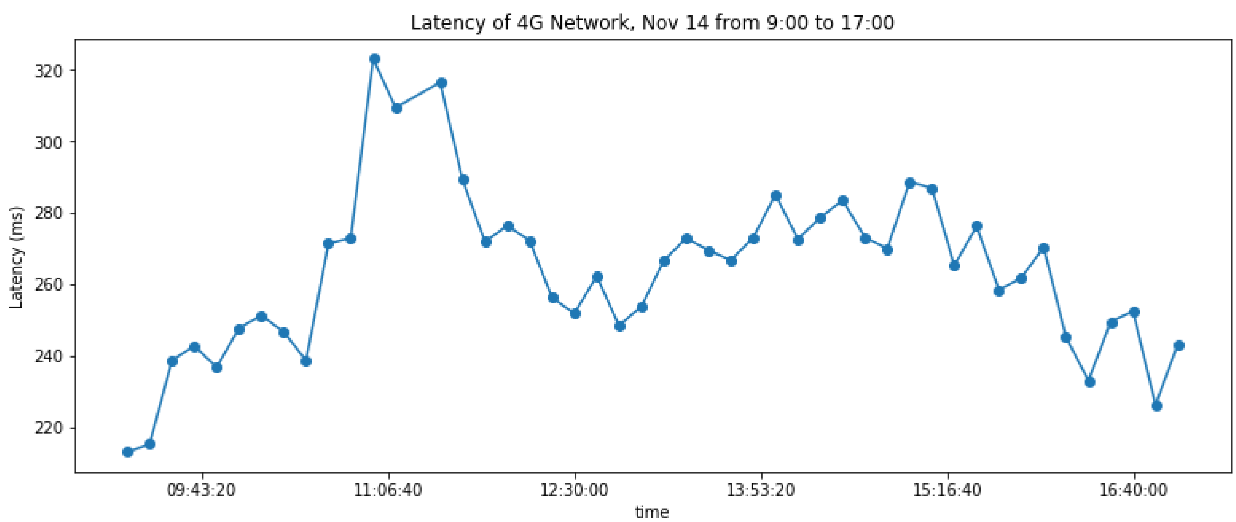


Fig 74: Latency change in milliseconds during 9:00 to 17:00, Nov 14, 2020. Sample collecting rate = 6 per hour.

2.2 Stability of Latency During Daytime (Short Term Behavior)

To investigate the short term stability of latency, we record a 2-minute long audio to analyze the latency change during 2 minutes. The sound source we used is different from the previous: the 2-second chirp signals will repeat every five seconds, i.e. it will play 12 times per minute. On November 19, we did 12*2 = 24 sets of measurements to determine the stability of latency during several minutes. And here are the results.

Table 9.Short Term Latency of 4G Network, Nov 19 from 12:33:00 PM to 12:35:00 PM

Time (Estimated)	Bin Shift at Min $\chi 2$	Bin Shift at Max Correlation	Latency by $\chi 2$ (ms)	Latency by Correlation (ms)	Chi square Test Precision ($10^{-3} ms$)
12:33:00	-427.33	-427.33	248.56	248.56	0.772
12:33:05	-366.78	-366.78	248.00	248.00	0.865
12:33:10	620.94	620.94	251.77	251.77	0.783
12:33:15	233.70	233.70	248.47	248.47	0.792
12:33:20	485.09	485.09	246.02	246.02	0.779
12:33:25	-827.28	-827.28	246.60	246.60	0.823
12:33:30	571.30	571.30	249.03	249.03	0.835
12:33:35	-321.54	-321.54	266.56	266.56	0.843
12:33:40	535.38	535.38	266.53	266.53	0.796
12:33:45	-442.69	-442.69	266.14	266.14	0.771
12:33:50	-623.73	-623.73	269.47	269.47	0.810
12:33:55	-926.11	-926.11	270.05	270.05	0.835
12:34:00	213.59	213.59	274.60	274.60	0.891
12:34:05	-379.84	-379.84	267.27	267.27	0.849
12:34:10	193.60	193.60	266.72	266.72	0.833
12:34:15	-184.32	-184.32	266.73	266.73	0.789
12:34:20	396.57	396.57	273.30	273.30	0.828
12:34:25	95.46	95.46	266.64	266.64	0.793
12:34:30	298.25	298.25	266.04	266.04	0.772
12:34:35	-484.58	-484.58	265.23	265.23	0.791
12:34:40	305.64	305.64	265.25	265.25	0.763
12:34:45	-329.80	-329.80	265.24	265.24	0.805

12:34:50	-339.42	-339.42	266.64	266.64	0.838
12:34:55	501.37	501.37	266.01	266.01	0.887

Mean = 261.95 ms; Standard Deviation = 9.06 ms; Maximum Latency = 274.60 ms; Minimum Latency = 246.02 ms, Mean Chi Square Test Precision: 0.814 us.

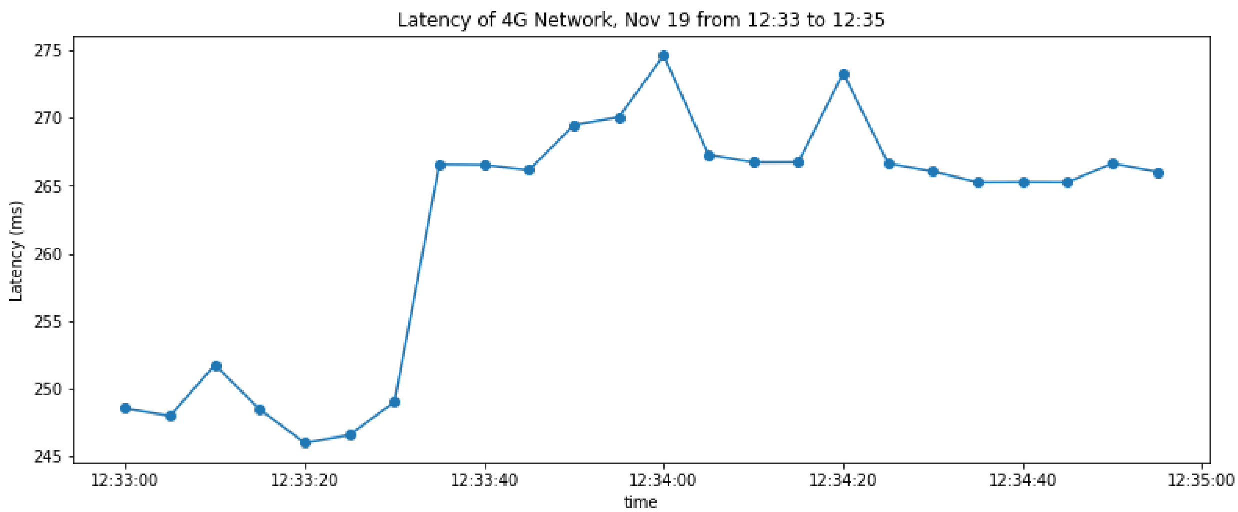


Fig 75: Latency change in milliseconds during 12:33:00 to 12:35:00, Nov 19, 2020. Sample collecting rate = 12 per minute.

3. Calls Under Different Distances

We also tried to find the relationship of latency with respect to the distance between sender and receiver. We measured latency when the sender and receiver were separated by 700 meters apart and 1400 meters apart. However, the latency in 700 meters was sometimes larger than in 1400 meters. We were unable to measure the latency when the sender and receiver were very far apart. Based on our tests, small distance change is not the dominant factor of latency change.

4. Summary of Results

Table 10. Summary of Test Results - Latency Change Under Different Type of Networks

Test #	Network Type	Time	Latency (ms)	Mean (ms)	Standard Deviation (ms)	Mean Chi Square Test Precision (10^{-3}ms)
1	4G VoLTE	Midnight Nov 11,	243.77	244.284	17.858	0.798
2			266.41			

3	On	2020	222.68			
4	4G VoLTE Off	Midnight Nov 11, 2020	266.01	280.268	20.890	0.758
5			264.99			
6			309.81			
7	3G	Midnight Nov 11, 2020	401.738	436.217	40.773	1.02
8			413.431			
9			493.483			
10	3G	Midnight Nov 10, 2020	360.683	357.814	2.035	0.665
11			356.582			
12			356.178			

Table 11. Summary of Test Results - Stability of Network

Test Type	Network	Time	Average Latency (ms)	Standard Deviation (ms)	Range (ms)	Mean Chi Square Test Precision (10^{-3}ms)
Long Term Stability (day scale)	4G VoLTE On	9:00 - 17:00 Nov 14, 2020	263.94	22.74	109.88	0.835
Short Term Stability (minute scale)	4G VoLTE On	12:33 - 12:35, Nov 19, 2020	261.95	9.06	28.58	0.814

Conclusion

We initially guess that the latency is dominated by the distance between sender and receiver. However, in our experiments there is no obvious difference in latencies measured under different distances. It also agrees with the fact that the EM wave goes so fast that it is not sensitive to small distance changes. What we found is that the latency mainly depends on the speed of the cell phone network - faster the speed is, the smaller the latency becomes. In our experiments, we used three different kinds of networks: 4G with VoLTE on (fastest), 4G with VoLTE off (slower), 3G (slowest). We found that the 4G VoLTE on latency is about 244ms, 4G VoLTE off latency is about 280 ms, and 3G latency is around 400ms. The precisions of our chi-square tests vary from 0.76 us to 1.02 us. Given the latency's dominant factor is Information Density/Speed, it agrees with the fact that 3G network's speed is much slower than 4G network.

We also investigate the stability of latency (of 4G network with VoLTE on) in both long time scale and short time scale. As for long time behavior, we recorded 48 sets of data during 8 hours, and we found that the latency was smallest at around 9:00 AM on that day, which is about 213 ms. More detailed, the average latency during that eight hours is 263.94 ms, with a standard deviation of 22.74 ms and a range of 109.88 ms. The precisions of chi-square tests are around 0.84 us. As for short time behavior, we recorded 24 sets of data during 2 minutes, the average latency during that two minutes is 261.95 ms, with a standard deviation of 9.06 ms and a range of 28.58 ms. The precisions of chi-square tests are around 0.81 us. For long-term behavior, sudden change of latency rarely appears in our measurements. For short-term behavior, the fluctuation of latency is only about 30 microseconds. This suggests that the latency of cell phone networks in a short term is stable.

References

[1] *Arduino - Products . (2020). Retrieved 12 November 2020, from https://www.arduino.cc/en/Main/Products*

[2] *Arduino. (2020). Retrieved 12 November 2020, from https://en.wikipedia.org/wiki/Arduino*

[3] *Chirp signal. (2020). Retrieved 4 November 2020, from https://en.wikipedia.org/wiki/Chirp*

[4] *Chi-squared test. (2020). Retrieved 6 November 2020, from https://en.wikipedia.org/wiki/Chi-squared_test*

[5] *Chicago Air Shower Array. (2020). Retrieved 20 November 2020, from https://en.wikipedia.org/wiki/Chicago_Air_Shower_Array*

[6] *Correlation and dependence. (2020). Retrieved 12 November 2020, from https://en.wikipedia.org/wiki/Correlation_and_dependence*

[7] *DBFS. (2020). Retrieved 9 November 2020, from https://en.wikipedia.org/wiki/DBFS#:~:text=Decibels%20relative%20to%20full%20scale,relative%20to%20overload%20(dBO).*

[8] *Deployment and calibration procedures for accurate timing and directional reconstruction of EAS particle-fronts with HELYCON stations. (2020). Retrieved 6 November 2020, from https://arxiv.org/pdf/1702.04902.pdf*

[9] *Pigeon, D. (2020). Custom Chirp and Sweep Tones | Audio Test File Generator. Retrieved 6 November 2020, from https://www.audiocheck.net/audiofrequencysignalgenerator_sweep.php*

[10] *Royon, C. (2015). Timing detectors in particle physics and applications. (2020). Retrieved 12 November 2020, from https://indico.cern.ch/event/396441/contributions/1836621/attachments/794751/1089383/prague2015.pdf*

[11] *Extremely Accurate I2C-Integrated RTC/TCXO/Crystal (2020). Retrieved 20 November 2020, from https://datasheets.maximintegrated.com/en/ds/DS3231.pdf*

[12] *X. Vilasís-Cardona, "Cellular Neural Networks for high energy physics," 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010), Berkeley, CA, 2010, pp. 1-6, doi: 10.1109/CNNA.2010.5430343.*