Statistical Time-of-Arrival Ranging by Measuring Round-Trip Time at Driver Layer in IEEE 802.11g Networks

by

Amogh Kashyap

A Thesis submitted to the Graduate Faculty of
Auburn University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

Auburn, Alabama
August 5, 2017

Approved by

Alvin S. Lim, Co-Chair, Professor of Computer Science and Software Engineering
Shiwen Mao, Co-Chair, Professor of Electrical and Computer Engineering
Thaddeus Roppel, Associate Professor of Electrical and Computer Engineering

ABSTRACT

Alternative indoor localization methods have long been researched due to the unsuitability of GPS and cellular technology in indoor environments. Indoor localization systems find their applications in many areas such as targeted advertising, inventory management and tracking, navigation in museums, arenas, hospitals and in first responder and disaster situations like fire. Wireless localization systems have become very popular in recent years, especially the ones based on Wi-Fi due to its existence in almost any indoor environment today. Our indoor localization system is based on Wi-Fi (IEEE 802.11g) and can be installed on existing wireless devices with a simple software patch. This is very attractive since there are a hundreds of millions of wireless devices already in use and a hardware upgrade would be impractical. Our indoor localization system consists of a target node unaware of its location and several reference peer nodes which are location aware. Target nodes communicate opportunistically with each reference node to measure the Round-trip time (RTT) between the transmission of data frames and reception of acknowledgement frames. This RTT is measured at the driver layer of the OS kernel. These RTT measurements are used to predict the distance between the reference and target nodes by using a Statistical TOA ranging method which compares the measured RTT against stored reference databases collected during an offline stage. The comparison based on a statistical distance measured called Bhattacharyya coefficient and the results of this new ranging method is fed to a position estimation algorithm based on Linear Matrix Inequality (LMI) to provide the position coordinates of the target node. This thesis describes this new indoor localization system followed by a thorough analysis of RTT measurements made in different indoor environments such as a hallway and multiple rooms. We compare the measured RTT data with the packet timings involved in the IEEE 802.11g standard and introduce our novel idea of Statistical TOA ranging.

We discuss how we can build reference databases effectively by using Bhattacharyya coefficient and finally, we provide some real world experimental results of the ranging method along with the Wireshark view of the DATA and ACK frames involved. The real world experiments have shown that the precision of Statistical TOA ranging is 10 metres, i.e. there is a significant difference between RTT distribution for distances of 10 metres and it is extremely difficult to predict ranges less than 10 metres accurately. Hence, our Statistical TOA ranging predicts if the range is either 10 m, 20 m or 30 m. If the actual range is say 14 m, Statistical TOA ranging result is 10 m since it reduces the overall error. The position estimation stage then constructs LMI equations by incorporating ranging errors. From our real world, real time experiments, we find the average error to be about 3.08 metres and percentage of accurate predictions is about 71%.

“It isn’t the mountains ahead that wear you down. It’s the pebble in your shoe”

–Muhammad Ali

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Alvin Lim for his support and guidance throughout my study here at Auburn. His teachings and passion has imbibed in me a great sense of wonder and curiosity to find answers to the unsolvable. I would also extend my gratitude to my advisory committee members Dr. Shiwen Mao and Dr. Thaddeus Roppel for their teachings and valuable inputs to this thesis work.

It would be remiss on my part if I do not thank my research mates Abhishek Kulkarni, Song Gao and Yufei Yan for their contributions to this research project. Our constant discussions, dedication and hard work has made this project come into fruition.

But most importantly, I would like to thank my mother and sister for their unwavering support and guidance, and my father for always believing in my dreams and abilities.

Table of Contents

List of Figures

List of Tables

Dedicated To My Guide, My Guru - My Father. . .

Chapter 1

INTRODUCTION

1.1 GPS and its History

GPS was initially designed for military applications and purposes after the Soviet's launch of the Sputnik spacecraft in 1957. In the early 1960s, several U.S. government organizations, including the Department of Defense (DOD), the National Aeronautics and Space Administration (NASA), and the Department of Transportation (DOT), were interested in developing satellite systems for three-dimensional position determination. The optimum system was viewed as having the following attributes: global coverage, continuous (all weather) operation, ability to serve high-dynamic platforms, and high accuracy [26]. The first satellite navigation system was called TRANSIT which was built by the US Navy in collaboration with the Applied Physics Laboratory (APL) at John Hopkins University. At the same time as the Transit enhancements were being considered, the Air Force conceptualized a satellite positioning system denoted as System 621B [26].

Finally, in 1969, the Office of the Secretary of Defense (OSD) established the Defense Navigation Satellite System (DNSS) program to consolidate the independent development efforts of each military service to form a single joint-use system. From this effort, the system concept for NAVSTAR GPS was formed [26]. It was later termed '*GPS*'. In 1983, after the Soviets shot down a Korean Air flight after it wandered into Soviet airspace, the then President Reagan's administration allowed all civilian aircraft to use the GPS system to improve air safety [44]. This was the first-time GPS was used beyond military applications, but still did not make its way to the general public.

The DoD dutifully carried out Reagan's instructions to make GPS signals available for civilian uses, but at first, they added random timing errors to the satellite signals accessible to nonmilitary

GPS units so they could determine locations to no better than 100 meters. Then in 2000, after President Bill Clinton ordered this purposeful degradation to be stopped, the error circle shrank to 10 meters or so. All of a sudden, GPS became extremely valuable for vehicle and even pedestrian navigation [40].

GPS is a 24 satellite system arranged in 6 orbital planes, 4 in each plane such that 4 satellites are visible to a receiver on Earth at any given time. To determine the location of a receiver, the satellites transmit radio waves along with their time (atomic clocks). The receivers then determine the Time of Flight (ToF) from each satellite and determine the location by using a technique called trilateration. The 4th satellite's data is used to correct the timing errors on the receiver.

1.2 Location Based Services

In recent years, the use of smartphone or more generally smart devices has greatly risen. Each of these smart devices have equipped with them a GPS receiver which enables them to decode GPS signals and thereby determine their position (location) on the Earth. This has opened up to a wide range of location based services (LBS) which is a software service that uses location data to provide information services to the users in areas such as health, entertainment, marketing, work, personal life (fitness regime) etc. LBS for example, can be used in roadside assistance, store or restaurant locators, targeted advertising and fraud prevention among others. Location based services could also be critical for government organizations in matters of national security, relief and disaster managements; to commercial businesses- to analyze human behavior and interests and determine their business strategies based on that data [52].

1.3 Pitfalls of GPS systems and Alternate Localization Methods

The location (position) of the user can be found accurately not only using GPS, but another service called Wireless Enhanced 911 (E-911). E911 is an ambitious and arguably much-needed national safety net for locating wireless callers via satellite global positioning or cellular tower triangulation technologies on a local, state, and national level [16]. However, the single most deterring factor of these technologies is that radio signals coming from distant satellites are heavily attenuated when the view of the sky is obstructed, which makes navigating in narrow canyons, urban areas, extremely tough and inaccurate. And these high-frequency signals bounce around so much when they hit metal that getting a good GPS fix indoors usually proves impossible [40]. GPS signals are attenuated in indoor environments and the low power levels affect the receiver coverage. Also, GPS does not provide height information i.e. we cannot determine on which floor the user is in a multistoried building. These shortcomings do not make GPS and Wireless E911 a viable option for indoor environments and researchers have been studying new technologies since the 1990s.

Indoor Positioning systems or Indoor geolocation systems is a system to locate the position (including the height information) of a user inside a building such as a mall, hospital, office. In GPS, the communication between the receivers and the satellites is through radio waves. But in Indoor positioning systems, there are a host of different technologies proposed. These include Bluetooth, Wireless Local Area Network (WLAN), Vision based (camera) [34], Ultrasound, Infrared, RFID. Bluetooth and WLAN still use radio waves but they are transmitted at different frequencies compared to the GPS signals. Although lot of research have been carried out in this field in the past two decades, these systems haven't been standardized yet [36].

1.4 Wireless LAN based Indoor Localization System

Ever since the IEEE 802.11 Wireless LAN (WLAN) standard was first published, wireless networks are configured almost everywhere right from university campuses, corporate buildings, malls, homes to streets. Every device today is capable of exchanging information over such wireless networks with small latency. The number of WLAN access points are in hundreds of millions which is significantly larger when compared to the 24-31 GPS satellites and hundreds of thousands of cellular towers (E911). These Wi-Fi APs can be leveraged opportunistically to determine the user's location in indoor environments or where other technologies are either not accurate or have large latency periods [37]. These advantages inspired us to build a wireless indoor localization system with the WLAN technology at its forefront.

Wireless networks are of two types- AP (Access Point) or the Infrastructure mode and Ad Hoc mode. IEEE 802.11 WLAN standard terms this as the BSS (Basic Service Set) and IBSS (Independent Basic Service Set) modes respectively. The infrastructure mode consists of a centralized Access Point (AP) and all nodes communicate through the access point. The APs are fixed and can connect to all nodes that are within its range. An adhoc network on the other hand is a peer to peer network. Each node can directly communicate with other peer nodes. The main advantage of the adhoc network is the minimal configuration and quick deployment, making it suitable for disaster and rescue situations since it is too cumbersome to establish centralized access points and there is a need for “on the fly” network creation [55]. It is due to this flexibility; we chose a wireless adhoc network as our mobile networking system.

Different applications may need different types of location information. Some types discussed in [35, 19] are physical location, absolute location, symbolic location and relative location. Physical location can be identified on a map and is expressed as coordinates. Symbolic location is less specific and expresses a location in a more abstract way such as in the kitchen, bedside the TV etc. Absolute location uses a shared reference grid for all located objects while relative location information is usually based on the proximity to known reference points or base stations [35, 19].

An Indoor wireless positioning systems consists of a target node which is unaware of its position and several reference peer nodes who are aware of their location coordinates. This is like a GPS system in which the reference nodes are the satellites and the target is the GPS receiver. In the indoor system, the reference nodes maybe for example an access point (AP)/ router which is aware of its location and the target node could be a smartphone or any handheld device. The communication between the two is through radio waves. Just like in GPS, determining the position in an indoor environment is twofold- first we need to determine where the target is with respect to the fixed references (distance or angle) a technique called as ranging, and second- the position coordinates of the target node through geometry techniques such as trilateration or triangulation.

Wireless indoor positioning systems use different ranging methods to determine the range between the target and the reference nodes. Lateration uses distance as the metric to determine the range between the target and the reference, while Angulation estimates the angles relative to the peer reference nodes.

In lateration techniques, received signal strength (RSS) or the Time of Arrival(TOA) of the radio waves is used to estimate the distance and in angulation, the Angle of Arrival(AOA) of radio waves is measured. RSS based systems estimate the distance based on signal attenuation since the strength of the radio waves decreases over distance [11]. The distance can also be estimated by measuring the propagation time of the radio waves from the transmitter to the receiver. This would require the two nodes to be accurately synchronized with each other and the timestamp has to be included in the transmitted signal. This requirement is less significant if the Round Trip Time (RTT) is measured at the transmitter side, but this would include the processing times at the receiver leading to less accurate estimations. AOA utilizes sophisticated directional antennas or an array of antennas and the location of the target is estimated by determining the angle of incidence at the receiving antenna [1]. In addition to the above mentioned localization methods, another technique called Fingerprinting using the signal strengths has been widely researched [34, 8]. The RSS values are measured continuously at each point and stored in a database to create what is called a radiomap. These values at each position are called fingerprint of that location. To determine the location of a target node, the real time RSS measurements are matched to the known observations (database) [8]. This overcomes the errors induced by multipath (of wireless signals) conditions that a traditional RSS technique would encounter.

In most of the above cases, the indoor positioning system needs specialized hardware to accurately measure either the signal strength or the propagation time of the wireless signals. This raises the overall cost of the positioning system. Some systems use cellular base stations to determine the location, but this would not be suitable in disaster and relief operations as there is a high probability of them being destroyed and non-operational. The cost can be significantly reduced if the reference and target nodes can be made available for wireless positioning with a simple software update rather than replacing the hardware such as for example- using sophisticated directional antennas as in the case of AOA technique. In other words, there is a need for opportunistic localization where the target and reference nodes can be existing devices such as smartphones and they are connected quickly “on the fly” in an adhoc network. To fulfill these needs we constructed a new opportunistic wireless localization method.

Our wireless indoor localization method consists of a target node who wishes to determine its location by communicating with location aware reference peer nodes. This communication between the nodes takes place in an ad hoc network over Wi-Fi using the IEEE 802.11g [50] communication protocol over 2.4 GHz frequency band. The localization process is twofold- the target node first detects the distance between itself and each of the reference nodes through ranging; secondly the target node then calculates its position using Linear Matrix Quality and Grid Method [48, 29]. We use Time of Arrival (TOA) ranging over the received signal strength (RSS) since it has its own limitations on accuracy in the indoor space [14]. Other ranging methods like AOA are an expensive affair with its high precision antenna requirements. Most of the literature focuses on new hardware and software to make indoor localization possible, but since there is already a plethora of wireless devices in use, an easier way to make a device localization ready would be to introduce a software update. This inspired to us to delve into the device driver layer of OS kernel to collect the TOA data. We use the FreeBSD operating system which is derived from the BSD version of UNIX [2] since it is easy to use, understand and modify device driver programs to collect TOA data. More specifically we use the RTT (Round Trip Time) as the time taken for the target device driver to send a UDP packet and receive an ACK from the reference node. This is used to determine the range (distance) between the target and reference peer nodes over TOA because it eliminates the requirement of synchronized clocks between the transmitter and receiver and enables all the processing to be done on the target. This would aid debugging. In this way, the TOA (RTT) data profile (probability density functions) is collected for different distances in different indoor environments to build a database. The real time RTT measurements are then matched to known observations (database) to determine the distance between the target and reference nodes. This is similar to the RSS fingerprinting, except that in this case, the database is profiles of time (RTT). This way, we can incorporate the multipath conditions which directly affects the Time of Arrival values since the RF signals bounce of several objects to arrive at different times at the receiver.

This thesis document mainly deals with the analysis of RTT measurements made at the driver layer of the OS kernel and introduces a Statistical TOA ranging method to determine the range between the target and reference nodes. The remainder of this thesis document is structured as follows. In Chapter 2, we provide a note about what motivated us to build a Wireless Indoor Localization System followed by the applications of our system in the real world. In Chapter 3, we provide a literature survey of several research works and ideas that are related to our technology for ranging and localization in indoor environments. In Chapter 4, we define the problem statement and introduce the different parts of our indoor localization system. In Chapter 5, we explain the Round-Trip Time (RTT) Samples collection methodology that is based on [64], followed by a detailed analysis of the RTT samples measured in different indoor environments. We introduce in Chapter 6, our Statistical TOA ranging method based on a statistical distance measure called Bhattacharyya Coefficient and Distance while also providing a model for the RTT data samples measured. This ranging method is used to determine the distance between the reference and target nodes and in Chapter 7, we evaluate the performance of our statistical TOA algorithm by performing real-world experiments. Finally, Chapter 8 draws the conclusion and mentions the future work.

Chapter 2

Motivation and Application

2.1 Motivation

The commercialization of GPS in 2000s combined with the technological advancements has brought cheap and accurate GPS receivers into every wireless device. Thus, GPS was extensively used for not just navigational purposes but also gave rise to Location Based Services, where businesses could offer unique services to customers based on their location. Users could also search for nearby restaurants, hospitals, malls and movies. Credit card transactions can be matched with user's location to provide an additional level of security. In an indoor setting, however, due to the signal attenuation caused by the roof, walls of the buildings, GPS is not suitable. Researchers have been investigating an alternate wireless indoor positioning system since late 1990s, that can provide a high level of accuracy. The accuracy requirement is high especially in indoor scenarios, since a small location error of a few meters can end up being in a different room or apartment altogether. Some indoor positioning systems based on Ultra-Wide-Band technology provide good accuracy but are very expensive to deploy. Other indoor tracking systems rely on existing infrastructure like cellular base stations, but they are not suitable during relief operations where there is a high probability of damaged infrastructure. This motivated us to develop a new technique to solve the indoor positioning system which is both accurate, cost effective and completely based on wireless ad-hoc networks which makes it suitable during relief operations. Our Wireless Indoor Localization Solution can be split into three stages- RTT data collection, TOA Ranging, and Position coordinates estimation. First, we measure the Round-Trip Times (RTT) at the driver layer of the OS kernel (operating system) which forms the basis of TOA ranging. This is the time taken for the data packets to be sent and the ACK (acknowledgement) to be received. We choose TOA ranging over RSS (Received Signal Strength) ranging because of the its unreliability. Second, we match the RTT data against a stored RTT databases for different distances using a statistical distance measure called Bhattacharya distance. This measures the similarity between the probability density functions of the RTT data. It is at the end of this stage, that we can determine the distances between the target and each of the in-range reference peer nodes. Finally, we determine the location coordinates of the target node by using a GPS trilateration type technique like LMI (Linear Matrix Inequality). The accuracy is further improved using Center of Gravity (CoG) technique.

2.2 Applications

Alternate Positioning and Tracking systems have long been researched because of the inability of GPS to be used in indoor settings. These indoor positioning systems find their use in complex environments like university halls and buildings, hospitals, malls. The following sections briefly describe in what settings our proposed Indoor Localization System can be used.

2.2.1 Hospitals and University Buildings

Hospitals usually have large buildings with several doctors each with a different specialty. It is always a challenge for patients to locate their offices in time for an appointment in such a large setting. For parents, to carry their babies all around the hospital in search of nurse and doctors is always difficult and it gets worse in the middle of the night, when the hospital has fewer personnel to seek assistance. In a university setting, most students tend to miss classes because they cannot locate a classroom. Universities prefer an interdisciplinary approach to their education system, with students taking classes from different departments. So, an electrical engineering student will have several classes in a computer science and mechanical departments which he/she is unfamiliar with. They quite often end up spending a lot of time in search of buildings and classrooms. In both these cases, it would be a boon to be able to locate yourself inside the building and get directions to the exact room of interest. Our proposed indoor localization system is very flexible and can easily be deployed on existing wireless devices such as a smartphone or tablet and the patients, students can easily navigate to the exact room within a building.

2.2.2 Airport, Railways and Subway Stations

Airports, railway stations are very complicated buildings and the only navigation for the passengers is displays that indicate the terminal, gate or platform numbers for different planes or trains. In bad weather conditions, passengers can be caught unaware of schedule changes and mostly change in terminals or platforms. While there are several apps that notify passengers of such changes, they do not navigate them to the changed terminal. This can be greatly stressful in busy and huge airports with very little time left between the changes. Our indoor navigation system in such a setting can provide very useful as it can notify the passengers of the changes and guide them through the airport or railway station to the appropriate terminal or platform. This can be done via an app on their smartphone and the passengers can skip the displays which are always very messy. Other points of interest like ATMs, shopping and food areas that the users may be interested in can also be displayed on the indoor map. This indoor navigation system can also be used by security personnel to keep passengers out of sensitive and construction areas inside the building.

2.2.3 Office Buildings and Warehouses

Our indoor localization system can not only be used by and on human beings, it can also be used to track objects of interest. In a warehouse, it can be quite a task to search for misplaced equipment like barcode readers, forklifts, goods and robots. We could deploy our technology into such devices to easily locate them. Similarly, in an office setting, it can be used for tracking and preventing theft of assets like laptops, printers and conference room reservations, canteen finders and facilities management. Sometimes, office meetings are scheduled in different buildings, in which case our indoor navigation system can lead people to the exact conference room in time.

2.2.4 Museums and Exhibitions

Museums usually have plenty of exhibits and visitors are handed an audio device which gives them a tour around the museum. Users must key in the code of each exhibit for a brief explanation of the history of the exhibit. Our indoor navigation system can be used in museums in the form of a museum app to provide the visitors a better user experience. It could guide and navigate the users through sections of the museum the users are most interested in and they could listen to the audio playback on their own phones. In trade fairs, the navigation system can lead people to the exact booth, parking lots, food area, restrooms and trade fair support staff. The location data could also be used to show other booths with similar products. For exhibitors, the location data can be used to analyze the visitors' response.

2.2.5 Parking Lots

Hospitals, Malls, Office buildings usually have attached with them a multistory parking lot in an adjacent building or underground. On a busy day, it is extremely difficult to find an empty parking spot and visitors usually spend close to an hour to find parking. Sometimes there is no way to figure out if the parking is full without human assistance and it in the middle of the night, it is extremely rare to find support staff. Further, parking lots are divided into different sections-one for employees, one for pre-reservations, and one for the general public. While sitting in the car, it is sometimes very difficult to distinguish between these sections. Our indoor localization system could be used to alleviate this problem. Each parking spot could be fitted with a small sensor whose position coordinates are known and our system can navigate to that. As soon as you enter the parking lot, the user device is brought into the parking lot's network, an empty spot is assigned, and the app can navigate to it. Since users tend to forget where they have parked their cars, the app can then navigate the user to and from the parking lot.

2.2.6 Targeted Advertising in Malls, Retail Stores

In the past, advertising was in the form of fliers, huge hoardings on freeways, and in newspapers. With the rapid development of digital media, retailers are finding new ways to advertise through text messages, emails and Facebook. They even track what users search for online, to determine the most relevant advertisements for them. But such a system is limited to online shopping.

Our Indoor tracking system can be utilized for a new form of personalized targeted advertising when users are shopping in a mall or a retail store. They can track where the users tend to move around inside the mall, and navigate them to the nearest store with customized offers and deals. Similarly, when inside a store, the users can be notified of a similar product from another brand which is cheaper or more nutritious. This could attract more customers while also making for a great shopping experience.

2.2.7 Indoor Fire Fighting and Warning Alerts

The Worcester Cold Storage disaster in 1999, brought into light the need to be able to locate emergency responders to better coordinate rescue efforts and for the safety of the rescue personnel [58]. On that fatal day, a fire broke out accidentally in the cold storage and since, it was not reported quickly, by the time the firefighters could get to the scene, the fire had grown significantly. The firefighters were unfamiliar with the layout of the building, making it harder to extinguish the fire. It led to the sad demise of six firemen and their bodies could not be located for days. Our technology can be useful for locating firefighters inside the building. They could carry a small device like a cell phone (without complex cellular functionality) in which our technology can be deployed, and they can be tracked by the Fire Chief who usually coordinates the rescue operations from outside. The technology can also be used to send warning and alert messages to people guiding them to the nearest exit doors in case a fire breaks out.

2.2.8 Sensor Networks

Wireless sensor networks are used to measure physical and environmental conditions such as temperature, sound, pressure [56]. They consist of autonomous sensors in a network placed in remote areas, and they transmit sensed information to a central device wirelessly. The sensors can be anywhere between 100-1000s of them in remote areas and it can quite a challenge locating each one of them in case they need to be serviced. Our technology can easily be integrated into a Wireless Sensor Network and it can provide accurate locations of each sensors making maintenance a lot easier.

2.2.9 Stadiums

In large stadiums, the seating is usually divided into different sections and a spectator navigates to his/her seat using the section name and the seat number in that section. With people arriving into the stadium at different times, they usually walk past people already seated in search of their seat and on most occasions spectators find the exact number on the seat but in the wrong section. This can be quite inconvenient to both. Our indoor navigation system can navigate people right from the parking lot to their exact seat making it convenient to the users and the people already seated so that nobody misses any part of the event. It can also navigate them to the nearest confectionery, restrooms thereby providing a better fan experience.

Chapter 3

RELATED WORK

3.1 Alternate Localization Systems

Alternate localization systems have been researched since the late 1990s due to the unavailability of GPS in indoor environments, narrow canyons and underground. In such environments, obtaining position information of a target node is a challenge due to existence of multipath and Non Line-of-Sight (NLOS) conditions, large number of obstacles that cause signal attenuation and diffraction. This demands a localization system with very high accuracy, particularly in indoor environments since a few metres error might end up in a different room. Enabling technologies in these alternative localization systems have been detailed in [36]. Localization systems can be based on infrared, ultrasound, or wireless (RF) based such as WLAN (Wireless Local Area Network), Bluetooth and RFID. They can also be vision based where videos captured by cameras are processed and evaluated.

Camera based systems are usually employed in robotics to determine the location of a mobile robot while also simultaneously building a map of the environment and this is termed as SLAM (Simultaneous Localization and Mapping). In [41] for instance, certain features called scale invariant feature transform (SIFT) in images captured by the camera are extracted to serve as visual landmarks. These landmarks are tracked over a period of time are are used for robot position estimation and map building.

3.2 Wireless Indoor Localization

In the past decade or so, wireless technologies and networks have entered the realms of personal computing, data communication, consumer applications including medical, industrial, public safety and other applications. Due to the growth and large availability of wireless systems, indoor positioning systems based on wireless technologies are very attractive. The survey paper [35], lists several wireless technologies used for indoor localization.

3.2.1 RFID

Paper [31] is a location sensing system based on RFID. The LANDMARC system performs location sensing based on signal strength using active RFID tags. The methodology consists of RFID readers, reference tags to help in location calibration, tracking tags as objects being tracked. RFID readers measure the signal strength from reference and tracking tags and Euclidean distance between the two readings is computed. Weighted kNN (k-Nearest Neighbors) method is then used to determine the location of tracking tags with the weighting factor depending on the Euclidean distance.

3.2.2 Cellular Technology

In literature, there are some indoor localization systems based on cellular technology. GSM network is well spread and if there are several base stations around a building, they can be leveraged for indoor positioning. One such system is introduced in [47], where the authors uses the concept of wide signal-strength fingerprints. Fingerprints are collected from the six strongest GSM cells and readings from upto 29 additional GSM channels which are strong enough to be detected but too weak and unsuitable for reliable and efficient communication. These additional channels improve the accuracy. The overall accuracy is as low as 2.5 metres and the system can distinguish between multiple floors in a building.

3.2.3 Bluetooth

Bluetooth is a short range communication system (10-15 m) operating at 2.4 GHz ISM band. In literature, a lot of research has been conducted to utilize bluetooth in indoor positioning [38, 24]. Bluetooth has an adaptive technique to control the transmit power based on RSSI (Received Signal

Strength Indicator) to improve signal-to-noise ratio. In [38], this feedback was turned off and the RSSI was used as a distance indicator. They used a line-of-sight radio propagation model which is the free-space propagation model to determine the distance between two Bluetooth enabled devices. Results show that the system has an accuracy of about 1.2 metres but it does not work well in NLOS conditions. Similar work is done in [24] but the estimated position is represented in the form of probability density functions.

3.2.4 WLAN

IEEE 802.11 WLAN access points (AP) number in hundreds of millions today and are easily available in almost every indoor building. These Wi-Fi APs have a much larger communication range when compared to Bluetooth or RFID and hence they can be leveraged opportunistically to locate mobile nodes on the Wi-Fi network [37]. Xiang *et al.* [66] propose an indoor positioning system which is based on signal strengths of WLAN APs. It consists of a training phase, where the signal distribution of APs is collected to train a position-determination model. It is followed by a working phase, in which the mobile node measures the WLAN signals and applies the position-determination model to calculate a position. To simplify the training procedure, model-based signal-distribution training scheme is proposed where certain characteristics of signal distribution are exploited to reduce the number of samples to be collected. The working phase involves a tracking algorithm to assist the position estimation. Since the estimated position of a node cannot change drastically in a short period of time, position determination relies on both collected signal strength and knowledge of space topology. Results show that the authors were able to achieve an accuracy of 2 metres for static cases and 5 metres when devices were mobile. Similarly, Saha *et al.* [39], described a WLAN indoor positioning system in which a database of signal strength information for various locations is built and this information is used to determine which location a given test data comes from. To classify the test data, they studied a nearest neighbor classifier, back propagation neural-network and histogram matching.

3.3 Ranging and other Localization Systems

In an indoor localization system, the target node communicates with other sensors to perform ranging i.e. to measure metrics like approximate distance or direction of arrival of signal. Accordingly, Time-of-Arrival (TOA), Received Signal Strength (RSS), Time Difference of Arrival (TDOA) and Angle-of-Arrival (AOA) are some of the techniques used in literature. In [7], signal strength (RSS) measured by the wireless NIC is recorded at reference points (base stations) for different positions and orientations of the target node. The authors then used a linear-search algorithm to search through recorded data to find the closest match to the real-time data. AOA estimation and source classification using antenna arrays is discussed in [33, 45]. For measuring TOA in indoor environments, detection of LOS path is necessary and in [6], authors describe the use of Direct Sequence Spread Spectrum (DSSS) to track the leading edge of direct path signal. TDOA based methods having the advantage that the initial synchronization between transmitter and receiver is eliminated [32]. Several hybrid techniques to achieve indoor localization has been an active research area. In [23], authors measured signal strength from arrays of directional antennas on each sensor node in a way to combine TOA and AOA, to demonstrate localization in sensor networks. Radiolocation can be very important in wireless E-911 to locate the callers and in [65], AOA and TOA methods were combined to distinguish between LOS and NLOS paths to achieve radiolocation using a single cellular base station. Hybrid localization methods using TDOA and AOA measurements to determine a mobile user location in wideband CDMA cellular systems are discussed in [30].

Signal strength in indoor environments can exhibit an unpredictable and random behaviour due to varying nature of the wireless channel. Thus, they are more susceptible to noise, interference and they are not accurate for indoor localization. The limitations of signal strength based indoor localization systems is detailed in [14]. Instead, TOA based systems are widely used and a survey of such systems is presented in [20]. TOA based systems suffer from NLOS conditions whic gives rise to incorrect distance estimation. This property is modelled in [20], where estimated distance includes a noise and an NLOS term along with the actual distance. In [4], the authors analyze the

TOA profiles and indicate that in addition to LOS and NLOS classification, there exists further classification that depends on the channel profile and the characteristics of the direct LOS (DLOS) path. When the DLOS path can be detected and has the strongest amplitude, it is called dominant direct path (DDP). If DLOS path is detected but is not the strongest, it is called non-DDP (NDDP) and when DLOS path is not detected, it is termed undetected direct path (UDP). This classification can provide a deeper insight into wireless channel modelling for indoor localization. In [18], authors present a performance evaluation testbed to test accuracy of TOA based systems in real time in the presence of multipath and NLOS conditions. In [64], authors introduce a new method of TOA measurement where the measurement is made in software i.e. driver layer of operating system. Thus, it can easily be installed on existing wireless devices. The measured TOA includes the software processing and processing time of PHY layer in wireless NIC. Euclidean distance was used to distinguish between two TOA profiles.

Wang *et al.* [61, 62] propose a RSS fingerprinting method for indoor localization. They use the CSI (channel state information) to obtain more fine grain information on wireless channel than other RSS based schemes. In the offline training phase, deep learning is utilized to train all the weights of a deep network as fingerprints. In the online localization phase, a probabilistic method based on the radial basis function is used to estimate location. Similar work is done in [60]; BiLoc system is based on CSI fingerprinting and operates at 5 GHz ISM band. The CSI information of each OFDM subcarrier is measured over three antennas and a bi-modal data is obtained including average amplitudes over pairs of antennas and estimated AOAs. A deep learning network was used to train the bi-modal data and in the test phase, a Bayesian approach based probability model was employed for estimating position with bi-model test data.

Paper [63] is an indoor localization system based on AOA technique. CA2T is a cooperative antenna array technique which estimates all the arriving angles of multipath components on two APs using the common MUSIC algorithm. Geometric relationship between the angles is exploited to obtain the arriving angle of LOS component. The distance between the two APs and the user is determined using a free scale path loss model which includes shadowing effect. Finally, with distance, angle and position coordinates of two APs, the user location is determined.

In paper [59], the authors exploited user's mobility to improve localization performance. Distance was estimated based on RSS measurements and LMI (Linear Matrix Inequality) was used to estimate the user's location. The data from accelerates and gyroscopes of the smartphone provide mobility information which can be used to narrow down the constraint area of LMI.

Chapter 4

PROBLEM STATEMENT

As mentioned in chapter 1, GPS signals undergo severe attenuation by walls, roof, furniture in an indoor setting and this makes them highly unsuitable for localization of objects and people in an indoor environment. Researchers began to work on indoor localization systems by tapping into existing technologies like Infrared, WLAN, Bluetooth, Cellular among many more. WLAN or Wi-Fi based systems are more attractive over the other technologies because of its range and its ubiquitous nature in almost every indoor environment. WLAN access points number in hundreds of millions and every device is Wi-Fi enabled, so the technology can be deployed very easily in these devices. But in first responder situations like indoor firefighting, there is a need for opportunistic localization with the reference and target nodes must be able to communicate 'on the fly'. With that in mind, our indoor localization system is designed to use IEEE 802.11 WLAN technology with the target node communicating opportunistically with its location aware reference nodes in an ad-hoc network. In a broad sense, it has two phases as depicted in the Figure 4.1.

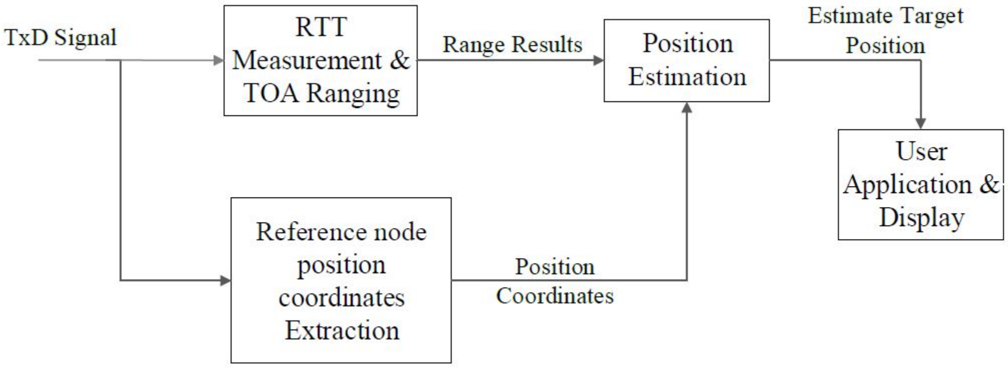

TxD Signal
RTT Measurement & TOA Ranging
Range Results
Position Estimation
Estimate Target Position
User Application & Display
Reference node position coordinates Extraction
Position Coordinates

Figure 4.1: Indoor Localization Architecture

In the first phase, the distance between the target and each of the reference nodes is determined by ranging. There are several methods of ranging like Received Signal Strength (RSS), Angle of Arrival (AOA), Time of Arrival (TOA). RSS ranging is less accurate and AOA needs sophisticated antenna configurations to achieve good accuracy which increases the cost and complexity. TOA technique needs no extra hardware and is generally more accurate. We measure the TOA or more specifically the Round Trip Time (RTT) at the driver layer of the OS kernel. This is very attractive because it can be deployed into any existing wireless device by means of a software patch without any hardware change.

In this thesis work, we analyze this RTT data in different indoor settings and show how the RTT data varies with different distances. Also defined is a technique to distinguish the RTT data at different distances. We employ a statistical distance measure called Bhattacharya Coefficient to quantify this differentiation. We compare this to the *Euclidean distance* measure to illustrate the advantages of the *Bhattacharya coefficient*.

In the second phase, we use the ranging data to determine the location coordinates of the target node by using a trilateration type technique like LMI (Linear Matrix Inequality). We also mention the second phase in this thesis as a sense of completeness since, the original idea was to build an indoor localization system, a system which was divided into 2 phases. One phase is discussed here and the other is discussed in detail in [29].

4.1 Conceptual Approach

The indoor localization system consists of a system of mobile nodes with the ability to communicate over radio access technologies like Wi-Fi, ZigBee, Bluetooth. Our localization however, is solely based on WLAN and hence we use devices with a Wireless NIC (Network Interface Card) interface. These network devices are either a target node or a reference node. A target node is one which is unaware and incapable of determining its location by GPS or Ultra-Wide-Band (UWB) techniques and seeks to determine its position opportunistically by communicating with reference nodes. These reference nodes have self-localizing capability with a certain accuracy which varies with time. But, in the real world, every node is a target since its unaware of its location inside the building. Each node will first try to determine its position by first communicating with certain fixed nodes which in theory can be fitted with GPS receivers (they are placed outside the building and can hence receive GPS signals). Once a node determines its position, it will then act as a reference to other target nodes.

The target and reference nodes are in an ad-hoc network in which each node can communicate with every other node without having to go through an Access Point (AP) thus removing the need for a centralized authority. This arrangement is very attractive in disaster and relief operations since ad-hoc networks can be configured quickly and a node does not need to be in range with the AP, it can instead be in range with some other node which can further forward the information using routing protocols such as OLSR (Optimized Link State Routing).

The target node first sends a 'hello message' to each of the reference nodes, to which the reference nodes respond to by sending an acknowledgment (ACK) frame containing its position coordinates, position error and time. The target measures the Round-Trip Time (RTT) by measuring the time between the transmission of a hello message and the reception of an ACK frame. This forms one RTT sample and the target collects several thousands of such RTT samples. This process is repeated with each reference node. To determine the distance, raw RTT samples are represented as a histogram and matched against a stored reference database for different distances (say 10, 20 m). The matching is done using a statistical distance measure called *Bhattacharya Coefficient*.

$$BC(p,q) = \sum_{x} \sqrt{p(x)q(x)} \tag{4.1}$$

where **p(x)** and **q(x)** are histograms of RTT data samples.

A high ***BC(p,q)*** value indicates that there is a strong match (similarity) between the real time RTT data and stored reference RTT data and a low value indicates less similarity. The range result (distance between target and reference nodes) is the reference database with the highest ***BC(p,q)*** value.

The position coordinates of each reference node along with the predicted range (distance) is then utilized to determine the position coordinates of target node by solving inequality equations using a technique called Linear Matrix Inequality (LMI) [13]. The accuracy of the predicted location can be further improved by taking Center of Gravity (CoG) of cluster of positions calculated by Linear Matrix Inequality [29].

4.2 Assumptions and Goals

To provide a proof of concept, we make a few assumptions. First, all the nodes; reference and target are stationary while communicating with each other. The target measures the Round-Trip Time when it receives the ACK frame for the data packet it sent to the reference. More detailed experimental set-up will be explained in the next chapters. Second, we assume there are at least three reference nodes which are in-range to the target. This requirement is similar to GPS where the receiver needs to see a minimum of 3 satellites to estimate its position. Third, all the nodes are equipped with a wireless NIC (Network Interface Card) capable of operating in IEEE 802.11a/g/n mode. Fourth, if a reference node is say 17 m from the target (actual distance), the most accurate TOA ranging prediction would be 20 m. This error is a part of the system and the subsequent stages of position estimation: LMI with Barycenter or Center of Gravity algorithm can reduce this error.

With that in mind, our aim is to determine the position coordinates of the target node by communicating opportunistically with in-range reference peer nodes. First, we measure the RTT samples at the device driver layer of the OS kernel by transmitting a data packet and waiting for an ACK frame. Second, the Statistical TOA ranging method involves comparing the measured RTT samples against stored reference databases to predict the distance between the two nodes. We then input this information along with the (x, y) position coordinates of reference peer nodes to the LMI algorithm which would then provide a position estimate. Finally, the Center of Gravity technique improves the accuracy of the estimated position over time.

4.3 TOA Ranging

As mentioned in chapter 1, researchers have used different ranging methods to determine the range between the target and the reference nodes. Received Signal Strength (RSS), Time of Arrival (TOA), Time Difference of Arrival (TDoA) are lateration techniques which measure distance while Angle of Arrival (AOA) is an angulation technique which measures angles of the received signals relative to the peer reference nodes. We perform Time of Arrival (TOA) ranging due to its reliability and low cost. Most research focuses on measuring the TOA at the hardware layer by using special devices that mimic the conditions of an indoor environment [18]. We on other hand, implement the TOA ranging and more precisely RTT (Round-Trip Time) data sample collection at the driver layer of the FreeBSD kernel or MAC layer of the TCP/IP protocol stack. Each node is configured to operate at 2.4 GHz and IEEE 802.11g mode. More details of how this done is explained in the later sections. Several hundreds of Round-Trip Time data samples are measured by repeatedly taking the time difference between the transmission of a data packet and reception of the ACK frame. RTT profile is constructed with these samples for different distances in different indoor environments and stored as a reference database for future range prediction. This is like the RSS fingerprints. Some noteworthy research work in RSS fingerprinting is in [36, 25, 7].

The distance between the reference and target nodes is predicted using a statistical distance measure called Bhattacharya coefficient. TOA ranging suffers from multipath in indoor environments and requires precise time synchronization. But this problem can be alleviated using the RTT sample profiles. The figure below shows our proposed TOA ranging architecture. In this thesis work, we analyze the RTT data samples collected at the MAC layer and create a reference database of RTT samples for different distances and different indoor environments. This is then utilized to determine the range between the target and reference nodes.

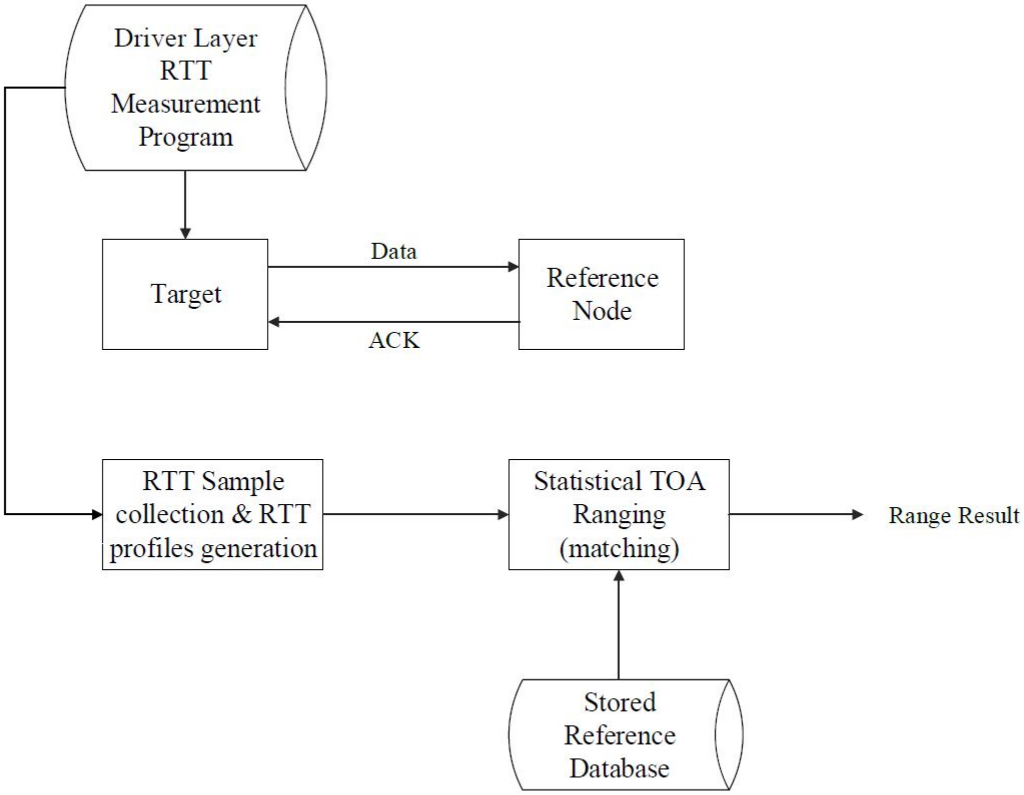

Driver Layer RTT Measurement Program
Target
Data
ACK
Reference Node
RTT Sample collection & RTT profiles generation
Statistical TOA Ranging (matching)
Range Result
Stored Reference Database

Figure 4.2: Statistical TOA Ranging Architecture

4.4 LMI

The last part of our indoor localization system is the Position Estimation block which takes as input-position coordinates of each reference node and the TOA ranging result to estimate the position coordinates of the target node. Finding the target node position estimate can be thought of as an optimization problem where we look to determine the most feasible solution satisfying certain constraints. Linear Matrix Inequality (LMI) [48, 13] is a constraint that can be solved using convex optimization algorithms and MATLAB's robust control toolbox provides excellent support.

Consider a reference peer node $\boldsymbol{i}$ to be at the center of a circle whose radius is equal to the range (distance) $\boldsymbol{R_i}$ between the target and reference node. So, we can say that the target position can be anywhere on the circle in an ideal scenario with no range errors, but in all practical scenarios, the target node lies anywhere within the circle. Similarly, the target should lie anywhere in the circle with some other reference node at the center and radius as the range. Thus, the target node lies in a region that is bounded by the intersection of all the circles.

Let $\boldsymbol{e_{ri}}$ represent the range error between the target and reference node $\boldsymbol{i}$ and $\boldsymbol{P_T}$ and $\boldsymbol{P_i}$ represent the target and reference node positions respectively. Thus, for reference node $\boldsymbol{i}$, we can write:

$$(R_i - e_{ri}) \leq |P_T - P_i| \leq (R_i + e_{ri}) \tag{4.2}$$

Equation 4.2 provides a constraint and can be written for each reference peer node $\boldsymbol{i}$ to form a Linear Matrix Inequality (LMI). Thus, we are trying to find a feasible solution or more specifically, target node position estimate that satisfies the LMI system created by the constraint in Equation 4.2.

The accuracy is improved over time with repeated communication with the same reference nodes and other reference nodes which may have now come in range with the target, to obtain new position estimates. Finally, we take the center of gravity of the new position estimates as:

$$C = \frac{\sum_{j=1}^{N} P_{T_j}}{N} \tag{4.3}$$

This is explained in detail in [29].

Chapter 5

ROUND-TRIP TIME (RTT) SAMPLE COLLECTION AND ANALYSIS

In this chapter, we discuss the RTT sample collection which is the first stage to TOA ranging between the reference and target nodes as shown in Figure 4.1. RTT samples are collected in a variety on indoor settings such as hallways where the target and reference are in a Line of Sight; multiple hallways with no Line of Sight; and target and reference nodes in different rooms.

5.1 RTT Data Sample Collection

RTT can be measured at different levels, like the hardware, firmware driver, OS or application [64]. Research shows that existing TOA ranging methods measure the time at the hardware which is the physical layer of the protocol stack using sophisticated devices which are usually very expensive [18, 4]. These techniques involve measuring the time taken for a signal to reach the reference node, but we measure the Round-Trip Time (RTT) at the device driver layer of the OS kernel using the Data packets and ACK frames. This is easier than to measure at (say) firmware level since most wireless network interface cards' firmware in a mobile node is proprietary and is difficult to access thus making it less flexible. The target sends a Data packet to the reference node and measures the time taken for the corresponding ACK frame to be received to give one RTT sample. This process is repeated for several thousand RTT data samples.

5.1.1 Experimental Setup

To test our concept in the real world and collect RTT data samples, we use a simple mini-PC with the configurations given below in the Table 5.1. We install a Wireless NIC and configure it to operate at 2.4 GHz, and in the IEEE 802.11g mode. Both the target and reference nodes have the exact same configuration.

Table 5.1: Test Model Specifications

Processor	Intel Atom D510 and N270 i386
CPU Frequency	1.6 GHz
RAM	2048MB (2GB)
OS Kernel	FreeBSD 10.1-RELEASE i386
32 bit/64bit	32bit
Wireless NIC	Ubiquiti Networks SR-71 (mini-PCIe)
WLAN protocol	IEEE 802.11 b/g
Operating Frequency Range	2.4 GHz
Data Rate	24 Mbps(OFDM)
Antenna Characteristics	TRENDnet Dual-Band 11a/g 7/5dBi Indoor Omni Directional Antenna (TEW-AI75OB)

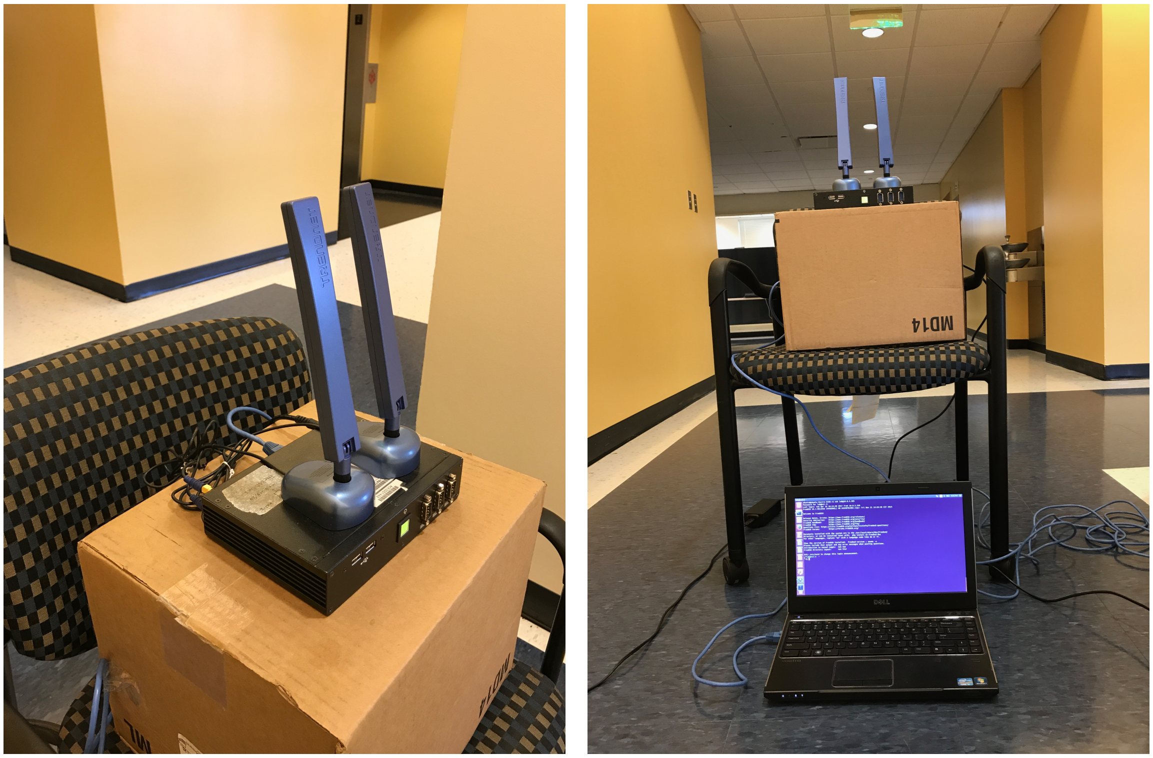


Figure 5.1: Target Node

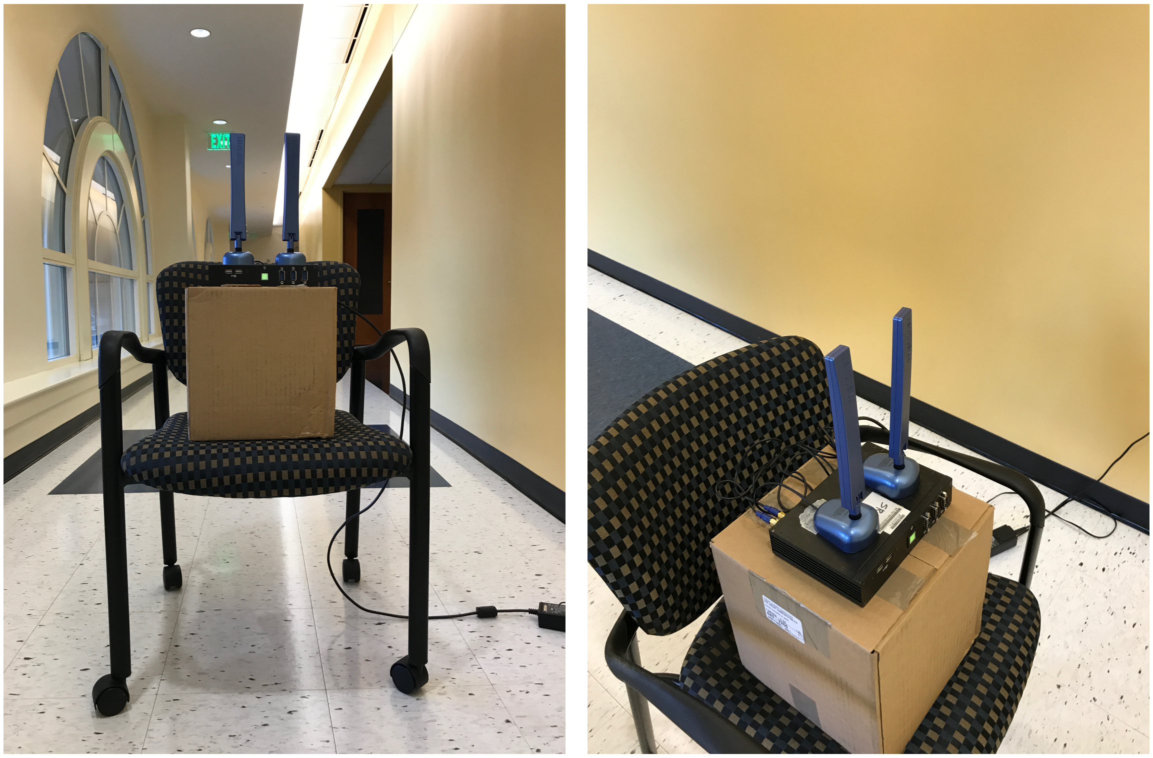


Figure 5.2: Reference Node

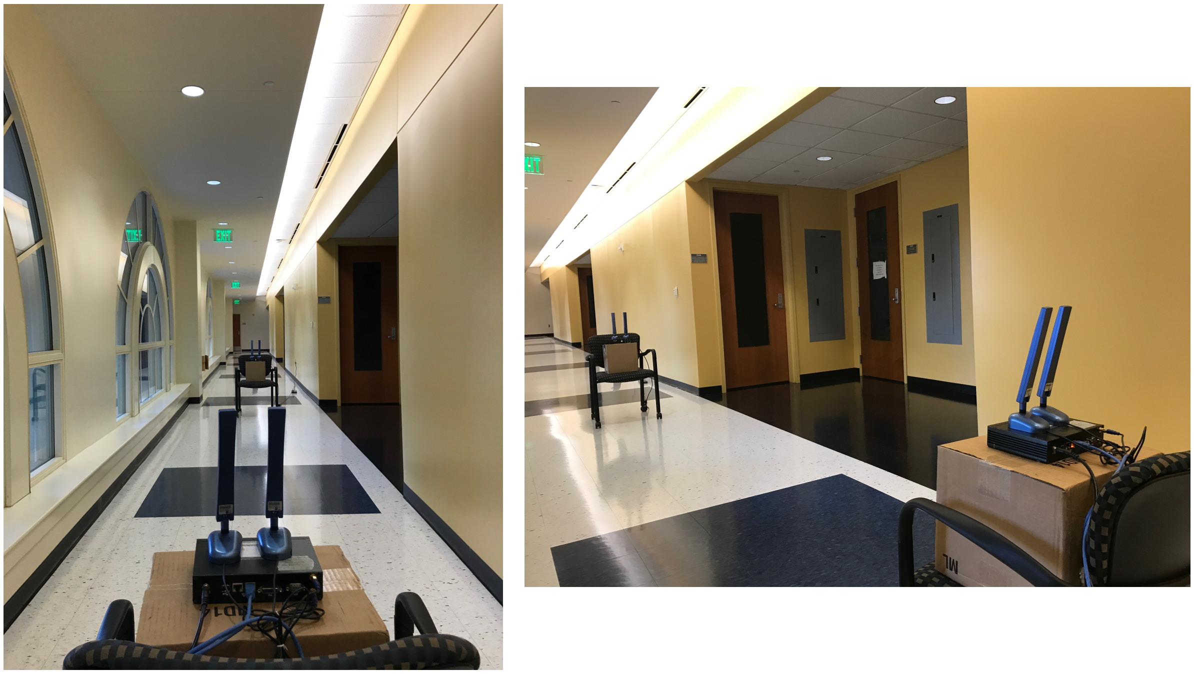


Figure 5.3: Experimental Setup

Figure 5.1 shows the target node connected to a laptop by means of the Ethernet port. We log into the node using a terminal on the laptop and run commands to collect RTT samples. Figure 5.2 shows the reference node and Figure 5.3 shows the experimental setup to collect RTT data samples.The target and reference node in Figure 5.3 were placed 5 m apart in Line-of-sight(LOS) condition. We also move the reference nodes to 10, 20, 30 m LOS and non-LOS conditions to collect RTT samples. We placed the device on a box on top of a chair to simulate a real world condition where the users hold their phones at that height.

5.1.1.1 A Note on FreeBSD Kernel

FreeBSD is a UNIX based operating system, an open source form of the BSD platform developed and distributed by the Computer Systems Research Group (CSRG) at University of California, Berkeley. BSD stands for Berkeley Software Distribution and it is available on a variety of platforms such as the Intel's i386, ARM and AMD's amd64. What makes FreeBSD attractive is that it provides a huge number of libraries and applications and has advanced networking features especially for IEEE 802.11 protocol. We can conveniently alter IEEE 802.11 modes, setup infrastructure and ad-hoc networks and the best of all-it's completely free [2, 49, 15].

5.1.1.2 iperf

As mentioned in the previous sections, the target node transmits a 'hello message' to the reference node to which it responds with an ACK frame containing its position information, position and timing errors if any. The target then estimates the range (its distance from the target) using statistical TOA method and calculates its position using the range estimate, position information of the reference by LMI technique. In this thesis work, to illustrate the RTT sample collection and statistical TOA ranging, we allow the target to send a simple DATA packet with a random payload to the reference node to which it responds with an ACK frame with no payload. So essentially we transmit a small UDP frame as the DATA packet. This is made possible with a simple application called '**iperf**'.

iperf is a network testing tool that can measure its quality and bandwidth to either test, optimize or tune the network. iperf requires the application to be installed on both the server and client and the throughput can be measured either unidirectionally or bi-directionally. With iperf, one can perform either a UDP test or a TCP test in which UDP packets or TCP packets are transmitted across the network. UDP tests can measure jitter and datagram losses, while a TCP test can measure bandwidth. It is an open source software that is available for Windows, Linux and Unix based platforms [22, 51].

The current version of **iperf** is 3.0, however we utilize version 2.0.5 since it requires **iperf** to be run only on the client side. We start the **iperf** on the target to send DATA packets, and the reference node responds to this DATA packet with an ACK frame. This is more realistic since it is the target node that is attempting to self-localize and hence he initiates the DATA packet transfer. The reference node simply responds to it while carrying out its own local tasks. It is important to know that the newer versions are NOT backward compatible with **iperf 2.0.5**. We could alternatively, use other versions but we must be careful to start the **iperf** on both the target and reference nodes (client and server).

5.1.1.3 ATH9K driver

Our method of TOA ranging involves measuring the Round-Trip Time (RTT) in the IEEE 802.11 driver layer of the FreeBSD kernel. In our experimentation, we use one of the most popular open source drivers called ATH9k. It is a kernel driver supporting Atheros 802.11n PCI/PCI-E chipsets [5], and compatible with Ubiquiti Networks SR-71 wireless card which is installed on the Mini-PC.

5.1.2 RTT Samples Collection Methodology

This subsection defines how an RTT sample is collected at '**ath9k**' driver layer of the FreeBSD kernel using **iperf**. It is inspired from the work in [64] whose author Dr. Alvin Lim is the head of our research group at Auburn University. This thesis work uses this RTT sample collection methodology to analyze the resulting RTT samples, and introduces statistical TOA ranging.

The target and reference nodes are configured to be in an ad-hoc network and the target node communicates opportunistically with in-range reference nodes to determine its range (distance) from each of them. The target transmits a DATA packet and measures the time when the ACK is received. The difference yields the round-trip time. IEEE 802.11 standard defines several request/response packet frame pairs such as Probe Request/Response, Request to Send (RTS)/Clear to Send (CTS) but we choose DATA and ACK frame pair since it is always present in all IEEE 802.11 networks. The other request/response pairs can be turned off either manually or automatically depending on network conditions and requirements.

Since, we measure the RTT at the driver layer, we cannot measure the time when the DATA packet was transmitted over the air interface. Instead we record a timestamp $\mathbf{t_s}$ when the DATA packet is passed on to the hardware layer from the kernel, and a timestamp $\mathbf{t_a}$ when the ACK is received. So, the round-trip time is,

$$RTT = t_a - t_s \tag{5.1}$$

These timestamps are introduced in the MAC layer, which is usually implemented in the driver layer such as ATH9K. The driver in the OS kernel, logs the cycle count whenever an event occurs. We edit the driver to log the cycle count of three important events- TX, TX_INT and ACK. TX is when *ath* driver is about to hand off DATA to lower (PHY) layer for transmitting. TX_INT is an interrupt that can happen in several situations, one of which is an acknowledgment being received. ACK happens when that IEEE 802.11 MAC layer acknowledgement is received and verified. However, TX_INT can occur for other events apart from the ACK and hence only TX_INT followed by ACK should be considered. Thus,the occurrence of the ordered tuple of (TX, TX_INT, ACK) is used to determine the Round-Trip Time (RTT) sample as the difference between the timestamps of TX and TX_INT [42].

The cycle counts are measured using the TSC (Time Stamp Counter) which is a 64-bit register present on all x86 processors that counts the number of cycles since reset. The events and their corresponding cycle counts are then logged using the *printk()* function. We use the frequency of the clock (CPU frequency) to convert the cycle counts into time in nanoseconds.

5.2 Analysis of RTT Data Samples

The measurement and monitoring of network round-trip time (RTT) is important as it allows network operators and end users to understand their network performance and help optimize their environment. Measuring network RTT is also important for Transmission Control Protocol (TCP) stacks to help optimize bandwidth usage. TCP stacks on end hosts optimize for high performance by passively measuring network RTT using widely deployed TCP timestamp options carried in TCP headers [43]. In our case, we measure the RTT values at the driver layer of the OS kernel as the time between the transmission of a DATA packet and the reception of the ACK frame. We use this RTT values to determine the range (distance) between the target and reference nodes.

5.2.1 Sample Size

In [64], the authors introduce a model to explain the RTT values measured at the driver layer. We use this model in this thesis work, but we substitute the RTT values we have measured through our experimentation, to explain the required number of RTT samples to perform our proposed ranging method-Statistical TOA ranging.

The round-trip time measured at the driver layer is greater than the actual round-trip time. Actual Round-Trip Time is total time of flight of the transmitted pulse over the air from the transmitter to the receiver and back to the transmitter. This can be considered as a non-random signal and hence ideally should have a constant value. However, since we use the methodology in [64], the RTT value measured at the driver layer is more than the actual RTT value and hence modelled as:

$$t = \tau + n \tag{5.2}$$

where **t** is the *RTT measured at driver layer*, τ is the *actual RTT* and **n** is *noise*.

The noise is due to the fact that wireless signals undergo reflections. In addition, the operating system processing time, time to process DATA and ACK frames at either end and hardware noises can also be attributed in this term. The noise term is random, actual RTT value is a constant and hence measured RTT value is random.

The Figure 5.4 shows a sample histogram of RTT data measured. As it can be seen, it resembles a Gaussian distribution and hence we assume noise ***n*** in Equation 5.2 follows a Gaussian distribution *N(0,σ^2)* and therefore ***t*** follows a Gaussian distribution.

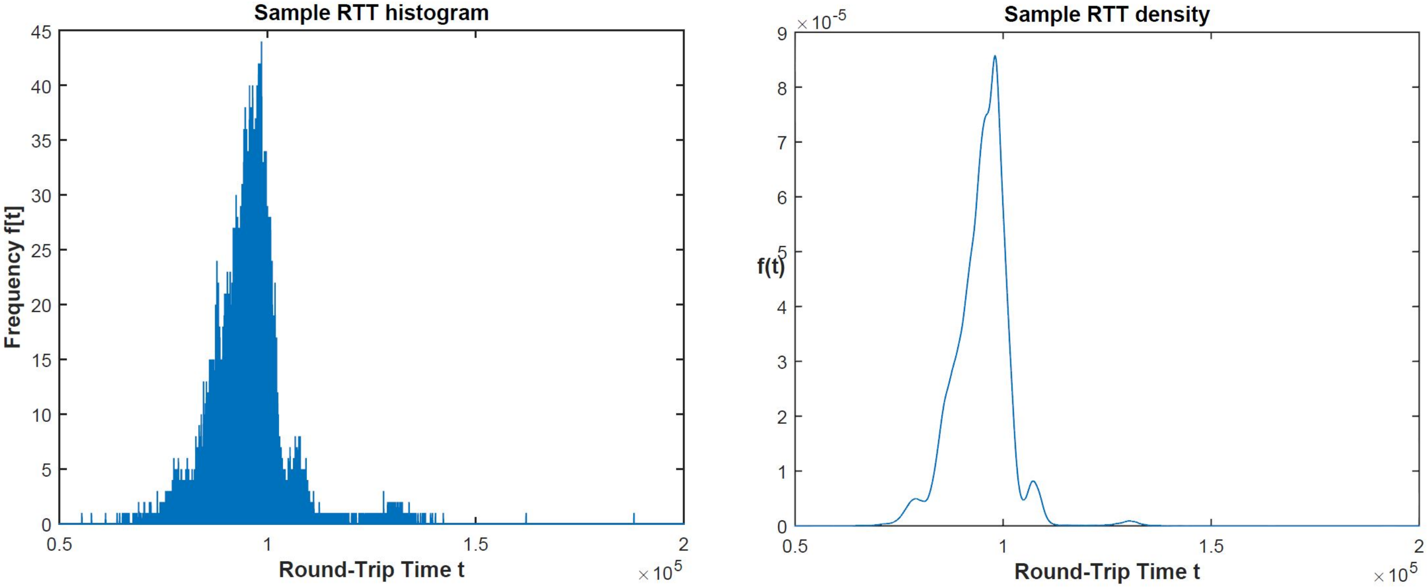


Figure 5.4: Sample RTT (ns) (a) Histogram and (b) Probability Density Function

Denoting ***T*** as the random variable defining τ and ***N*** defining ***n***, the signal to noise ratio is defined as:

$$SNR = \frac{E[T^2]}{E[N^2]} \tag{5.3}$$

where ***E[ ]*** is Expectation.

The variance of a random variable ***T*** can be expressed as:

$$var(T) = E[T^2] - \{E[T]\}^2 \tag{5.4}$$

Ideally, we expect the actual RTT to be a constant and hence its variance is 0. The Noise on the other hand is a Gaussian distribution with zero mean, so therefore ***E[N^2] = var(N)***. Thus, the *SNR* is given by:

$$SNR = \frac{\mu^2}{\sigma^2} \tag{5.5}$$

which can also be expressed in *dB* as:

$$SNR = 20log\left(\frac{\mu}{\sigma}\right) \tag{5.6}$$

The ratio (*μ/σ*) is called *'reciprocal of coefficient of variation'*. The mean of the sample RTT histogram in the figure above is approximately *99.38 μs* and the standard deviation is *6.65 μs*. The *SNR* is given as:

$$SNR = 20log\left(\frac{99.38}{6.65}\right) = 23.49dB \tag{5.7}$$

For a better performance, noise must be reduced, which implies that we need to lower variance. To obtain an accuracy of say *50 ns* or *0.05 μs*, the *SNR* should be increased to:

$$SNR = 20log\left(\frac{99.38}{0.05}\right) = 65.97dB \tag{5.8}$$

One of the most common ways of increasing the *SNR* is to increase the sample size (so that variance is reduced), i.e. more RTT measurements are made. If we collect ***m*** samples, then:

$$t_i = \tau + n_i \quad \text{; where} \quad i = 1, 2, 3...m \tag{5.9}$$

Each sample is independently collected and hence the samples $\{$***t$_i$***$\}$ are independent of each other. If ***X*** represents a random variable on the set of outcomes $\{$***t$_i$***$\}$, ***X*** follows a Gaussian distribution with a non-zero mean, since the noise $\{\boldsymbol{n_i}\}$ is a Gaussian distribution.

$$X \sim N(m\tau, m\sigma^2) \tag{5.10}$$

where $m\tau$ is the mean and $m\sigma^2$ is the variance.

Thus, the SNR of $\boldsymbol{X}$ is given as:

$$SNR\{X\} = \left(\frac{m\tau}{\sqrt{m}\sigma}\right)^2 = m\left(\frac{\tau}{\sigma}\right)^2 \tag{5.11}$$

Clearly, from the above equation, the *SNR* has improved by a factor of $\boldsymbol{m}$ and thus the effect of noise can be reduced since the variance is reduced by a factor of $\boldsymbol{m}$.

$$\frac{SNR\{X\}}{(\tau/\sigma)^2} = m \tag{5.12}$$

Thus, to achieve an accuracy of *0.05 μs*, *SNR* of the measured RTT samples should increase from about *23.49 dB* to *65.97 dB*. Therefore, the minimum sample size required is:

$$10log(m) = 65.97 \quad \Rightarrow m = 3.95 \times 10^6 \approx 2^{22} \tag{5.13}$$

This means we need to collect almost 4 million samples to achieve an accuracy of *0.05 μs*. In our experimentation, we see that it takes almost 2 minutes to collect 2^{16} samples at the driver layer. So, to collect 4 million samples it would take an approximate 120 minutes or 2 hours. This is not practical, considering most reference nodes will be out of range by the time the entire RTT sample collection is completed. Also, for our indoor localization system we need 5-10 reference nodes to determine the target's location with a high accuracy. This means it would take almost 20 hours in the worst case scenario to determine the target's location. Thus, we need an alternate approach to reduce the sample size.

If the RTT data samples are viewed from the distribution perspective, i.e. in terms of probability density functions, we could reduce the number of samples significantly. To explain this intuition, let us consider the case of a simple coin toss in which the probability of either a head or a tail is 0.5. However, if we toss the coin 3 times, the outcome maybe 2 heads and 1 tail. So, by the definition of probability, the probability of a head is 0.75 and that of a tail is 0.25. However we know that the probability of seeing a head when a coin is tossed is 0.5 and that of a tail is also 0.5. But if we perform say 10,000 experiments the probability almost converges to 0.5 and the probability is exactly 0.5 if we perform 1 million experiments. It is this intuition that led us to perform TOA ranging by collecting fewer samples than 4 million as indicated above. The RTT data samples measured will be similarly distributed whether we collect 4 million or 65536 (2^{16}) samples. We chose 65536 samples since it meets the practical considerations. Figure 5.5 shows the RTT sample distribution for 2^{16} (64k) and 2^{15} (32k) samples. We can clearly see that reducing the number of samples still retains the underlying distribution. Thus, we can assume that 4 million samples will also have the same distribution.

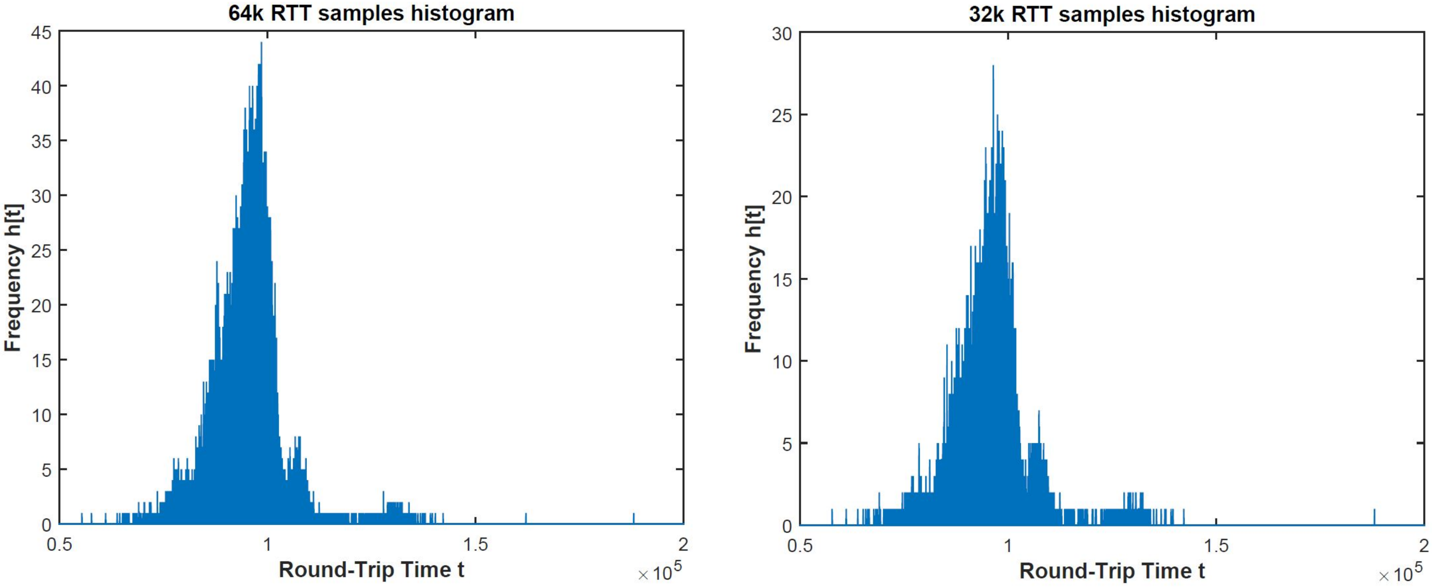


Figure 5.5: RTT (ns) Histogram of (a) 64k samples and (b) 32k samples

When we construct a probability density function (pdf) to the histograms as shown in Figure 5.5 using **MATLAB's** built-in function *ksdensity*, it yields similar density function estimate for 2^{16} (64k) and 2^{15} (32k) samples as shown in Figure 5.6.

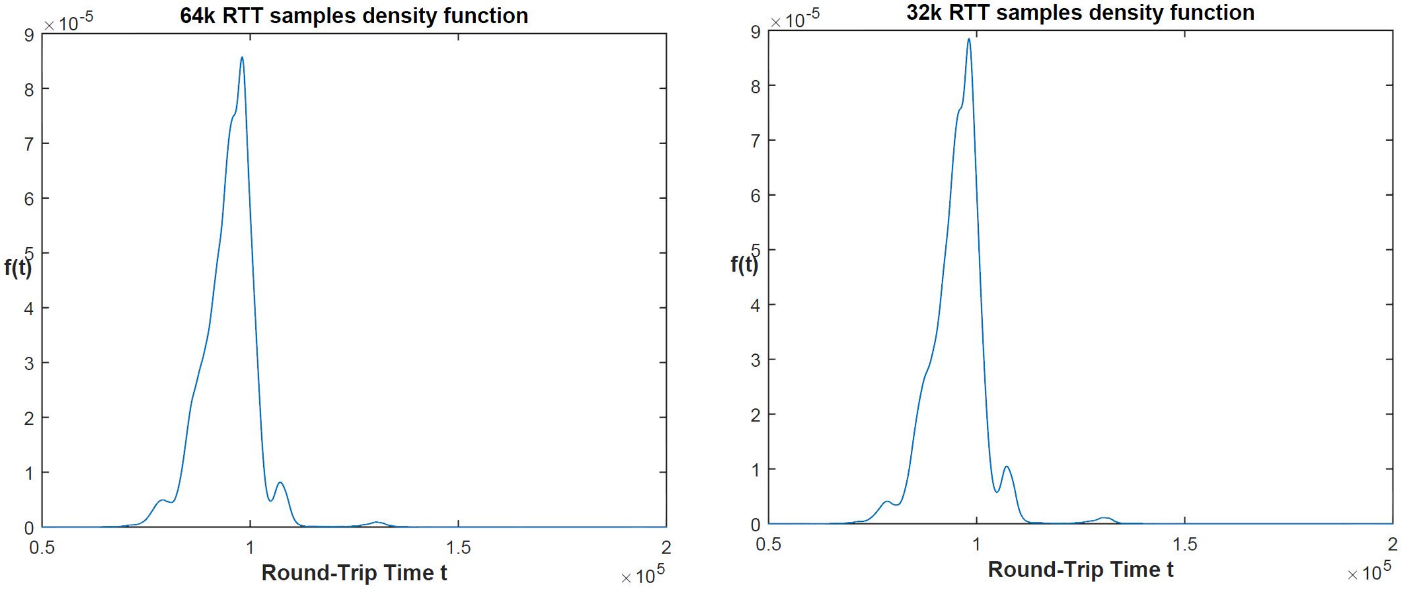


Figure 5.6: RTT (ns) pdf of (a) 64k samples and (b) 32k samples

5.2.2 Indoor Environments

Using the above methodology of round-trip time (RTT) measurement at the kernel driver layer, we collected RTT samples in different indoor environments to analyze and build a reference database to perform Statistical TOA ranging to determine the distance between the target and reference nodes. In this section, we provide an overview of the different indoor environments where we measured the RTT data samples. Figure 5.7 serves as a legend for the maps of those indoor environments.

5.2.2.1 Hallways

Almost all indoor environments have long hallways connecting one part of the building to another. In hotels for instance, room numbers are often confusing and without proper sign boards, people cannot locate on which side of the hallway is their hotel room. Our indoor navigation

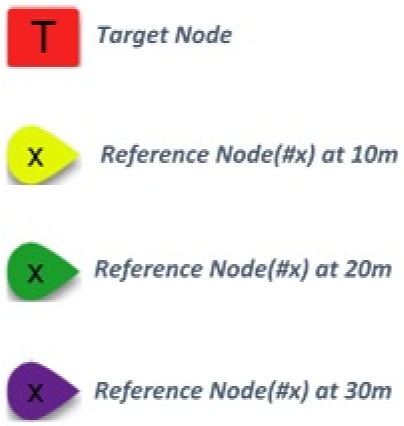

T
Target Node
X
Reference Node(#x) at 10m
X
Reference Node(#x) at 20m
X
Reference Node(#x) at 30m

Figure 5.7: Legend for Maps in this section

system can aid users to find their way to a room or any other place of interest. In such hallways, there are usually no objects along the way and the RF signals travel in a direct path from the target to the reference node and also bounce off (reflections) the walls on either side. We conducted several experiments along the hallway on the second floor of Shelby Center (Right Wing) near our research lab 2323 at Auburn University (Fig. 5.8). The hallway is bound by glass walls on one side and wooden walls on the other. We also conducted field tests on the second floor of Broun Hall, Auburn University (Fig. 5.11). This building is comparitevely older and the walls are made of concrete. The map below shows the target and reference nodes which are at 10, 20, 30 m distances and are placed in locations where the target is in sight and out of sight (line-of-sight-LOS and non-LOS) to each reference node.

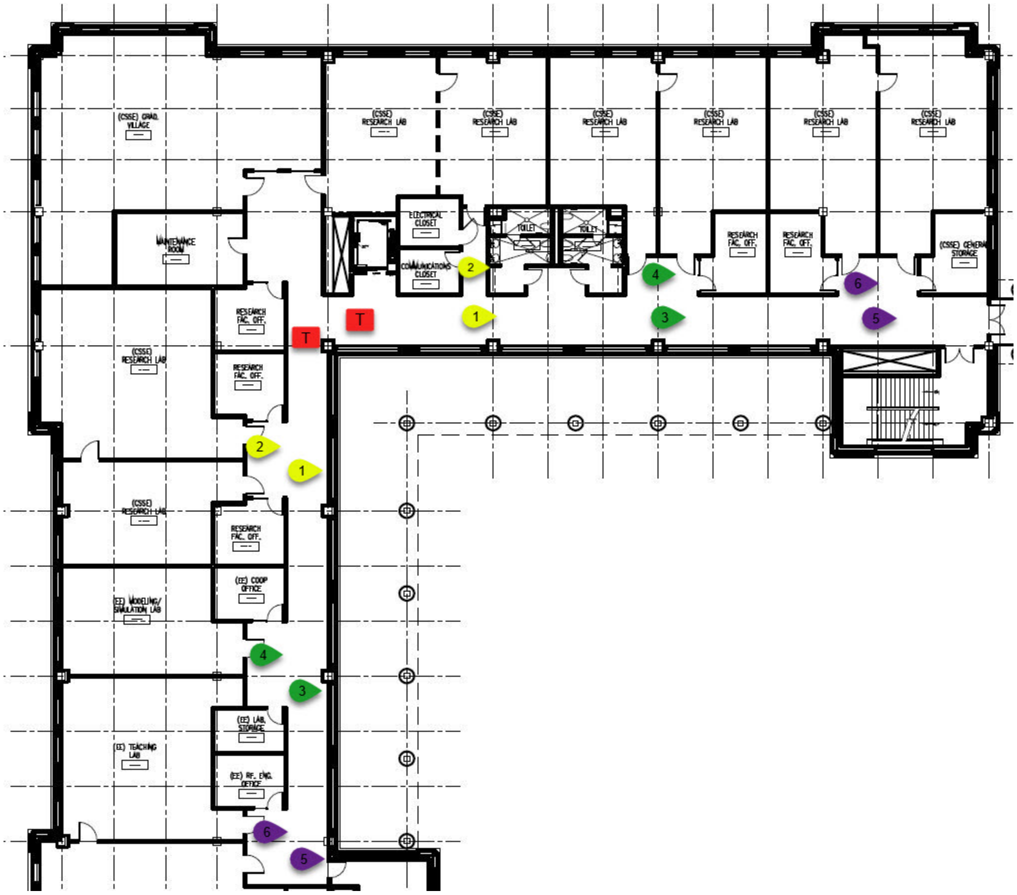


Figure 5.8: Shelby Floor Map with target and reference nodes positions

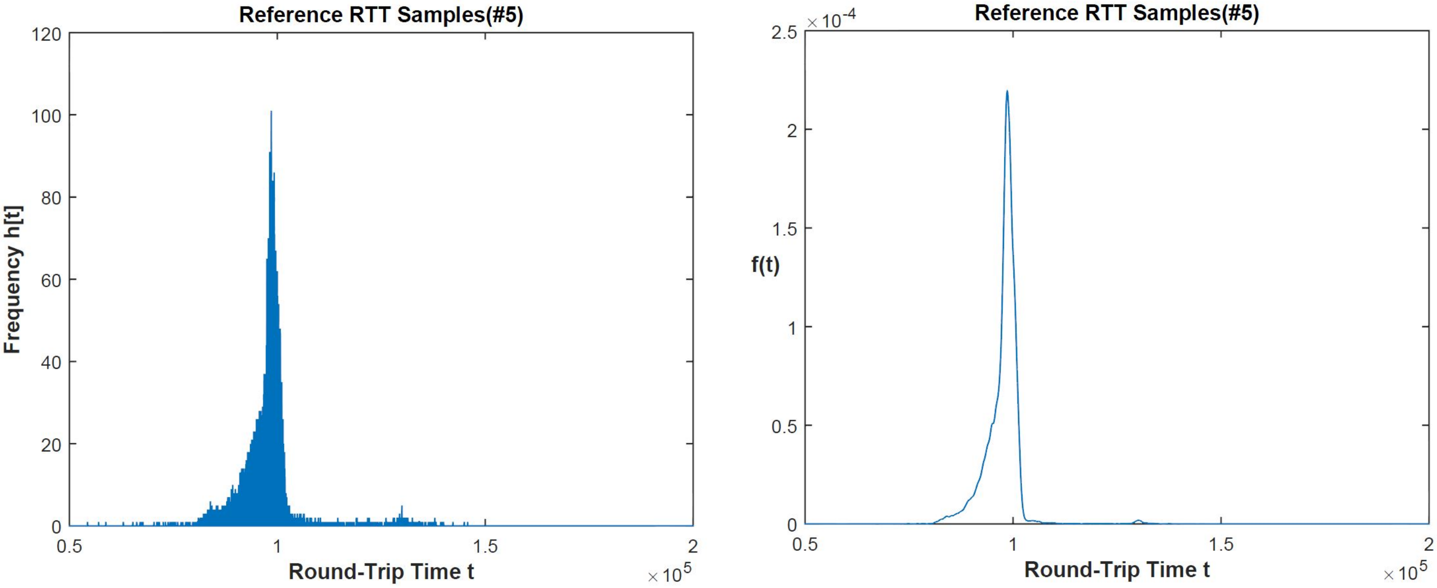


Figure 5.9: RTT (ns) data for node #5 in Fig 5.8 (a) Histogram (b) pdf

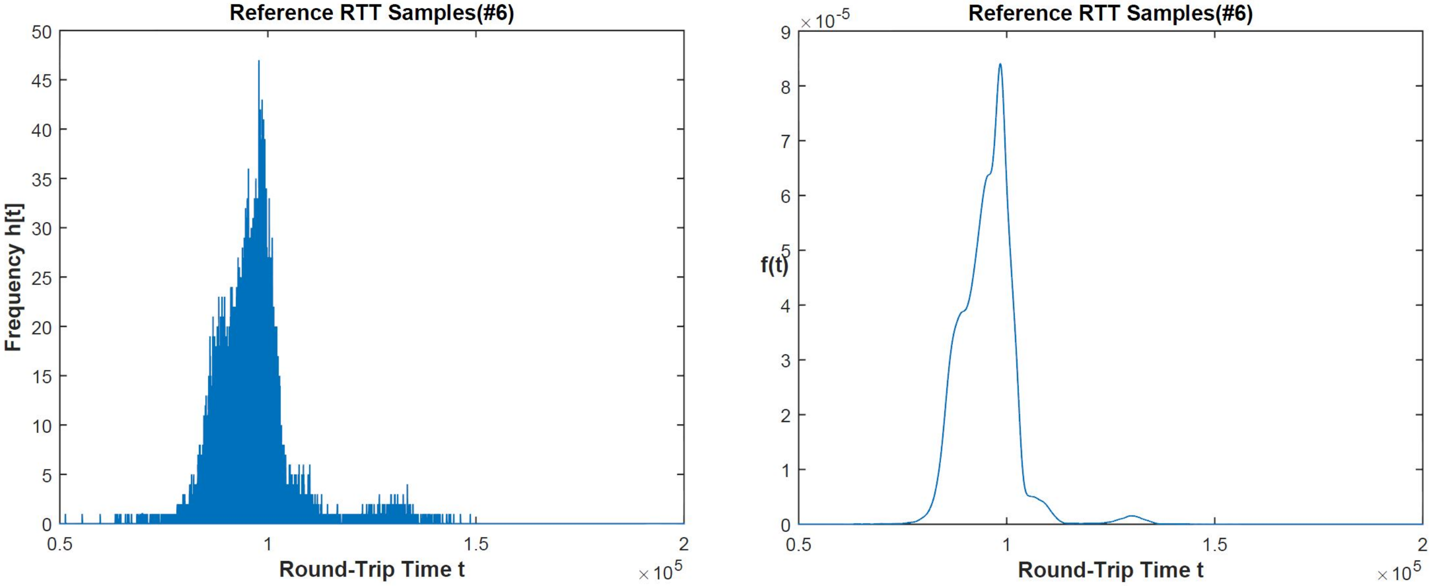


Figure 5.10: RTT (ns) data for node #6 in Fig 5.8 (a) Histogram (b) pdf

Let us consider the RTT samples collected when the target communicates with reference node 5 and 6 (Fig. 5.8) at the end of the Shelby hallway which are 30 m apart. Their histograms and pdf (probability density function) are as shown in the Figures 5.9 and 5.10. Since the target and reference node 5 are in a direct line of sight with each other, we can see in figure 5.9 there a prominent narrow peak, but on the other hand Figure 5.10 comparatively has a slightly broader peak. This is due to the non-LOS condition between the target and reference node 6 and some RF signals arrive in paths which are in close vicinity of the Direct path, causing the peak to be slightly broad.

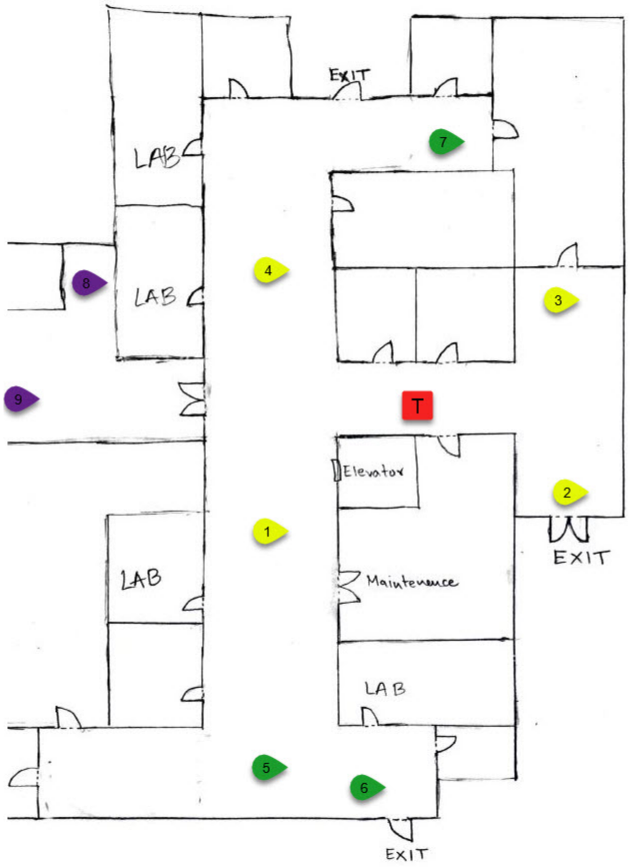


Figure 5.11: Broun Hall Floor Map with target and reference nodes positions

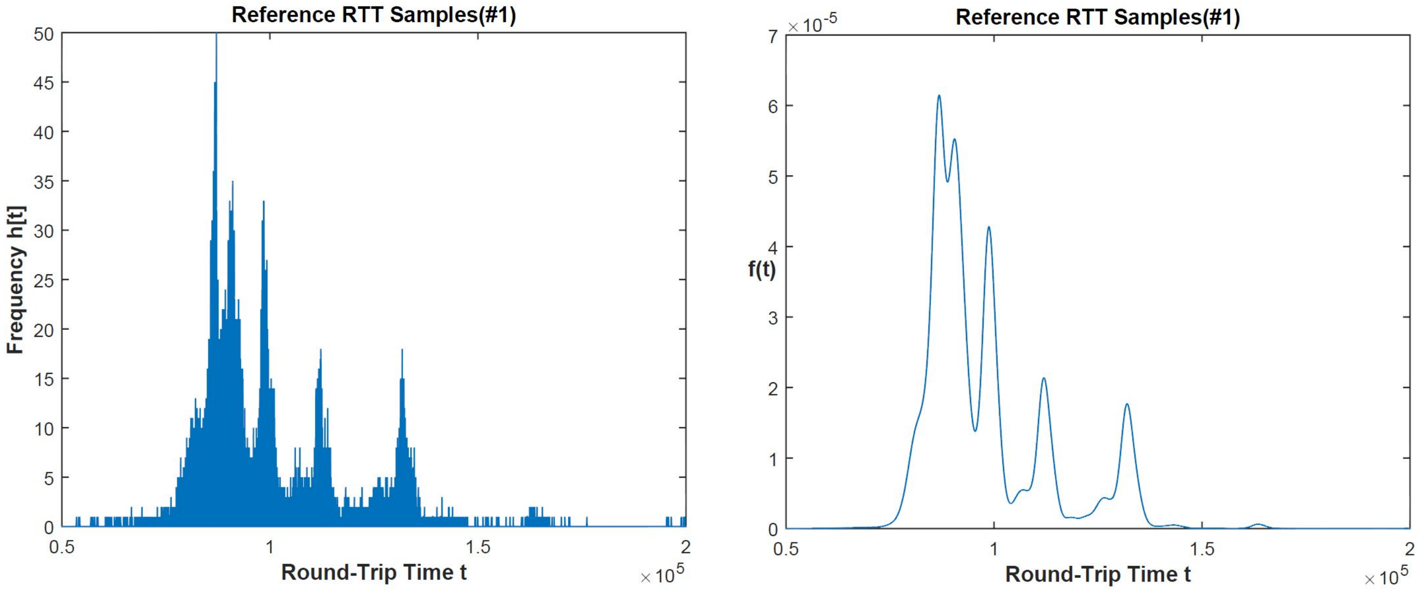


Figure 5.12: RTT (ns) data for node #1 in Fig 5.11 (a) Histogram (b) pdf

Figure 5.12 represents the histogram of the RTT data collected between target and reference node 1 in Broun Hall which are 10 m apart. This RTT sample distribution has multiple peaks. With our repeated experimentation, we have observed that the peaks in the RTT sample distributions occur at quantum/discrete locations. These peaks occur at approximately 86 μs, 90 μs, 97 μs, 111 μs, 113 μs and we account this quantum shifts to the differing processing times during packet retransmissions due to queuing and thermal noise. This is explained in the next chapter. In Figure 5.9, the strongest peak is at 97 μs and this also occurs in Figure 5.12 except that it is not the most dominant peak anymore.

5.2.2.2 Open Space

An open space in a building is usually like a reception area or a front lobby. In such areas, we expect the RF signals to travel in a direct path from the target to reference and multipath conditions exist when RF signals undergo diffraction when the signals deflect off objects like furniture, desks and others. Reflections off the walls are less significant in such environments. We conducted several experiments in the Front lobby of the Shelby Center (Fig. 5.13), Foy Hall (Fig. 5.14) of Auburn University, Auburn, AL. The walls in Shelby are made of wood, while that in Foy Hall is of concrete. The map below shows how the target and reference nodes are placed while performing the experiments. Once again, the reference and target nodes are placed 10, 20, 30 m apart.

Let us consider the RTT samples collected when the target communicates with reference node 1 both placed in Shelby Center front lobby (Fig. 5.13) and are 10 m apart. Its histogram is shown in Figure 5.15. It can be seen that the peak is at approximately 82-83 μs instead of the 97 μs we observed in the hallway. This is because, the Front Lobby is more of an open space and hence the RF signals undergo less reflections on their way to and from the target node. Similar result is seen in Foy Hall (Fig. 5.14) reference node position 2 which is also an open space and this is shown in the Figure 5.16.

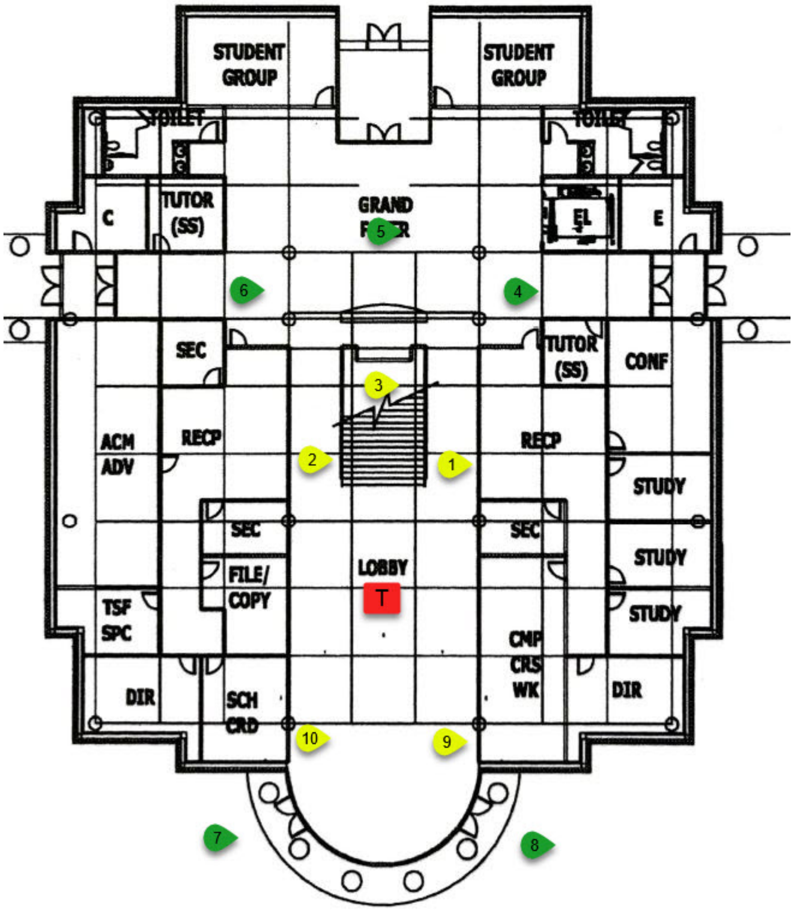


Figure 5.13: Shelby Front Lobby Floor Map with target and reference nodes positions

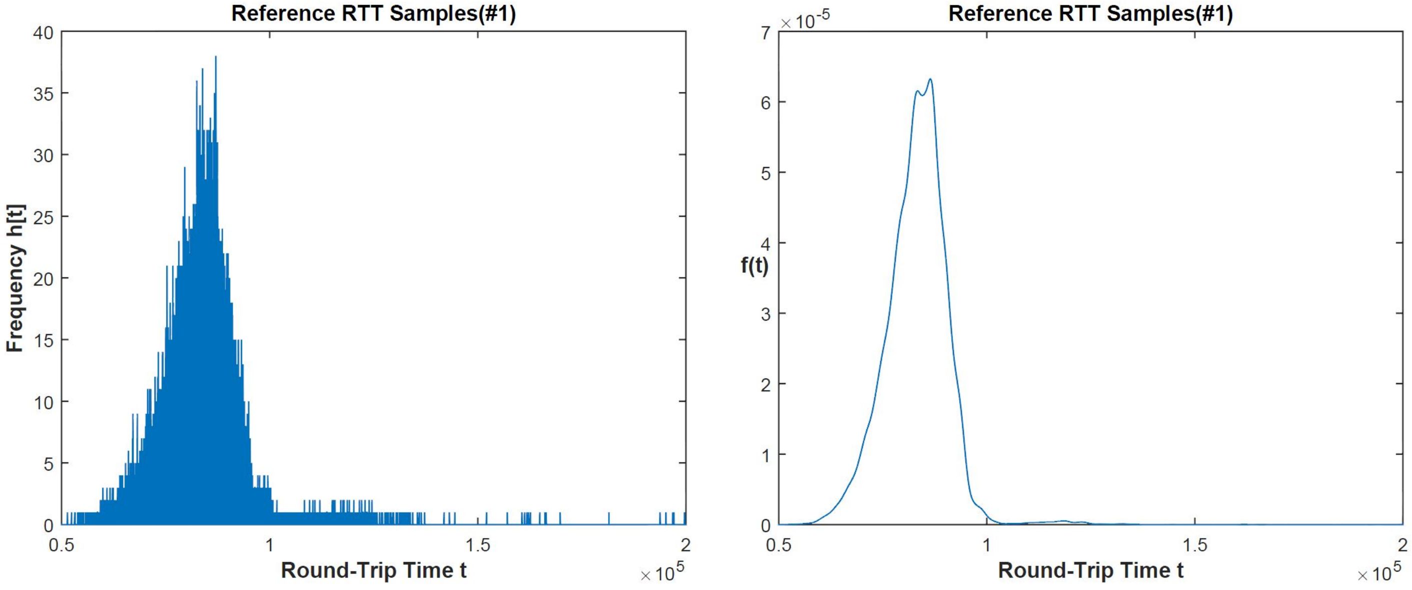


Figure 5.15: RTT (ns) data for node #1 in Fig 5.13 (a) Histogram (b) pdf

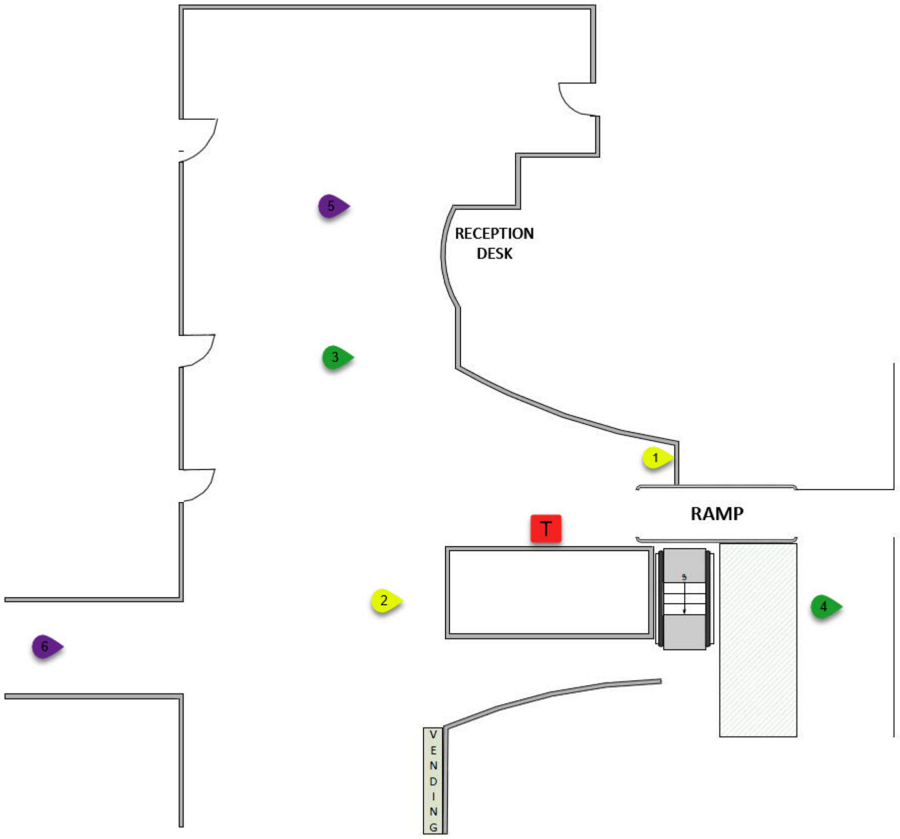


Figure 5.14: Foy Hall Floor Map with target and reference nodes positions

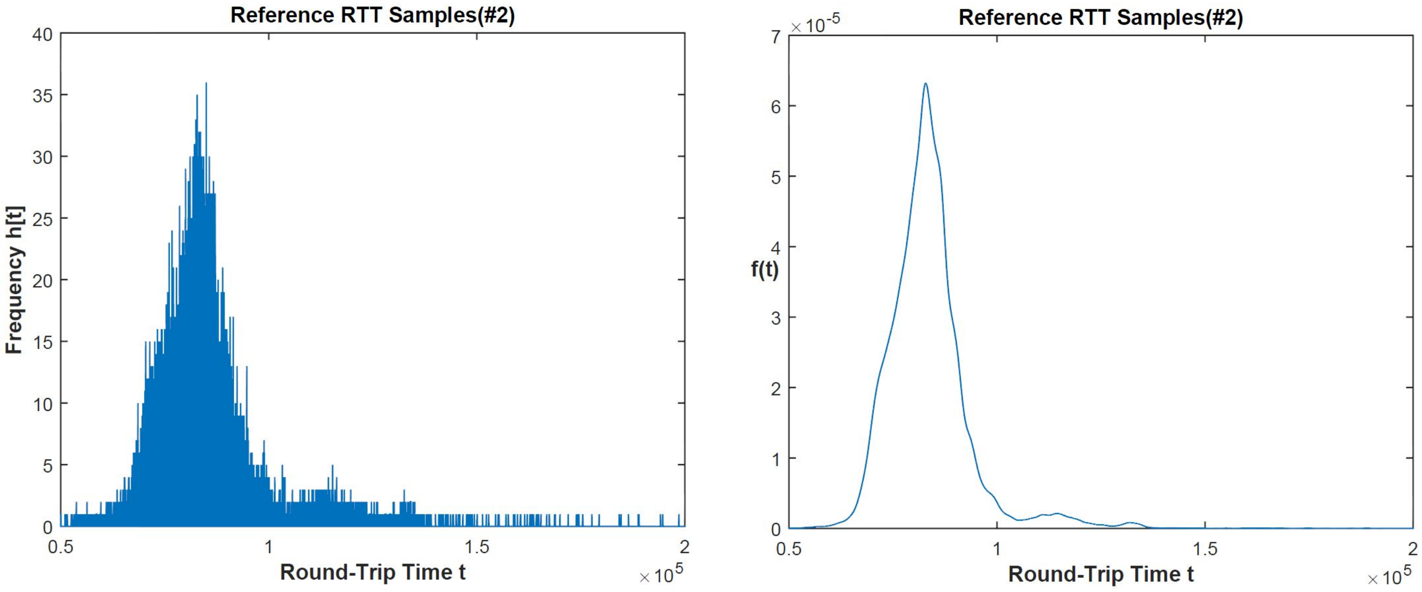


Figure 5.16: RTT (ns) data for node #2 in Fig 5.14 (a) Histogram (b) pdf

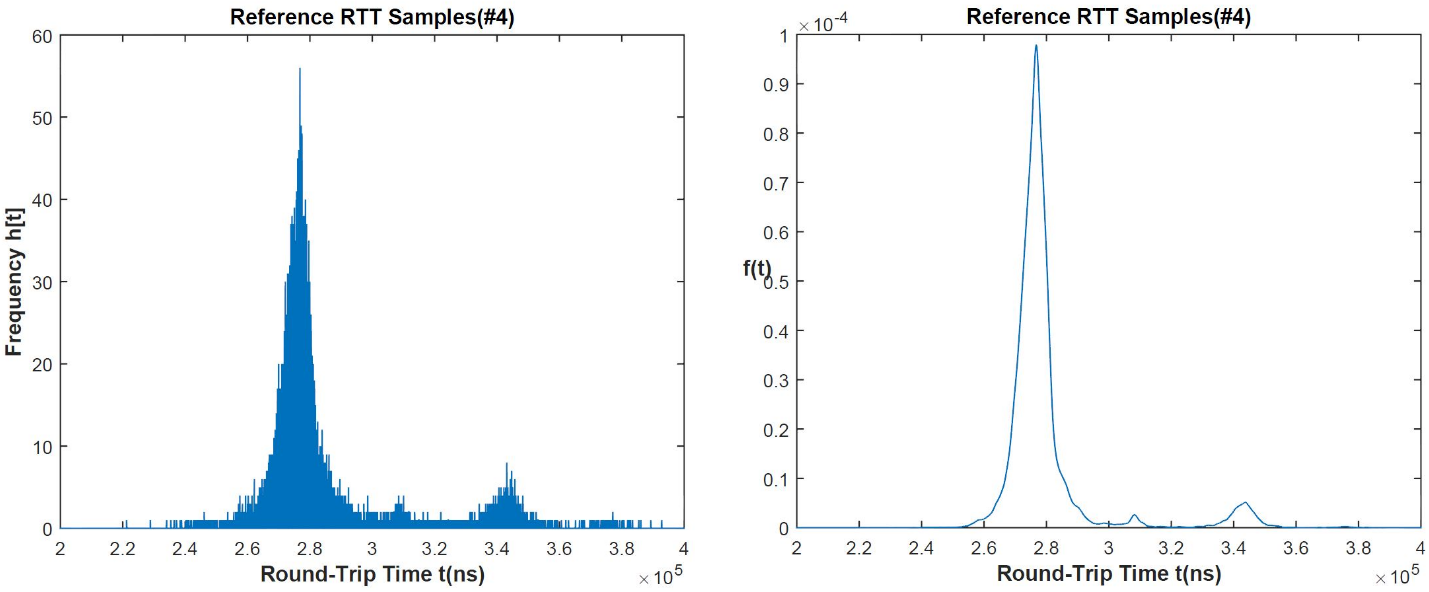


Figure 5.17: RTT (ns) data for node #4 in Fig 5.13 (a) Histogram (b) pdf

While conducting the experiments in these two buildings, we came across another type of RTT sample distribution whose peaks are located at approximately 276 μs. This is depicted in Figure 5.17. We observe that such results are seen when there is a complete NLOS (non-line-of-sight) condition between the reference and target nodes and they are either 20 or 30 m apart. In Figure 5.13 this result is seen at reference node positions 4 and 6 and they are in complete NLOS with the target. This result is also seen in Figure 5.14 at reference node positions 4, 6 which are 20 and 30 m from the target respectively. This peak at 276 μs is another one of the quantum/discrete shifts we observe in our RTT measurements. We amount this to the increased processing times due to several packet losses and errors which result in several retransmissions and this can be expected when the reference nodes are in complete NLOS with each other. The devices usually queue packets to be transmitted and any event of retransmission can cause the queue to be flushed, and hence the increased processing times.

5.2.2.3 Multiroom

In indoor environments, there can be a scenario where the target is in one room and the reference node is in another room and these rooms are separated by either wooden or concrete walls. The RF signals have to penetrate through the walls or travel in a longer direct path from target to the room's door, along the hallway, and then through the other room's door to the reference node. In such scenarios we expect the RTT values to be much higher and several packet losses resulting in retransmissions which could add to the processing times. We conducted several such multiroom experiments by placing the target in our Research Lab 2323 and the reference nodes in adjacent Server room, Graduate Students Desk room and also along the hallway of Shelby Center, Auburn University.This is shown in Figure 5.18.

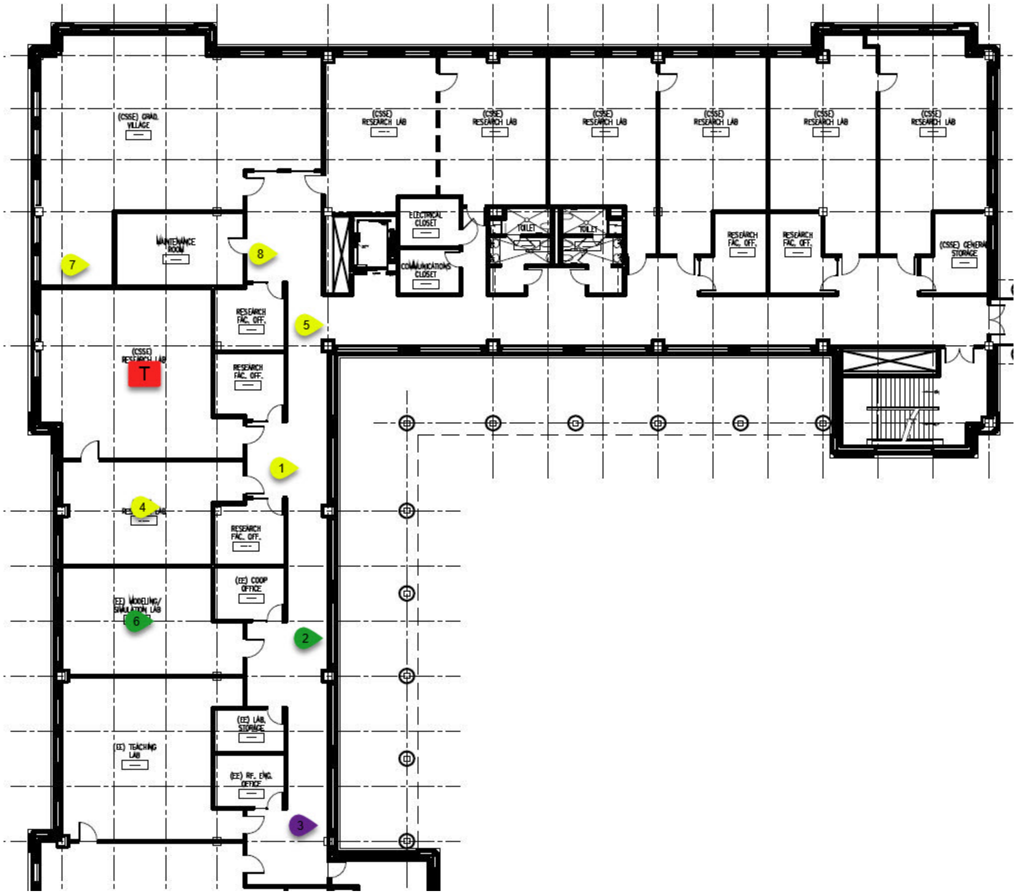


Figure 5.18: Shelby Right Wing Building 2nd floor with target and reference nodes positions

Finally, let us consider the RTT samples collected when the target communicates with reference node 4, with the target placed in our Research Lab 2323 and reference node is in the adjacent Server Room. They are 10 m apart. The Server room has metal rack that houses the servers and

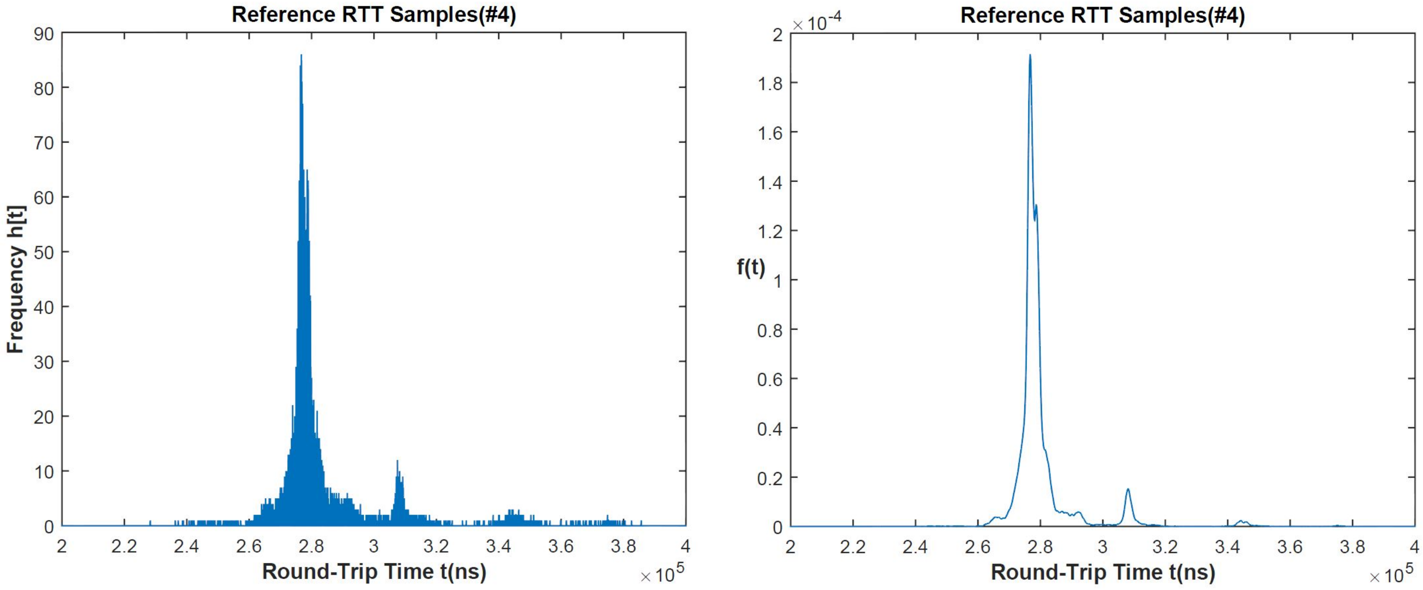


Figure 5.19: RTT (ns) data for node #4 in fig.5.18 (a) Histogram (b) pdf

has a relatively cooler temperature inside. Figure 5.19 shows the histogram of the RTT data samples measured. Once again, we have the peak located at approximately 276 μs and this is because the two nodes are in complete NLOS with each other and RF signals penetrate the walls. This causes several packet losses resulting in increased processing times adding to the overall RTT. In addition, the server room has a metal rack and since RF signals cannot penetrate through metals, packet losses and errors are expected.

Figure 5.20 shows the histogram and pdf of the RTT samples collected between the target and reference node 3 which are 30 m apart. Here we can see the peak is at roughly 343 μs. This is another one of the quantum/discrete shifts we observe in our RTT measurements due to varying processing times causing packet losses resulting in retransmissions.

To summarize, we have collected a large number of RTT data sets in different indoor environments and in different buildings. We observe that the RTT distribution has several dominant modes (peaks) occurring at quantum/discrete locations like 82, 87, 89, 97, 111, 276 and 343 μs irrespective of the distance between the target and reference. Our reference database is built for 10, 20, 30 m by taking all this into consideration. We introduce in the next few chapters, a statistical

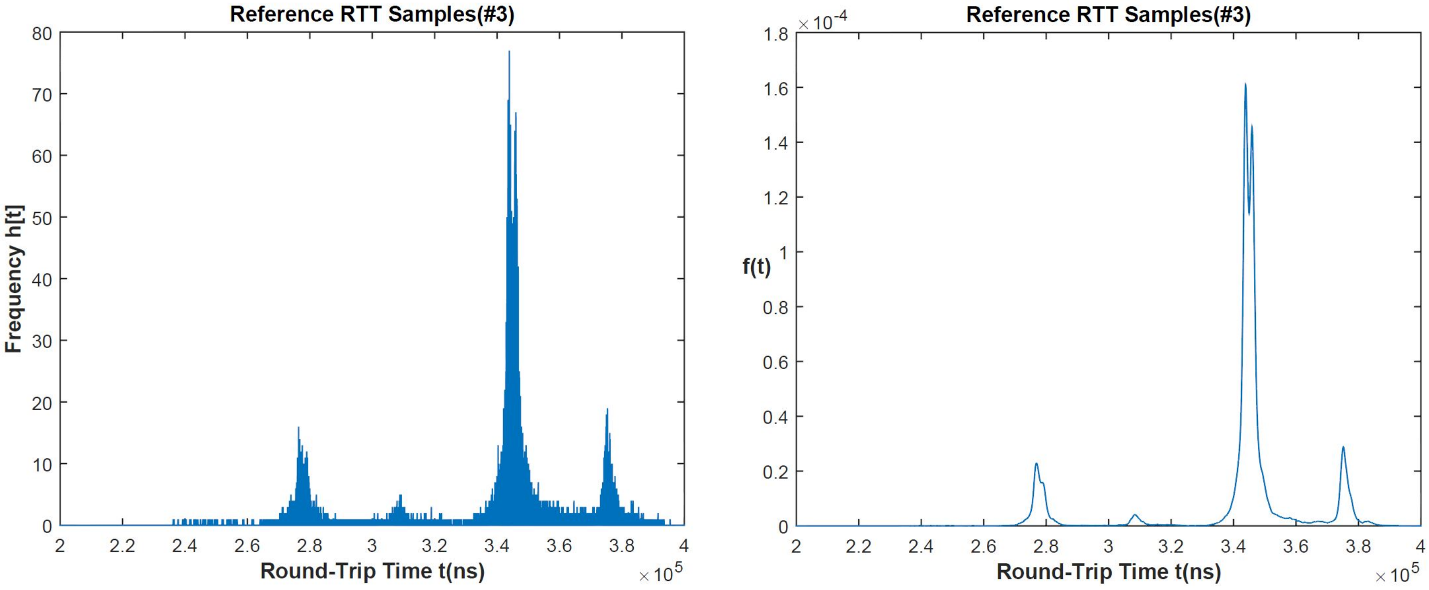


Figure 5.20: RTT (ns) data for node #3 in fig.5.18 (a) Histogram (b) pdf

distance measure called Bhattacharyya coefficient which we use to distinguish between the RTT distributions for different distances and build the database based on it.

5.3 Relating RTT Values to IEEE 802.11g standard

In this section, we analyze and relate the round-trip times measured in the previous section with the IEEE 802.11 standard. The IEEE 802.11 standard defines the MAC and PHY layer of the TCP/IP protocol stack. Different variants of the standard such as IEEE 802.11b/a/g/n have different MAC and PHY layer specifications to provide better data rates, improve latency and reduce errors. Since we operate the Wireless NIC (network ineterface card) in IEEE 802.11g mode, we refer to the IEEE 802.11g standard [21].

IEEE 802.11g standard specifies a contention free and a contention based MAC protocol which are termed as Point Coordinate Function (PCF) and Distributed Coordination Function (DCF) respectively. In PCF, the Access Point (AP) coordinates access for each station (STA), while DCF includes CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) mechanism to provide access to each STA [9]. Since the target and reference nodes in our indoor localization system are in an ad-hoc network, there is no centralized authority (AP) and hence each node contends for medium access through CSMA/CA.

In CSMA/CA with QoS control,if the medium is free,the node waits a certain time called Arbitrary Interframe Space (AIFS) before transmitting a data frame. Similarly, the node waits a Short Interframe Space (SIFS) before transmitting an acknowledgement. In the ATH9k driver, the MAC layer implements the IFS periods along with contention window and backoff periods [17].

As mentioned before, we measure the RTT at the driver layer just before the packet is delivered to the hardware layer for transmission over the wireless interface. Thus, our Round-Trip Times measured does not include the IFS periods but since the hardware layer, which is denoted as PHY layer in IEEE 802.11 standard, performs additional processing before the actual transmission, this processing time is part of the Round-Trip Time (RTT).

5.3.1 PHY Layer

MAC layer adds a MAC header to the DATA frame containing the source and destination address (MAC address), QoS control information and other flags to form a MAC Protocol Data Unit (MPDU). This MPDU is passed to the PHY layer which then includes its own headers. PHY layer consists of PLCP and PMD sublayers. Physical Medium Dependent (PMD) sublayer provides transmission and reception of PHY layer protocol data units between two stations via the wireless medium. To provide this service, the PMD interfaces directly with the wireless medium (i.e, RF in the air) and provides modulation and demodulation of the frame transmissions. Physical Layer Convergence Protocol (PLCP) sublayer prepares the MPDU into a frame format suitable for transmission. It also delivers incoming frames to the MAC layer. It appends a PLCP preamble and header fields (to the MPDU) that contain information needed by the Physical layer transmitters and receivers to form a PPDU (PLCP PDU) [12]. IEEE 802.11 does not specify any explicit synchronization procedure between the transmitters and receivers and PLCP Preamble enables the receiver to synchronize to the incoming signal properly before the actual content of the frame arrives. PLCP header provides information about the frame such as the modulation format, time needed to transmit MPDU etc. Thus, the time difference between sending a DATA frame and receiving an ACK is computed as:

$$\begin{aligned} t_{ACK} - t_{DATA} = {} & [Preamble + Header + DATA + Propagation] \\ & + [SIFS + Preamble + Header + ACK + Propagation] \end{aligned} \tag{5.14}$$

where DATA and ACK terms represent the time taken to transmit the data frame and acknowledgement MPDU respectively and length of SIFS in IEEE 802.11g networks is 10 μs.

5.3.1.1 ERP and OFDM

The IEEE 802.11g uses DSSS (Direct Sequence Spread Spectrum) and OFDM (Orthogonal Frequency Division Multiplexing) technology to provide data rates upto 54 Mbps (Mega Bits per second) at the 2.4 GHz ISM band. For basic data rates-*1, 2, 5.5, 11 Mbps*, it uses DQPSK/BPSK/CCK modulation with DSSS and QPSK, 16/64 QAM modulation with OFDM for higher data rates-*6, 9, 12, 18, 24, 36, 48, 54 Mbps*. These high data rates are termed as ERP (Extended Rate Physical). Table 5.2 gives the different modulation schemes used with OFDM for ERP data rates. The coding rate is lesser than 1 because extra redundant bits are added to protect against errors. The coding scheme used is Forward Error correction (FEC) [50, 21, 28].

The OFDM structure consists of 52 subcarriers, 48 are for data and 4 are pilot subcarriers with a carrier separation of 0.3125 MHz (20 MHz/64). Each of these subcarriers can be a BPSK, QPSK, 16-QAM or 64-QAM modulated. The total bandwidth is 20 MHz with an occupied bandwidth of 16.6 MHz. Symbol duration is 4 μs, which includes a guard interval of 0.8 μs [50]. Figure 5.21 shows the ERP-OFDM structure [46]. The *PLCP preamble* has a duration of 16 μs while *Signal* is 4 μs and every frame is followed by a period of no transmission called *Signal extension* (not shown in Fig. 5.21) which is of length 6 μs. This is to ensure that the data bits can be decoded.

Data Rate	Modulation Scheme	Coding Rate	Coded bits per Carrier	Coded bits per OFDM symbol	Data bits per OFDM symbol
6Mbps	BPSK	1/2	1	48	24
9Mbps	BPSK	3/4	1	48	36
12Mbps	QPSK	1/2	2	96	48
18Mbps	QPSK	3/4	2	96	72
24Mbps	16 QAM	1/2	4	192	96
36Mbps	16 QAM	3/4	4	192	144
48Mbps	64 QAM	2/3	6	288	192
54Mbps	64 QAM	3/4	6	288	216

Table 5.2: Data Rates Specifications-IEEE 802.11g

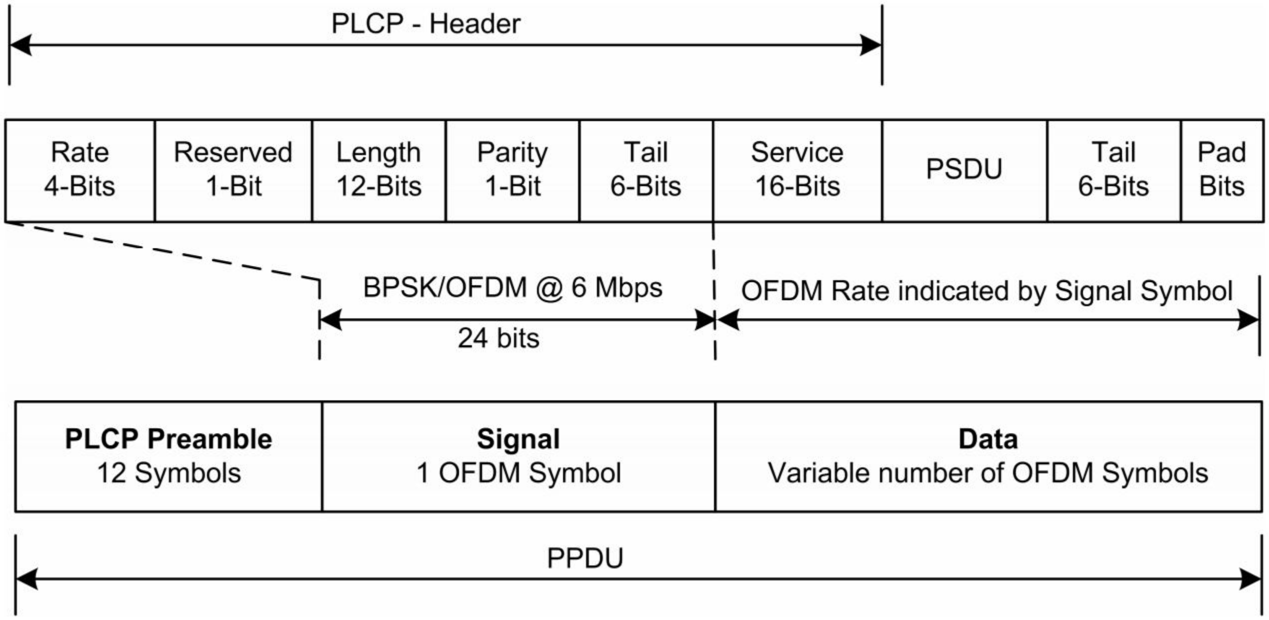


Figure 5.21: ERP-OFDM PPDU Structure

The target and reference nodes in our indoor localization system are set to operate at 24 Mbps. Using Wireshark packet capture tool, we determine the DATA frame MPDU size to be 76 bytes and ACK MPDU is 12 bytes in length. Referring to Table 5.2, for a data rate of 24 Mbps, there are 96 useful data bits in one OFDM symbol. Therefore, the number of OFDM symbols for DATA and ACK frames are:

$$Dataframe = \left\lceil \frac{76 \times 8}{96} \right\rceil = 7 \qquad Ackframe = \left\lceil \frac{12 \times 8}{96} \right\rceil = 1 \tag{5.15}$$

Since each OFDM symbol time is 4 μs, the terms DATA and ACK in Equation 5.14 is 28 μs and 4 μs respectively. DATA is followed by a *Signal extension* of 6 μs. The propagation time is of the order of <1 μs and is negligible. Thus from equation 5.14,

$$\begin{aligned} t_{ACK} - t_{DATA} &= [16 + 4 + (28 + 6) + Propagation] \\ &\quad + [10 + 16 + 4 + 4 + Propagation] \approx 88\mu s \end{aligned} \tag{5.16}$$

We observed in Section: 5.2 that the distribution of RTT values collected during our experimentation have discrete/quantum shifts in peaks (dominant modes) at approximately 90 μs, 97 μs, 111 μs, 113 μs. This increase in RTT values compared to 88 μs in Equation 5.16 is due to the processing time in the hardware layer. We also observe RTT values with peaks at 276 μs, 343 μs which are probably caused by retransmissions which may trigger flushing of queues among other things resulting in increased processing times.

Chapter 6

STATISTICAL TOA RANGING

In this chapter, we provide a more accurate model of the RTT data measured in the previous chapter and introduce our Statistical TOA ranging method to determine the distance between target and reference nodes. We introduce a statistical distance measure called Bhattacharyya distance and compare it with well-known Euclidean distance measure.

TOA ranging requires precise timing synchronization between the target and reference nodes. However, we can eliminate this strict requirement by measuring the round-trip time instead of time of arrival since time is now measured at either the transmitter or receiver. We measure the round-Trip Time between the transmission of a unicast data and reception of the corresponding ACK frame.

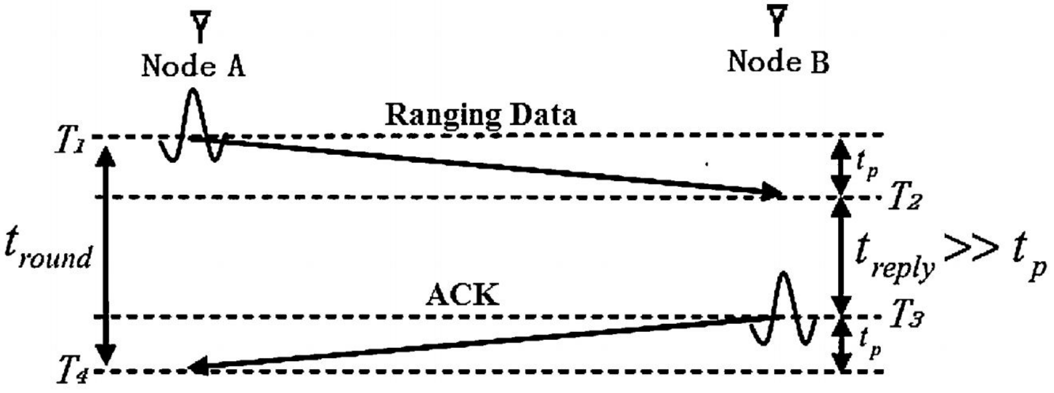


Figure 6.1: Round-Trip Time overview

In [18], the authors introduce this measurement of round-trip time (RTT) and corresponding range estimation. Figure 6.1 is a figurative description of RTT measurement and distance between the target and the reference node is estimated as:

$$\hat{d} = t_p \times c = \frac{t_{round} - t_{reply}}{2} \times c = \frac{(T_4 - T_1) - (T_3 - T_2)}{2} \times c \tag{6.1}$$

where $\hat{d}$ is estimated distance; t_p is propagation time; c is speed of electromagnetic waves (3×10^8 m/s); T_1 is transmitted data pulse timestamp; T_2 is received data pulse at receiver; T_3 is transmitted ACK pulse timestamp; T_4 is received ACK pulse at transmitter timestamp.

However, this range estimation comes with a caveat that the transmitted pulse (RF signals) traverses in one path- the LOS (line-of-sight) path, which is the shortest path between the transmitter and receiver. But in an indoor setting, these RF signals undergo multipath conditions due to the diffraction and reflection off the walls and other objects such as tables, chairs, furniture. This is a common problem in a wireless system, but is more pronounced in an indoor environment. Thus, the round trip time is usually more than expected line of sight RTT and hence the above model of range estimation (Eqn. 6.1) is incorrect.

In the previous chapter, we introduced a model for the measured Round Trip Time based on [64]. Our argument to reduce the sample size to *65536* samples from 4 million samples was that RTT data will have roughly the same distributions, so there is no point collecting so many samples. In our research, we measure the RTT at the driver layer, which means there is additional processing before the DATA and ACK frames can be transmitted. This processing time deals with packet encapsulation at the MAC and PHY layer as defined in IEEE 802.11 standard. So here we provide a more accurate representation of the RTT data measured. The RTT measured can be modeled as:

$$t_{RTT} = \tau + t_{proc_{DATA}} + t_{proc_{ACK}} + n \tag{6.2}$$

where t_{RTT} is the RTT measured at the driver layer; τ is actual RTT; t_{proc} is the processing time before transmission over air interface; n is noise due to multipath, electronic noise.

The physical (PHY) layer translates the binary 0's and 1's of the DATA and ACK frame into electromagnetic signals which are transmitted over the air. Actual RTT in the above equation is the sum of the time of flight of Data frame over the air, and the time of flight of the ACK frame over the air. The processing time on the other hand refers to the processing that the end terminals do (both reference and target nodes) before transmitting signals over the air. Since we measure the RTT at the driver layer, the frame is passed from the MAC layer to the PHY layer which then adds several headers and trailers specific to synchronization of the transmitter and receiver and modulation technologies such as BPSK, QPSK and OFDM used. This was discussed in the previous chapter. As mentioned earlier, multipath effects cause the RTT time to deviate from its expected LOS (line of sight) value. In addition, there is always the effect of electronic hardware noise, operating system overhead.

As seen from Equation 6.2, RTT measurements are very noisy and hence cannot be used to directly compute the range of a target node from the reference node. The Figure 6.2 shows the histogram of *65536* RTT samples measured at the driver layer. There are about 6-7 peaks; one more dominant than the other. This clearly verifies our model above in Equation 6.2, with the measured RTT value drifting away from the actual value due to the multipath and processing times.

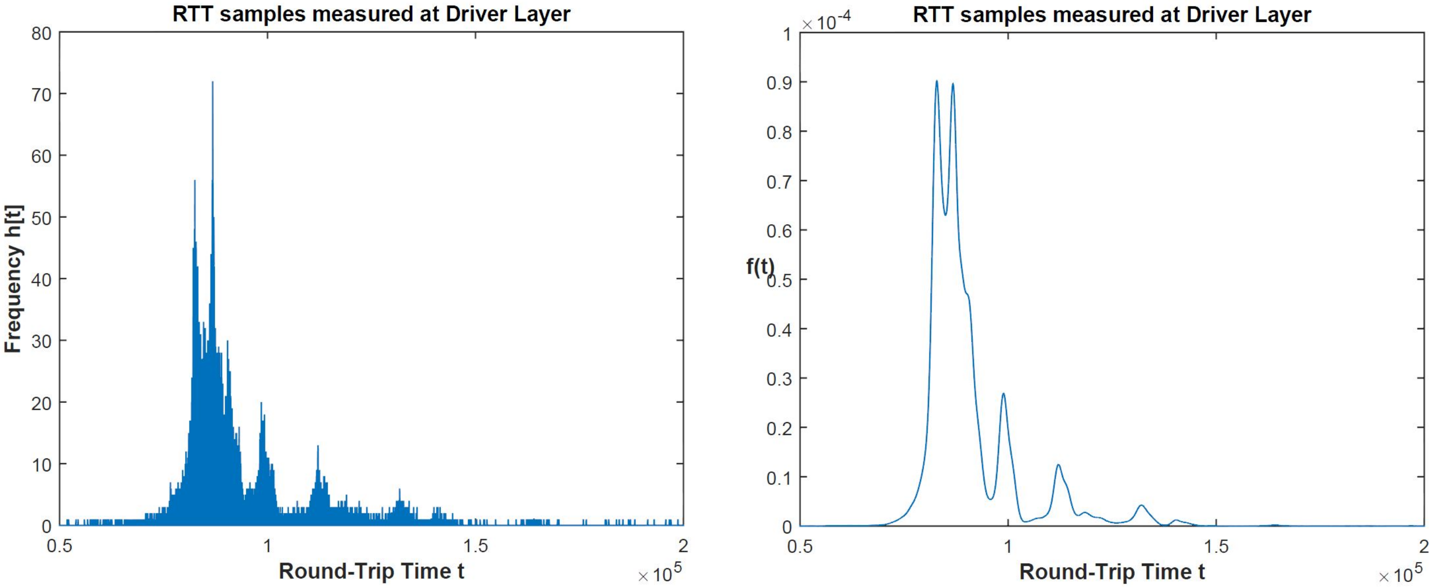


Figure 6.2: An example of RTT samples (measured at driver layer) with multiple peaks (a) Histogram (b) pdf

Thus, we cannot employ Equation 5.2 to estimate the distance between the target and reference nodes and there is a need for a new method to estimate the range by taking into consideration the varying nature of RTT values due to multipath, noise such as the hardware electronic noise, OS overhead such as software queue flushing, and in some cases the processing time due to packet losses.

One method is to distinguish RTT samples for different distances is based on the mean (of data samples). Longer distances should have higher mean values, but due to the varying nature of the of the RTT samples about the mean, this would not be a good solution. But this deviation from the actual value can be useful if it can distinguish two sets of data. Different distances can have different deviations, or in other words, different distances can have different distributions, i.e. probability density functions. We exploit this idea in our proposed method to estimate the range from the measured RTT samples.

Our TOA ranging method has two stages: first, we repeatedly measure the RTT for transmission of DATA packets and the corresponding ACK to be received at the driver layer until 2^{16} samples are collected, and second, we construct probability density functions for this RTT samples to match against the probability density functions of reference databases for different distances to determine range of the target node from the reference node. The matching is performed by using a statistical distance measure called Bhattacharyya Distance and hence we term it, 'Statistical TOA Ranging'. It involves an offline stage where we build the reference database by measuring RTT in different environments and at different distances, and an online stage where the measured real time RTT samples are compared to the stored reference databases using Statistical Distance measures. The first stage was explained in the previous chapter. In the following sections, we discuss this statistical distance measure in detail.

6.1 Statistical Distance

There are several distance measures which find their relevance in different fields such as statistics, physics, chemistry, information theory, mathematics, geology etc. It is used extensively for clustering, pattern classification, clustering, and information retrieval problems. Distance is usually defined as a quantitative degree of how 'far' apart two objects are. Those distance measures satisfying the metric properties are simply called ***metric*** while other non-metric distance measures are occasionally called ***divergence*** [10].

As indicated in [54], the metric properties of a distance function are:

1. Non-negativity: $\boldsymbol{d(x,y) \geq 0}$

2. Identity of Indiscernibles: $\boldsymbol{d(x,y) = 0}$ ***if and only if*** $\boldsymbol{x = y}$

3. Symmetry Property: $\boldsymbol{d(x,y) = d(y,x)}$

4. Triangle Inequality: $\boldsymbol{d(x,z) \leq d(x,y) + d(y,z)}$

For all x, y, z in X such that $\boldsymbol{d{:}X \times X \rightarrow R^+}$, i.e. $\boldsymbol{d}$ is a function on $\boldsymbol{X}$ with a domain of non-negative real numbers.

The real time and reference RTT data can be thought of as two Random variables. The statistical distance quantifies the difference between the two or, put in other words, is a measure of the similarity of their probability distributions. Statistical Distance measures may not form a metric and some of them might not be symmetric. Hence, they are also termed divergences, measures of discriminatory information, or measures of separation.

In literature, there are many statistical similarity measures such as *Kullback-Liebler divergence*, *Hellinger Distance*, *Mahalanobis distance*, *Bhattacharyya coefficient*, *Matusita distance*, *chi-squared distance*. Among this, the *Bhattacharyya distance* and its coefficient is simple to implement and very robust. It can be applied to discrete functions (on histograms) and is symmetrical. In fact, some of the distance measures like Hellinger, Matusita, chi-squared are all closely related to the Bhattacharyya coefficient. On the other hand, the Kullback-Liebler divergence method needs the two distributions to be absolutely continuous.

6.2 Bhattacharyya Coefficient and Distance- Mathematical Formulation

The Bhattacharyya coefficient measures the relative closeness of the probability density functions of the two random variables. In the continuous case, it measures the similarity of density functions, while in the discrete case, it measures the similarity between the probability mass functions.

Let us consider two random variables, A and B. Their probability density functions are given as $\boldsymbol{f_A(x)}$ and $\boldsymbol{f_B(x)}$. Then the Bhattacharyya coefficient is given by:

$$BC(A,B) = \int \sqrt{f_A(x) f_B(x)} dx \tag{6.3}$$

In the discrete case, let ***p(x)*** and ***q(x)*** be the probability mass functions of A and B.

$$BC(p,q) = \sum_x \sqrt{p(x)q(x)} \tag{6.4}$$

The Bhattacharyya distance is then given by,

$$D_B(p,q) = -\ln(BC(p,q)) \tag{6.5}$$

In either case, $\boldsymbol{0 \leq BC(p,q) \leq 1}$ and $\boldsymbol{0 \leq D_B(p,q) \leq \infty}$. It is important to note that the $\boldsymbol{D_B(p,q)}$ does not satisfy the *triangle inequality* condition and hence does not form a metric.

6.3 More Details on the Bhattacharyya Coefficient

The Bhattacharyya coefficient is named after a statistician, Anil Kumar Bhattacharyya from the Indian Statistical Institute, India in the 1930s. He considered two multinomial populations, i.e. two sets of samples having a multinomial distribution. Each multinomial distribution consists of ***'k'*** categories with success probabilities (p_1,p_2,p_3....p_k) and (q_1,q_2,q_3....q_k) respectively such that $\sum p_i = 1$ and $\sum q_i = 1$. Geometrically, these two distributions can be plotted as points in a k-dimensional space referred to a system of orthogonal co-ordinate axes by taking ($\sqrt{p_1}$, $\sqrt{p_2}$, $\sqrt{p_3}$.... $\sqrt{p_k}$) and ($\sqrt{q_1}$, $\sqrt{q_2}$, $\sqrt{q_3}$.... $\sqrt{q_k}$) can be considered as the direction cosines to the two position vectors defined by the points in the k-dimensional space. If θ is the angle between these two vectors, then:

$$\cos\theta = \sum_{i=1}^{k} \sqrt{p_i q_i} \tag{6.6}$$

This forms the background on which the Bhattacharyya coefficient was derived on [3].

6.4 Bhattacharyya Coefficient vs Euclidean Distance

Consider two sets of RTT data values each having ***N*** samples: X = $\{x_1, x_2, x_3 .... x_N\}$ which is the measured real-time RTT and Y = $\{y_1, y_2, y_3 .... y_N\}$ is the stored reference database for a known distance. The *Euclidean distance* suggests that the shortest distance between two points is a line and accordingly it is defined as:

$$ED(X,Y) = \sqrt{\sum_i (x_i - y_i)^2} \tag{6.7}$$

If the mean of *X* is represented as μ_X, then the variance is given as:

$$\sigma_X^2 = E[X^2] - (E[X])^2 = \frac{\sum_i x_i^2}{N} - \mu_X^2 \tag{6.8}$$

This can be simplified as:

$$\sum x_i^2 = N\sigma_X^2 + N\mu_X^2 \tag{6.9}$$

Combining equations 6.7, 6.8, 6.9 yields:

$$ED(X,Y) = \sqrt{N}\sqrt{\sigma_X^2 + \sigma_Y^2 + \mu_X^2 + \mu_Y^2 - 2N\mu_x\mu_Y} \tag{6.10}$$

In [64], the authors have used Euclidean distance to distinguish between two sets of RTT samples. They use the above equation to argue that it considers both mean and variance of the RTT samples and hence can be very attractive statistical distance measure to distinguish the RTT data samples. However, we prove below that the Euclidean distance cannot distinguish between data with roughly similar distributions, which is usually the case in indoor RTT measurements.

On the other hand, the Bhattacharyya coefficient and it's distance can be very good indicators of dissimilarity.

Let us consider an example to prove this point. Let *R* be the RTT data set measured in real time and *X*, *Y* be stored reference datasets. Their histograms are shown respectively in Figure 6.3(a), (b), (c).

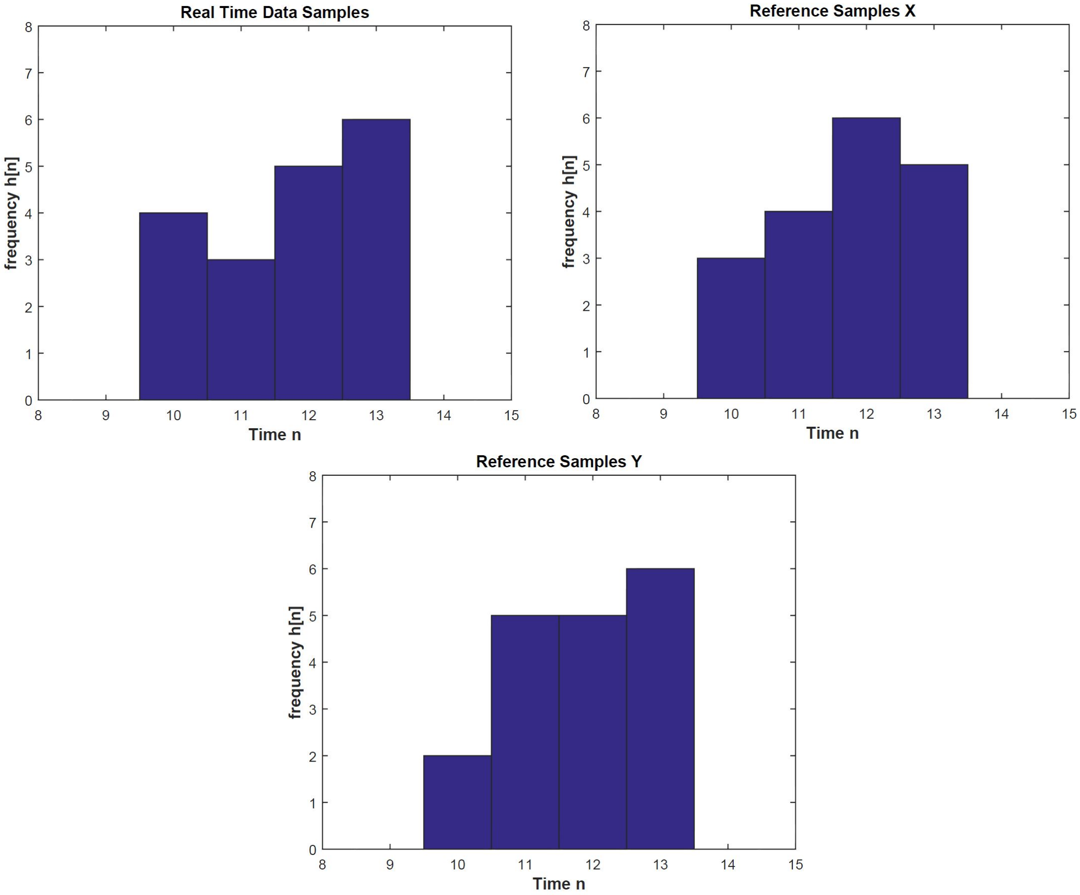


Figure 6.3: (a) RTT samples measured in real-time (b) X (c) Y

It is important to note that the Euclidean distance utilizes the raw samples to distinguish the two sets of RTT samples while the Bhattacharyya Coefficient requires the relative frequency of occurrence of each unique sample, i.e. probability mass function. Computing the Euclidean Distance and Bhattacharyya Coefficient we find that ***ED(R,X) =1.4142*** and ***BC(R,X)=0.9935***. Similarly, ***ED(R,Y)=1.4142*** and ***BC(R,Y)=0.9834***.

Table 6.1: Mean and Variance of RTT samples

Data Sample	μ	σ^2
R	11.72	1.3117
X	9.5	1.089
Y	11.83	1.021

We can see from the above example that even though X and Y have different distributions, the Euclidean distance yields the exact same result. It is important to note that all three data sets have different mean and variance values, Euclidean distance (Eqn. 6.10) cannot distinguish between them. If X and Y represent the RTT reference data sets for say 10 m and 20 m distances, Euclidean distance cannot determine if the target is 10 m or 20 m from the reference node. However, with Bhattacharyya coefficient, we can easily make that distinction and predict the range result. This is an important property since the RTT data collected at different distances in indoor environments can be very similar and in such scenarios, this clear distinction can avoid incorrect ranging of nodes. Thus, the Bhattacharyya coefficient and Bhattacharyya distance prove to be more robust statistical distance measure than Euclidean distance.

6.5 Statistical TOA Algorithm

Referring to Figure 4.2, the Target node collects RTT data samples by transmitting unicast data to the Reference node and waiting for the ACK frame to be received. It then transfers the RTT data samples file to a Statistical TOA program whose algorithm is given below. The Statistical TOA ranging (matching) stage involves matching the real time RTT data with the stored reference databases for each distance using the Bhattacharyya coefficient and Bhattacharyya distance measures.

1. First, we construct a probability density function for the collected RTT data samples using the ***ksdensity*** function in MATLAB.

2. The reference databases were formed during the offline stage based on the analysis in the previous chapter. We have stored reference databases for 10, 20, 30 m since the number of in-range peer nodes are the highest at these distances.

3. The next step is to construct probability density functions for each stored reference database.

4. Compute the Bhattacharyya coefficient and Bhattacharyya distance values for collected RTT data samples and each stored reference database (Equations 6.4, 6.5).

5. The reference database that is closest to this RTT data distribution is the one with the highest ***BC*** value or the lowest $\boldsymbol{D_B}$ value. This is distance of the target node from the reference node.

6.6 Kernel Density Estimation

In the algorithm mentioned above, we indicated the use the probability density function (pdf) as an input to the Bhattacharyya coefficient computation. In Equation 6.4, ***p(x)*** and ***q(x)*** can be either histograms (probability mass functions) or probability density functions. Even though the histogram of the samples can be used, it yields in unstable results. When we construct a histogram, we need to consider the width of the bins and the end points of the bins (where each of the bins start). As a result, the problems with histograms are that they are not smooth, and can have a different shape depending on the width of the bins and end points of the bins chosen.

To avoid such instability, we use the Kernel Density Estimation. Kernel density estimators belong to a class of estimators called non-parametric density estimators. In comparison to parametric estimators where the estimator has a fixed functional form (structure) and the parameters of this function are the only information we need to store, non-parametric estimators have no fixed structure and depend upon all the data points to reach an estimate. To remove the dependence on the end points of the bins, kernel estimators center a kernel function at each data point. And if we use a smooth kernel function for our building block, then we will have a smooth density estimate. This way we can eliminate the two problems associated with histograms [27]. As indicated earlier, the kernel density estimate in MATLAB is ***ksdensity***.

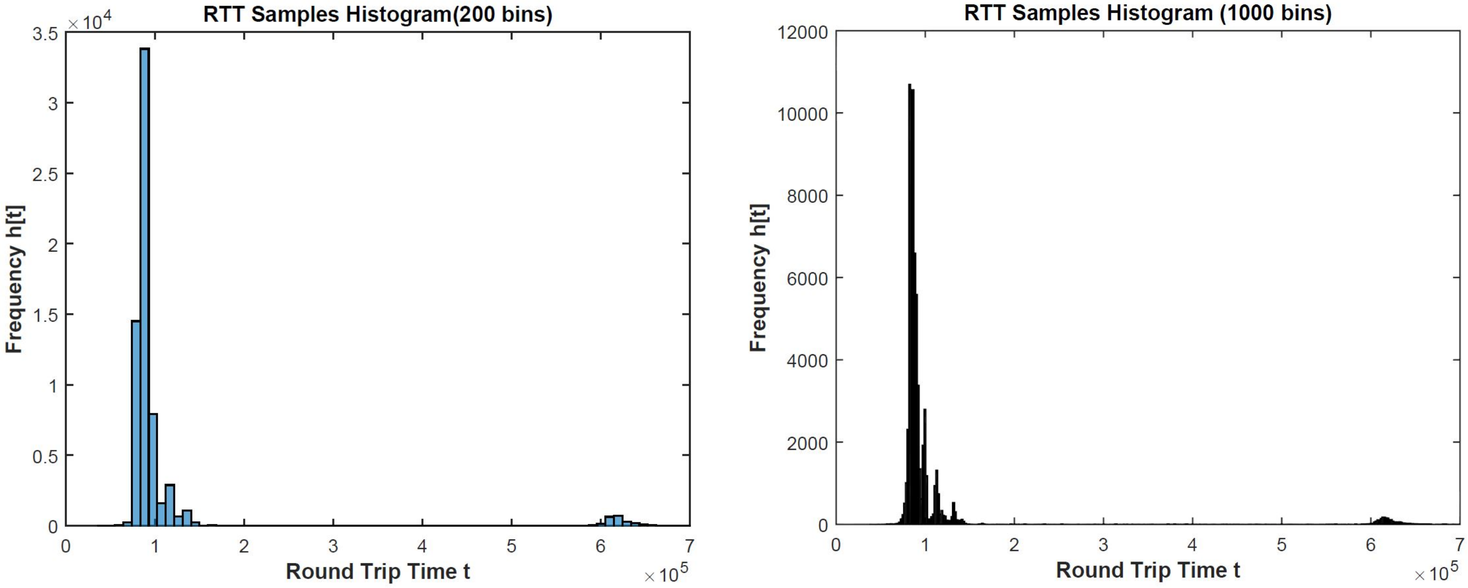


Figure 6.4: RTT samples Histogram (a) 200bins (b) 1000bins

Figure 6.4 depicts how the shape of the histogram changes with number of bins and hence width of bins. We can see how several smaller peaks show up when the number of bins is increased or in other words, the width of the bins is reduced. If we apply Bhattacharyya coefficient and distance measures to each of these with its corresponding reference databases (with similar bin widths), it yields different results. Since we measure Round-Trip time at the driver layer, the value is dominated by the processing times and the RTT values between different distances (say 10 m and 20 m) are very similar. In such cases, we need a unique RTT data representation and for this reason, we employ the Kernel Density Estimators over histogram representations.

6.7 Kernel Probability Distribution vs Gaussian Probability Distribution

In the previous chapter, we assumed the RTT samples measured to be a Gaussian distribution to discuss the required sample size of the RTT samples. While this was a good assumption for a modelling, we cannot use this assumption while employing the Bhattacharyya coefficient and distance. In indoor environments, we have seen that the RTT samples collected at different distances have similar distributions. In such cases, fitting a Gaussian distribution to the data would lead to ranging inaccuracies.

To prove this point, let us consider three RTT sample distributions we collected at three different distances. Figures 6.6, 6.5, 6.7 represent the histograms of the RTT samples measured at 15 m, 18 m and 30 m respectively. Also shown is the probability density functions of each when the RTT data is fit with Kernel and Gaussian probability distribution.

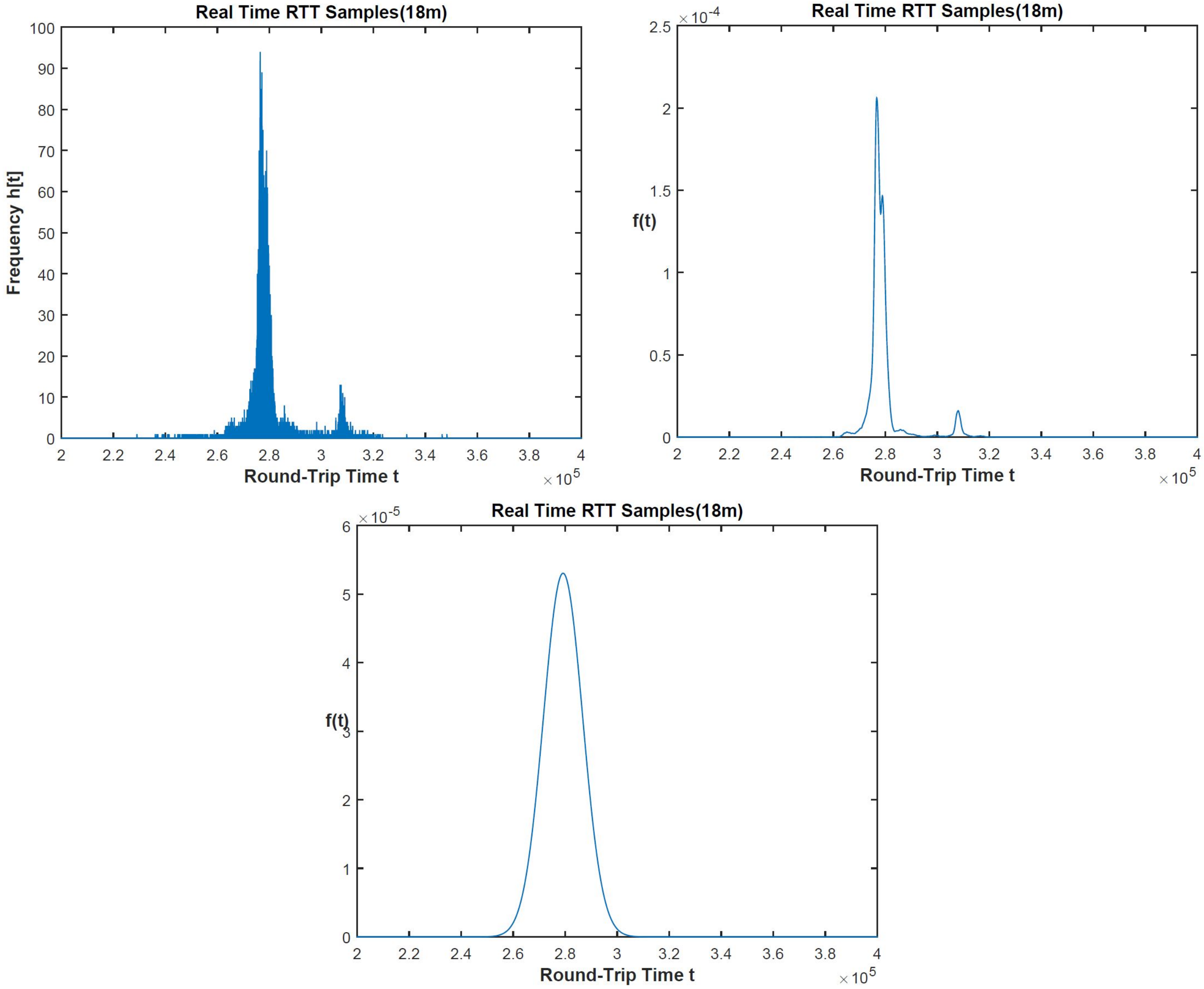


Figure 6.5: RTT (ns) samples measured in real-time (18 m) (a) Histogram (b) Kernel Density (c) Gaussian Distribution

Let us assume our reference database consists of the RTT distributions ***R1*** (15 m) and ***R2*** (30 m) and the RTT samples ***T*** (18 m) are collected in real time. Table 6.2 below shows the result of Bhattacharyya coefficient and distance for Gaussian and kernel probability distributions.

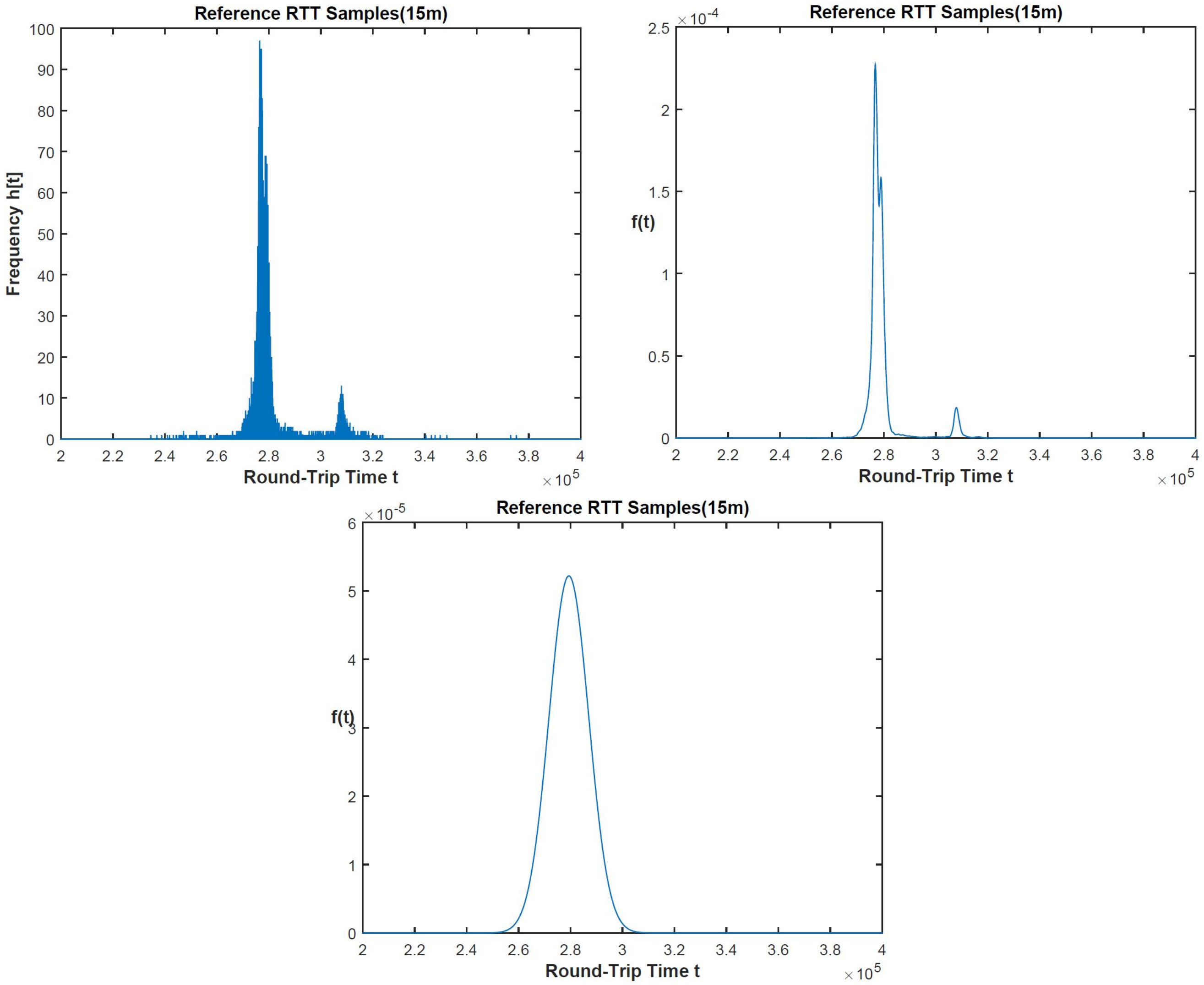


Figure 6.6: Reference RTT (ns) Samples (15 m) (a) Histogram (b) Kernel Density(c) Gaussian Distribution

Referring to the Kernel distribution fit for the RTT data from the above table, since the BC value is highest for ***R1***, we can say that ***T*** is 15 m away from reference node ***R1*** and this would be an accurate result considering the range error is 3 m when compared to 12 m with ***R2***. However, with a Gaussian distribution fit, we cannot make that distinction.

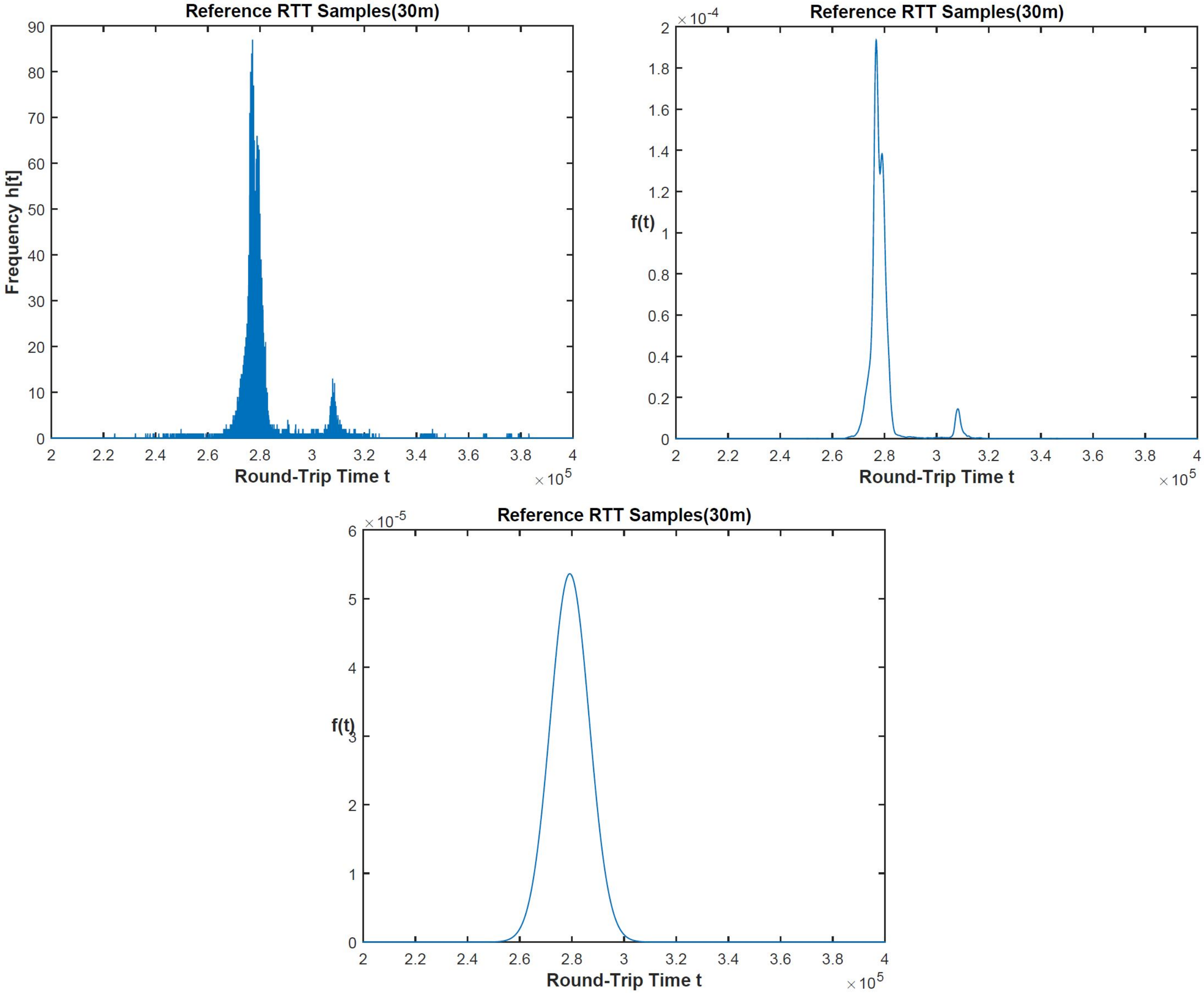


Figure 6.7: Reference RTT (ns) Samples (30 m) (a) Histogram (b) Kernel Density(c) Gaussian Distribution

Matching	*BC(p,q)*		$D_B(p,q)$	
	Gaussian Distribution	Kernel Distribution	Gaussian Distribution	Kernel Distribution
(T,R1)	0.9999	0.9951	1×10^{-4}	4.912×10^{-3}
(T,R2)	0.9999	0.9915	1×10^{-4}	8.536×10^{-3}

Table 6.2: *BC(p,q)* for Kernel and Gaussian distribution fits

Chapter 7

Performance Evaluation

In this chapter, we discuss some experimental results for the Statistical TOA ranging method we proposed in the previous two chapters. We first discuss how we build the reference database followed by some use cases where we applied our statistical TOA ranging methodology in the real world.

7.1 Building the Reference Database

Our reference database consists of RTT samples collected at 10 m, 20 m, 30 m distances in different indoor environments. We chose these distances because there is a high probability of finding large number of reference peer nodes in-range with the target at these distances. If a target is at say 13 m from a reference node, the most accurate ranging result would be 10 m since it reduces the overall range error. We then use the LMI method to determine the position of the target while also reducing the error. This was explained in the previous chapters and in [16]. In this section, we detail how we select the reference database by using the Bhattacharyya coefficient and distance measures.

We collected RTT data samples at various indoor locations and at different distances as detailed in Chapter: 5. We then grouped the measured RTT samples based on distance i.e. 10, 20 and 30 m. As explained in the Section: 5.2.2, we identified different types of probability distributions in different environments. Some have dominant modes (peaks) located at 82-83 μs or 96-98 μs, some with peaks at 276-277 μs or 343-344 μs and some with multiple peaks at approximately 86 μs, 90 μs, 98 μs, 111 μs, 130 μs. So we grouped the RTT samples for each distance based on these types. In order to determine the location of the most dominant peak and distinguish between different probability distributions, we used a 10 μs window which contained the maximum number

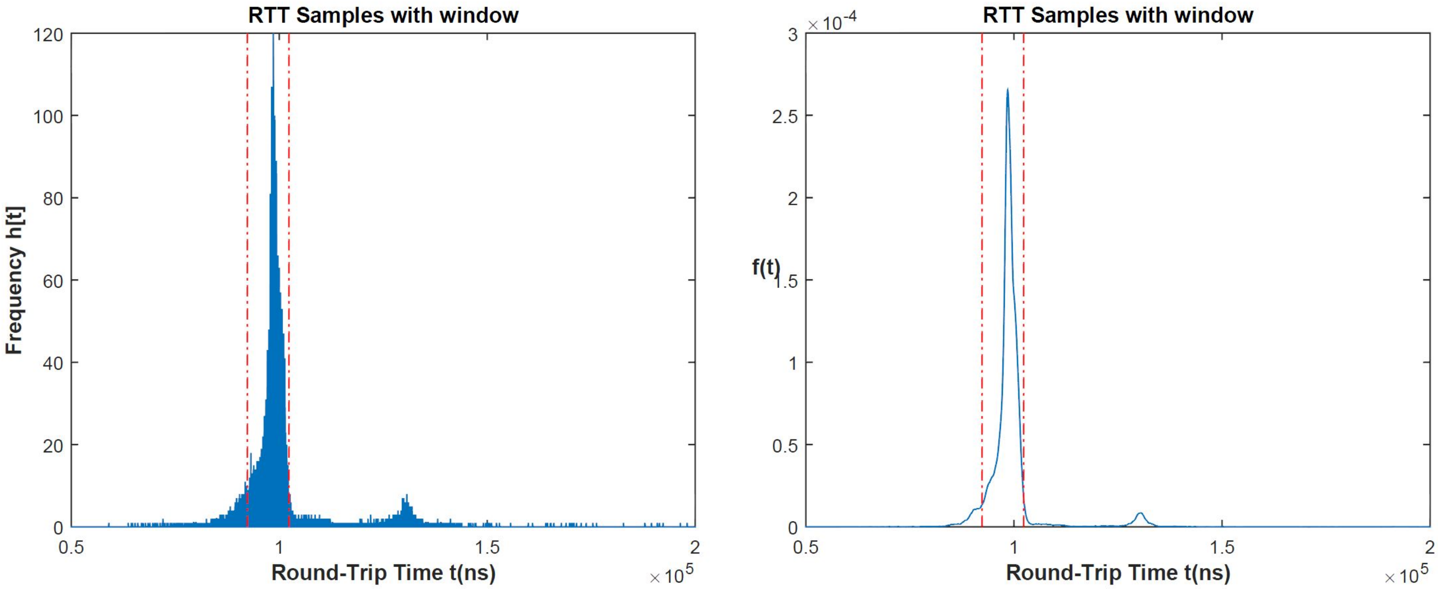


Figure 7.1: RTT (ns) data with 10 μs window (a) Histogram (b) pdf

of samples in the histogram. We chose a 10 μs window because it usually encloses the dominant mode in the distribution. Alternatively, we could also determine the dominant mode by finding two valleys on either side of the peak. Figure 7.1 shows the 10 μs window.

Another key observation is that RTT data samples with dominant modes at 276-277 μs have different distributions depending on the indoor environment. When there is a complete NLOS condition between the target and reference nodes in the hallway, RTT samples tend to have window (10 μs) mean lesser than 276 μs and those RTT samples collected in open space or multiroom environment tend to have a larger mean. The RTT samples in open space and multiroom environments are then differentiated based on the heights of the less dominant peaks.

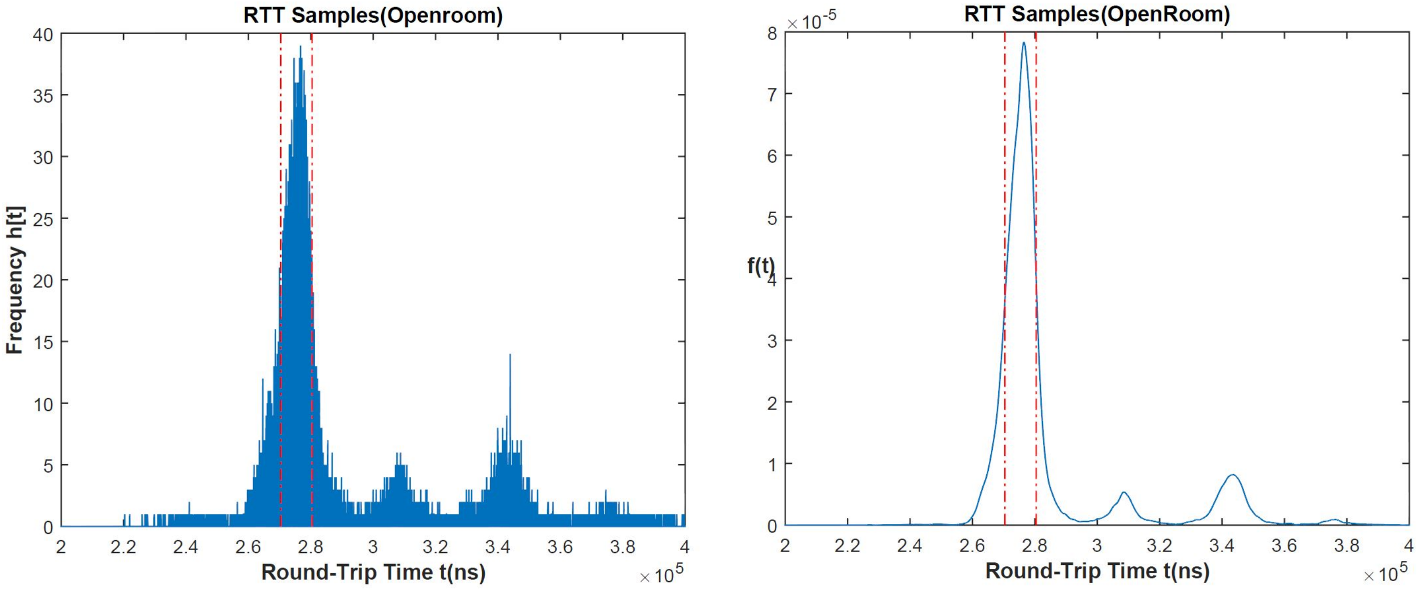


Figure 7.2: RTT (ns) data collected in an Open Space Environment(a) Histogram (b) pdf

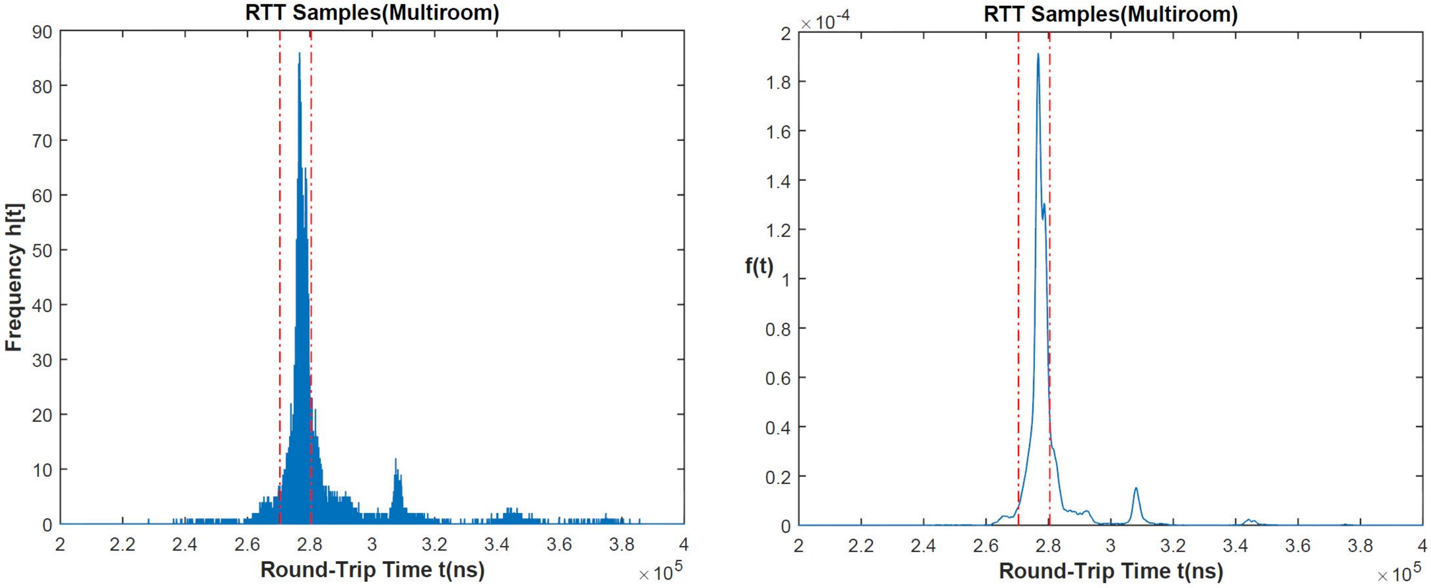


Figure 7.3: RTT (ns) data collected in a Multiroom Environment (a) Histogram (b) pdf

Figures 7.2, 7.3 show the RTT sample distribution collected at 20 m in an open space and multiroom environment. We can see the third mode (peak) is taller in Figure 7.2 than in Figure 7.3. We believe this is because of increased processing time due to packet losses. In an open space environment, when the target and reference nodes are in a complete NLOS condition, data packets can be corrupted when RF signals are diffracted and reflected off the objects. This triggers the retransmission procedure which then increases the processing time.

This classification of RTT samples into different groups for the reference database can be summarized as shown in Figure 7.4. Once these classifications are made, each group has several RTT sample files for each distance. To choose the best RTT samples representation for each group in each distance, we use the Bhattacharyya coefficient measure.

We first divide all the RTT sample files into 10 m, 20 m, 30 m groups and then determine the mean of all the samples that lie in 10 μs window that contains the maximum number of samples. The files are then separated into several subgroups as shown in the Figure 7.4. We then take each sample file in say *subgroup1* of ***10 m*** group and match it with every other sample file in the same sub-group using the Bhattacharyya coefficient. We also match this file with other files in *subgroup1* of ***20 m*** and ***30 m*** groups. If this file yields high ***BC(p,q)*** values with files in 10 m (*subgroup1*) and low ***BC(p,q)*** values with other files in *subgroup1* of 20 m, 30 m groups, this can be an accurate 10 m RTT sample representation for *subgroup1* of 10 m. We repeat this procedure for every other file of each subgroup in each distance group.

7.2 Use Cases

The experiments were conducted using the Mini PCs running FreeBSD operating system. The configuration is as mentioned in Table 5.1. All the computations and simulations have been performed in MATLAB. Here we discuss a few test cases. As discussed in the Statistical TOA algorithm, the probability density function for each RTT data set is constructed and matched against the reference databases using the Bhattacharyya coefficient and distance measures.

7.2.1 Experiment 1 : Shelby Hallway

The map in Figure 7.5 shows the target node placed in one hallway and six reference nodes distributed in the other hallway on the second floor of Shelby building such that all the reference and target nodes are in NLOS condition with each other. Reference node 1, 2 are about 10 m from

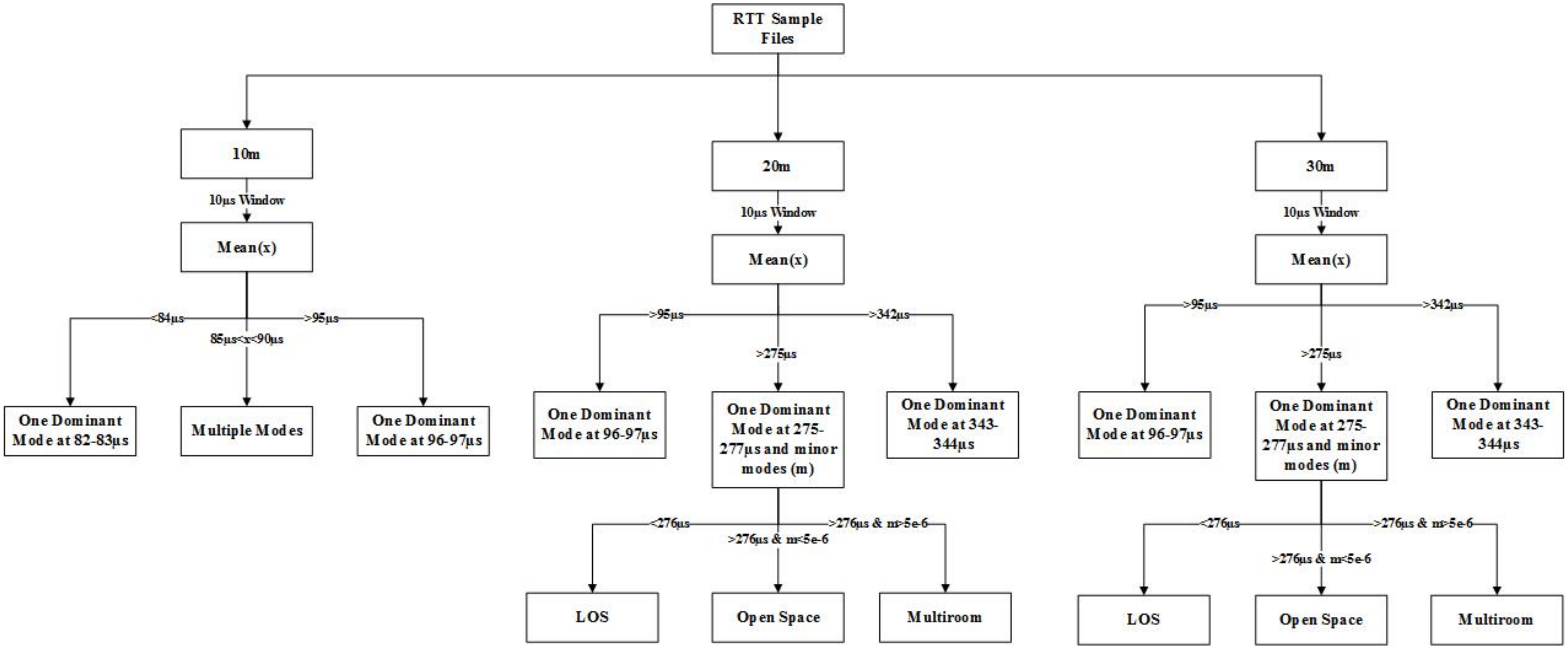

RTT Sample Files
10m
10µs Window
Mean(x)
<84µs
85µs<x<90µs
>95µs
One Dominant Mode at 82-83µs
Multiple Modes
One Dominant Mode at 96-97µs
20m
10µs Window
Mean(x)
>95µs
>275µs
>342µs
One Dominant Mode at 96-97µs
One Dominant Mode at 275-277µs and minor modes (m)
One Dominant Mode at 343-344µs
<276µs
>276µs & m<5e-6
>276µs & m>5e-6
LOS
Open Space
Multiroom
30m
10µs Window
Mean(x)
>95µs
>275µs
>342µs
One Dominant Mode at 96-97µs
One Dominant Mode at 275-277µs and minor modes (m)
One Dominant Mode at 343-344µs
<276µs
>276µs & m<5e-6
>276µs & m>5e-6
LOS
Open Space
Multiroom

Figure 7.4: Decision Tree to Select Reference Databases

the target, nodes 3, 4 are about 20 m and nodes 5, 6 are about 30 m from the target. Refer Figure 5.7 for a legend to read this map (Fig. 7.5).

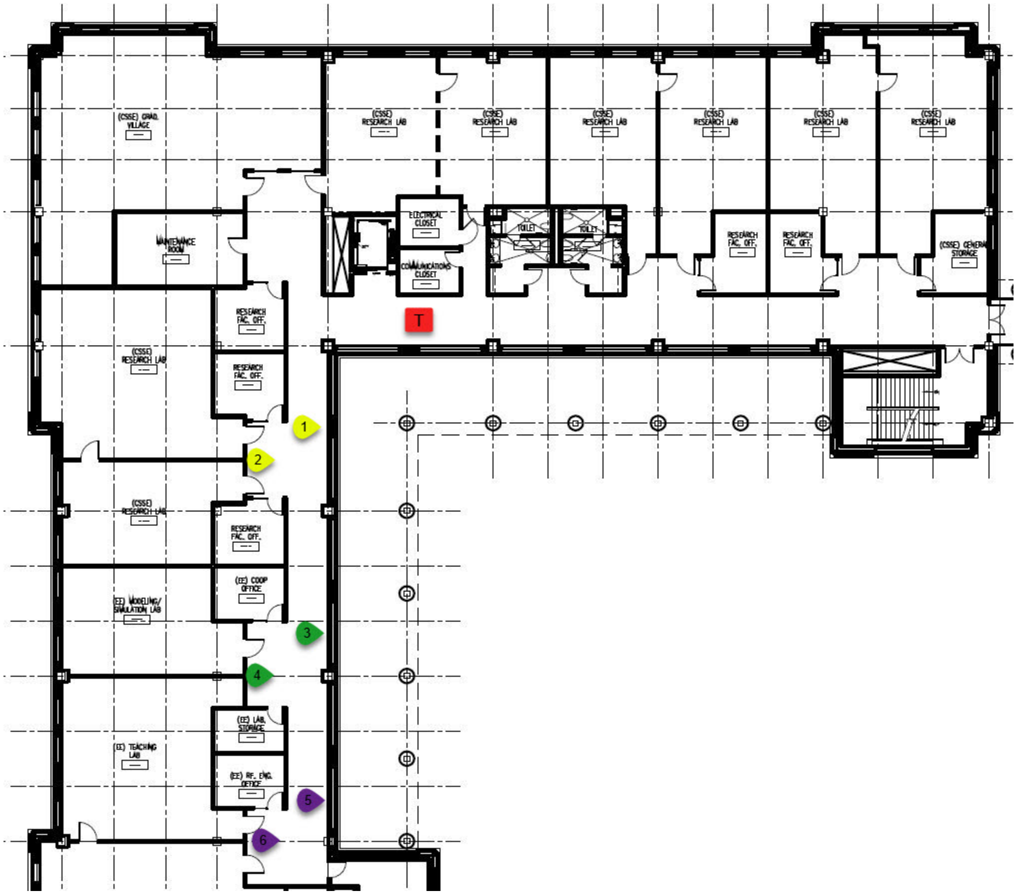


Figure 7.5: Experiment 1-target and reference nodes positions

The target node communicates with each reference node at a time and collects *65536* RTT samples. Once the samples are collected, the probability density function (pdf) of the data is constructed and finally matched with the reference databases (Refer Section: 6.5). The reference database with the highest ***BC(p,q)*** value or the lowest $\boldsymbol{D_B(p,q)}$ value is the estimated range between the target and that reference node.

Assuming the target position coordinate is *(0,0)* the Table 7.1 gives the exact position coordinates (X-axis-towards left of Fig. 7.5, Y-axis-downward) of the 6 reference nodes, their corresponding ***BC(p,q)*** values when matched against the reference databases, and the predicted distances using the Statistical TOA algorithm detailed in Section: 6.5.

Reference Nodes	Reference Coordinates	*Bhattacharyya Coefficient BC(p,q)*			Predicted Distance
		10m	20m	30m	
#1 (10.57m)	(6,8.7)	**0.9856**	0.9739	0.9744	10m
#2 (14m)	(8.5,11.1)	**0.9586**	0.9363	0.9314	10m
#3 (20.3m)	(6,19.4)	xxxx	**0.9689**	0.9265	20m
#4 (23.4m)	(8.5,21.8)	xxxx	**0.9857**	0.9422	20m
#5 (30.2m)	(6,29.6)	xxxx	0.9451	**0.9784**	30m
#6 (33.1m)	(8.5,32)	xxxx	**0.9596**	0.9116	20m

Table 7.1: Experiment 1 Summary

We first determine the 10 μs window which contains the maximum number of samples and then based on the decision tree we match with the appropriate databases. Reference nodes 3, 4, 5, 6 have empty 10 m reference database entries since the RTT samples collected at the target by communicating with these reference nodes have a window mean of approximately 276 μs and are of the open space type (see Fig. 7.4) and such distributions are found to not occur at 10 m distances.

7.2.2 Experiment 2 : Shelby Front Lobby

Figure 7.6 shows the target node placed resting on a wall and the six reference nodes distributed in the front lobby of Shelby building such. Reference node 1, 2, 3 are about 10 m from the target and in LOS, while nodes 4, 5 are 20 m and in complete NLOS with each other.

The target node communicates with each reference node at a time and collects *65536* RTT samples. Once the samples are collected, the probability density function (pdf) of the data is constructed and finally matched with the reference databases (Refer Section: 6.5). The reference database with the highest ***BC(p,q)*** value or the lowest $\boldsymbol{D_B(p,q)}$ value is the estimated range between the target and that reference node.

Assuming the target position coordinate is *(0,0)* the Table 7.2 gives the exact position coordinates of the 6 reference nodes, their corresponding ***BC(p,q)*** values when matched against the reference databases, and the predicted distances using the Statistical TOA algorithm detailed in Section: 6.5.

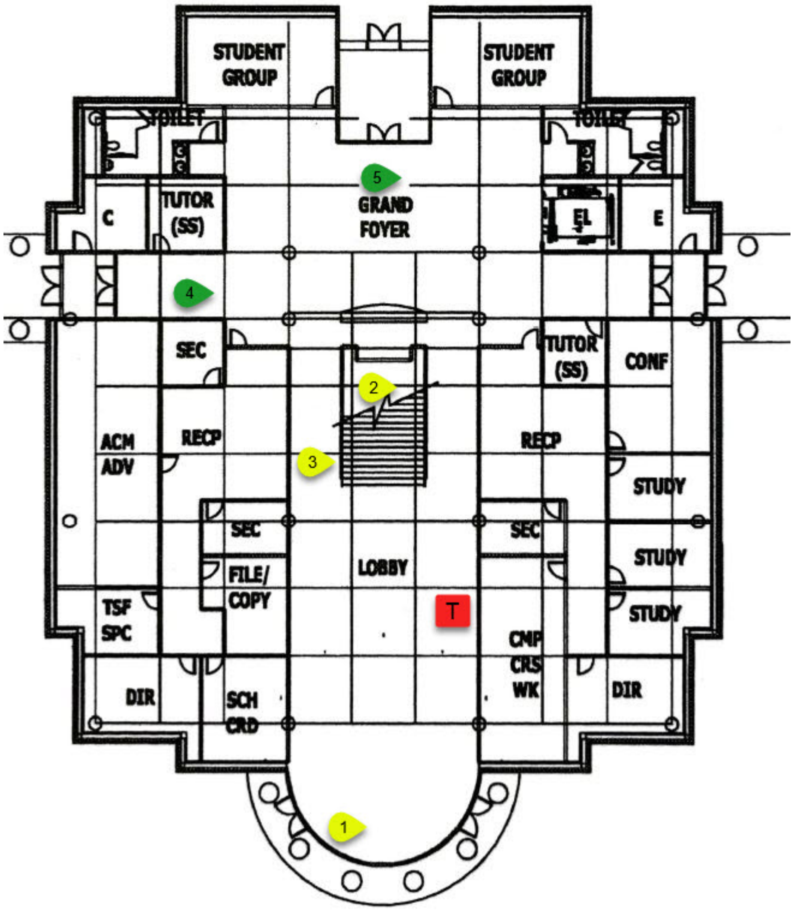


Figure 7.6: Experiment 2-target and reference nodes positions

Reference Nodes	Reference Coordinates	*Bhattacharyya Coefficient BC(p,q)*			Predicted Distance
		10m	20m	30m	
#1 (10.3m)	(3.3,9.75)	xxxx	xxxx	xxxx	10m
#2 (10.1m)	(4.35,-9.1)	**0.9882**	0.9649	0.3228	10m
#3 (10m)	(7.3,-6.83)	**0.9876**	0.9622	0.9618	10m
#4 (19.9m)	(13.2,-15)	xxxx	**0.9834**	0.9429	20m
#5 (19.9m)	(3.4,-19.68)	xxxx	**0.9366**	0.9091	20m

Table 7.2: Experiment 2 Summary

We first determine the 10 μs window which contains the maximum number of samples and then based on the decision tree we match with the appropriate databases. In the Table 7.2, reference nodes 1 is predicted at 10 m distance without computing the ***BC(p,q)*** value because the window mean is approximately 82 μs and we have found that all RTT samples distributed with such window mean values are at 10 m distances only. Reference nodes 4, 5 have empty 10 m reference database entries since the RTT samples collected at the target by communicating with these reference nodes have a window mean of approximately 276 μs and are of the open space type (see Fig. 7.4) and such distributions are found to not occur at 10 m distances.

7.2.3 Experiment 3 : Shelby Research Labs and Hallways

Figure 7.7 shows the target node placed in our research lab 2323 in the Shelby building and reference nodes distributed in the adjacent rooms and in the hallway. Reference node 1-5 are at about 10 m from the target, while nodes 6 and 7 are about 20 m and 30 m respectively.

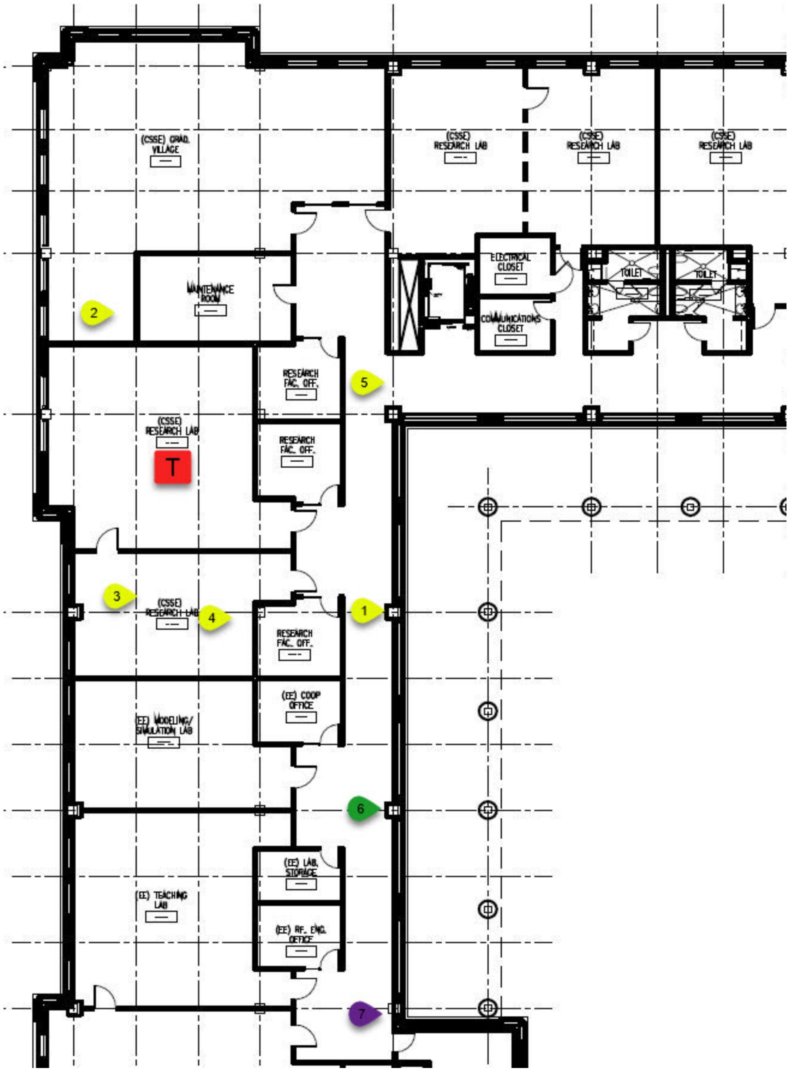


Figure 7.7: Experiment 3-target and reference nodes positions

The target node communicates with each reference node at a time and collects *65536* RTT samples. Once the samples are collected, the probability density function (pdf) of the data is constructed and finally matched with the reference databases (Refer Section: 6.5). The reference database with the highest ***BC(p,q)*** value or the lowest $\boldsymbol{D_B(p,q)}$ value is the estimated range between the target and that reference node.

Assuming the target position coordinate is *(0,0)* the Table 7.3 gives the exact position coordinates (X-axis-towards right of Fig. 7.7, Y-axis-upward) of the 6 reference nodes, their corresponding ***BC(p,q)*** values when matched against the reference databases, and the predicted distances using the Statistical TOA algorithm detailed in Section: 6.5.

Reference Nodes	Reference Coordinates	*Bhattacharyya Coefficient BC(p,q)*			Predicted Distance
		10m	20m	30m	
#1 (13m)	(9.53,-8.99)	**0.9904**	0.9704	0.4863	10m
#2 (8.3m)	(-3.96,7.24)	xxxx	xxxx	xxxx	10m
#3 (8.6m)	(-2.74,-8.1)	0.9668	**0.9846**	0.3389	20m
#4 (9.8m)	(3.89,-8.99)	xxxx	xxxx	xxxx	10m
#5 (11.7m)	(9.53,6.78)	**0.9843**	0.9685	0.3265	10m
#6 (22.7m)	(9.53,-20.57)	0.9489	**0.9755**	0.3619	20m
#7 (32.4m)	(9.53,-30.93)	0.1439	0.1500	0.9450	30m

Table 7.3: Experiment 3 Summary

We first determine the 10 μs window which contains the maximum number of samples and then based on the decision tree we match with the appropriate databases. In the table above, reference nodes 2 and 4 are predicted at 10 m distance without computing the ***BC(p,q)*** value because the window mean is approximately 82 μs and we have found that all RTT samples distributed with such window mean values are at 10 m distances only.

7.2.4 Average Error

As mentioned in Section: 7.1, our reference databases consists of RTT samples files for 10, 20 and 30 m distances for different indoor environments as shown in Figure 7.4 which means our Statistical TOA method predicts from one of these three possibilities. So if the actual distance between the reference and target node is say 12 m from the target, the most accurate prediction made by our Statistical TOA ranging method would be 10 m since it reduces the overall range error. This was one of our assumptions we made in Section: 4.2. Average error is given as:

$$Avg.Error = \frac{\sum_r (ActualDist(r) - PredictedDist(r))}{numberofref.nodes} \tag{7.1}$$

The table 7.4 shows the average range error for each of the use cases we discussed in Section: 7.2. This error is a part of the system and the subsequent stages of our indoor localization system-LMI with Barycenter or Center of Gravity algorithm can significantly reduce this error and provide accurate position coordinates of target node.

Experiment	Avg. Error
#1	3.6 m
#2	3.12 m
#3	3.3 m

Table 7.4: Average Error

7.2.5 Experiment 4 - A Special Use Case

In Section 7.2,we showed several online (real-time) experiments that we conducted in different indoor locations. The target node sends a DATA packet to the reference node to which it responds with an ACK frame and the target node measures the difference between these two timestamps as one RTT sample. In this way, it collects a total of *65536 (2^{16})* samples. The target node repeats this process in a round-robin fashion communicating with each reference node one after the other. Referring Figure 7.7, when target communicates with reference node 1, the time taken to collect *65536 (2^{16})* is roughly 90-120 seconds. So this experiment would take approximately 10-14 minutes to communicate with each reference node. This is not entirely practical because reference nodes could go out-of-range during that period.

In this section, we propose an alternate solution. Firstly, going back to the discussion in Section: 5.2.1, instead of *65536 (2^{16})* samples, we can collect *16384 (2^{14})* samples since the overall distribution and shape would be quite similar to the *65536 (2^{16})* case. This would reduce the time taken by a factor of 4 to approximately 22-30 seconds. Also, instead of round-robin scheduling, the target node can communicate simultaneously with several reference nodes using different threads for reference nodes. However, this can lead to several packet losses and errors due to interference.

Since the distribution of RTT samples is slightly different and simultaneous communication can cause packet losses, we need a rethink of reference databases. But we can follow the method described in Figure 7.4 and Section: 7.1 to create a new reference database.

We conducted this experiment with a new device which has a better processor and a smaller form factor compared to the previous we used in Table 5.1. The specifications are shown in Table 7.5.

Table 7.5: Device Specifications

Processor	Intel Celeron N3160
CPU Frequency	1.6 GHz(upto 2.24 GHz)
RAM	4096MB (4GB)
OS Kernel	FreeBSD 10.1-RELEASE i386
32 bit/64bit	64bit
Wireless NIC	Ubiquiti Networks SR-71 (mini-PCIe)
WLAN protocol	IEEE 802.11 b/g
Operating Frequency Range	2.4 GHz
Data Rate	24 Mbps(OFDM)
Antenna Characteristics	TRENDnet Dual-Band 11a/g 7/5dBi Indoor Omni Directional Antenna (TEW-AI75OB)

With repeated experimentation with this new device, we have found that there are 5 different types of RTT data distributions for each distance(10, 20, 30 m). These are shown in Figure 7.8. So our new reference databases consist of 5 folders each consisting of RTT reference data samples for 10, 20 and 30 m distances.

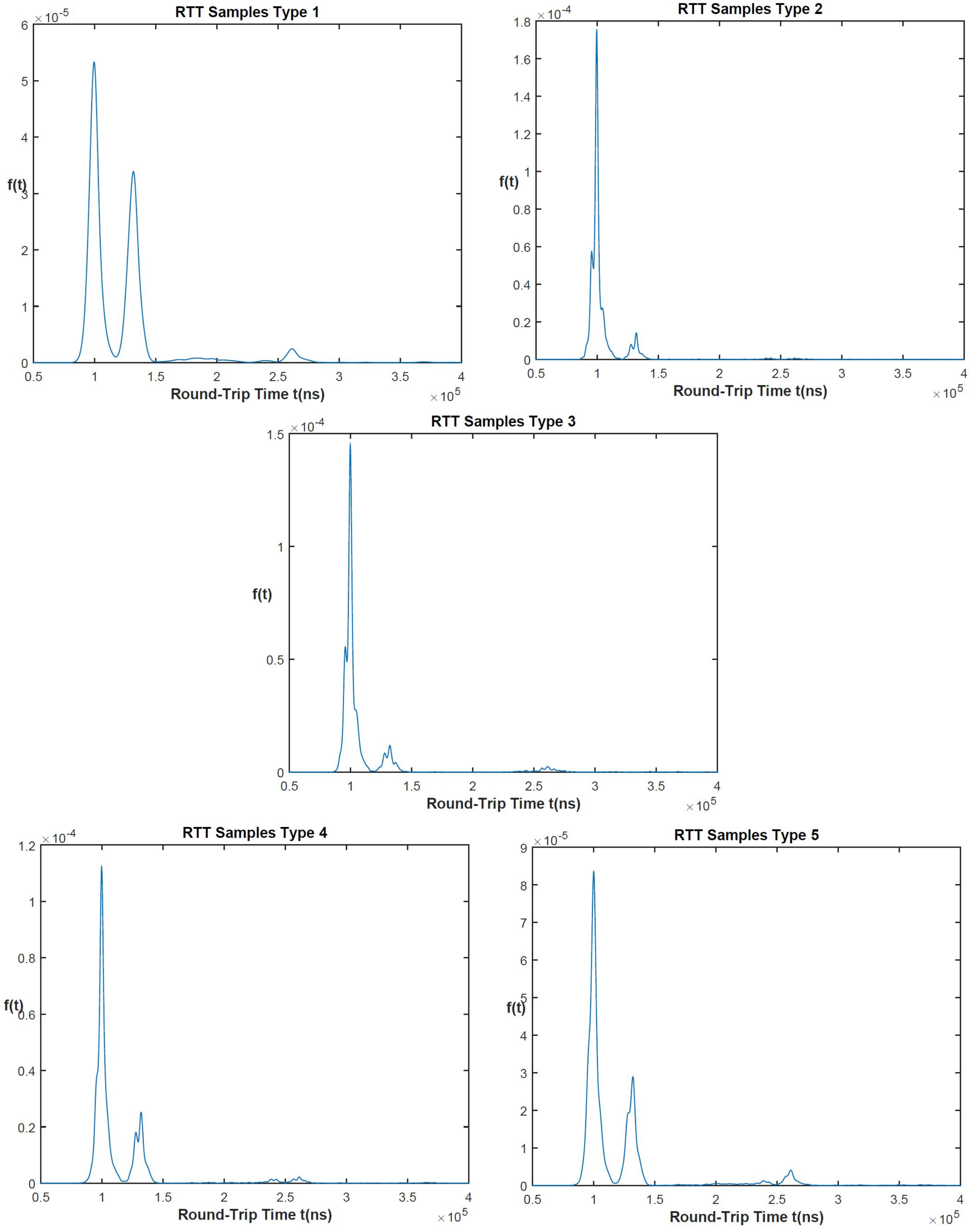


Figure 7.8: Reference RTT Samples (pdf) (a) Type-1 (b) Type-2 (c) Type-3 (d) Type-4 (e) Type-5

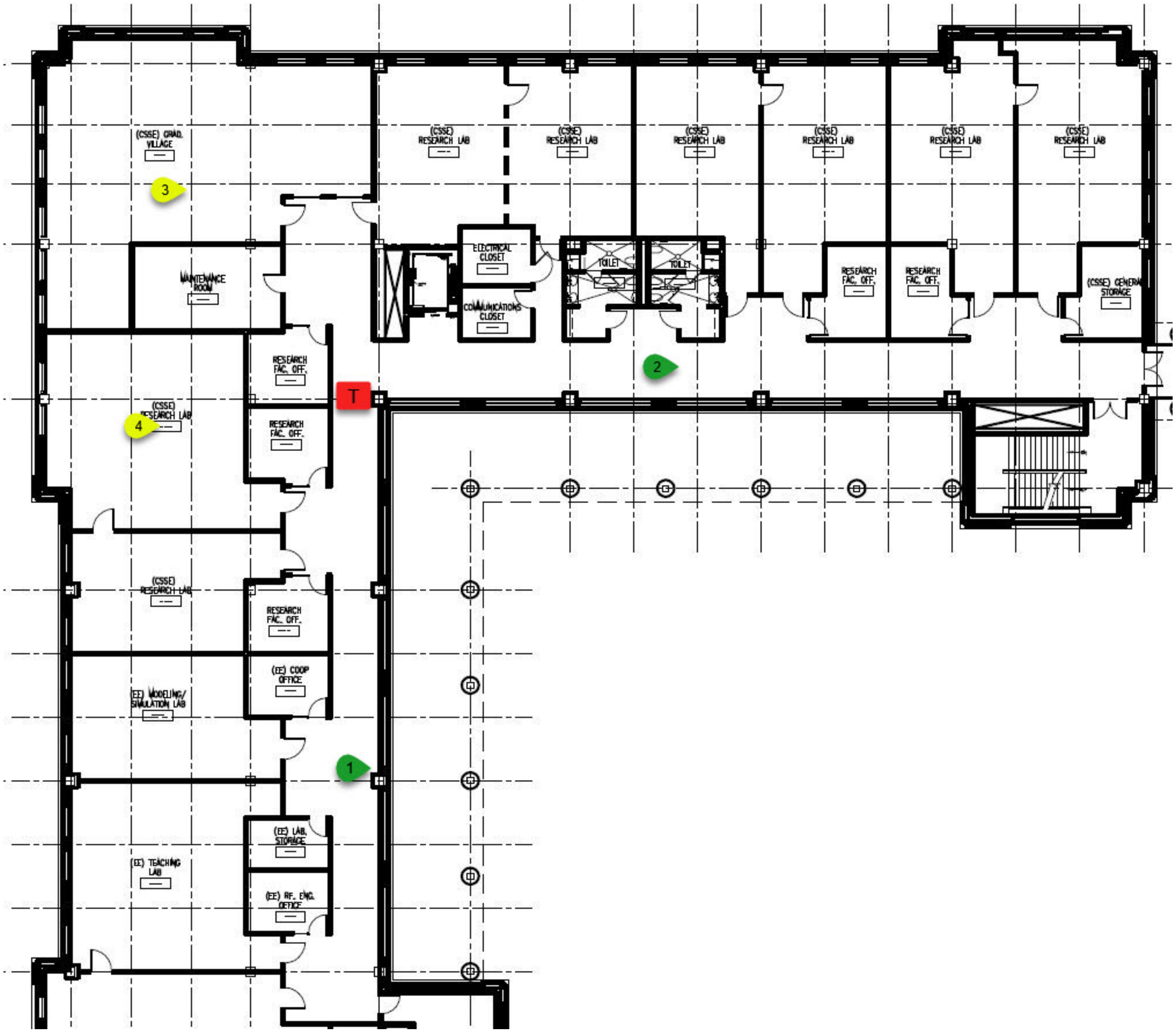


Figure 7.9: Experiment 4-simultaneous reference node communication

Figure 7.9 shows target and reference nodes locations on the map. Each reference node communicates with the target node simultaneously and we collect *16384 (2^{14})* samples. We then employ the algorithm in Section: 6.5 to predict the range of each reference from the target.

Ref. Nodes	Coordinate (x,y)	Dist. Type	*Bhattacharyya Coefficient BC(p,q)*			Predicted Distance
			10m	20m	30m	
1 (18m)	(15,20)	2	0.9880	**0.9929**	0.9892	20m
2 (16m)	(31,39)	4	0.9655	**0.9701**	xxxx	20m
3 (13.5m)	(6,48)	5	**0.9871**	0.9821	0.9705	10m
4 (10m)	(5,37)	2	**0.9848**	0.9806	0.9809	10m

Table 7.6: Experiment 4 Summary

The average error as per Table 7.6 is approximately 2.3 m.

7.2.6 Percentage of Accurate Prediction

In the previous sections we provided the best results of our Statistical TOA ranging method. However, since the RTT data measured over different distances are all very similar to each other, our algorithm can predict incorrect results. In each position on the maps shown in previous sections, we collect several RTT sample files and the table below summarizes the percentage of accurate predictions in each experiment.

Experiment #	Accurate Prediction%
1	72%
2	76%
3	71%
4	63%

Table 7.7: Percentage of Accurate Prediction

7.3 Wireshark View

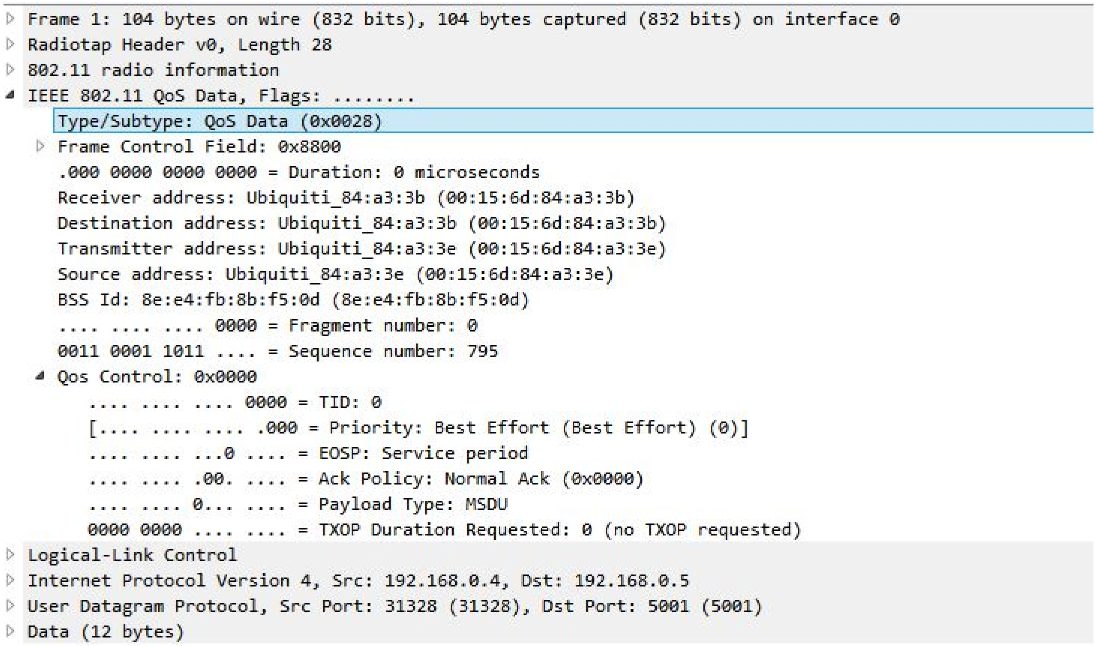
In this section, we provide a Wireshark view of the DATA and ACK frames that are transmitted between the reference and target nodes. There are several packet sniffing tools like Riverbed's AirPCAP but these would sniff the packets that it receives, but we have no way to tell if the target node received those frames. As a solution to this, we run Wireshark packet capture on the target node itself by configuring a new virtual interface different from the one that is used for transmitting the packets.

Wireshark is an open source packet capture and analysis tool used essentially for network troubleshooting, analysis, software and communications protocol development, and in education for academic and teaching purposes [57]. *Wireshark* and its command line tool *Tshark* can be installed and used on Linux, BSD and Windows systems. It's functionality is like 'tcpdump' but it has a graphical front-end and a lot of sorting and filtering mechanisms for better viewing. As a packet capture tool, Wireshark can be used to capture packets that are addressed to the computer on which it is running. The wired or wireless network interface card (NIC) passes the captured packets to the CPU where it is decoded based on different protocols and displayed by Wireshark.

The IEEE 802.11 wireless NIC can be put into seven modes: Access Point, Client, Ad-hoc, Mesh, Repeater, Promiscuous, and Monitor mode. The promiscuous and monitor modes are essentially used for packet sniffing. In the promiscuous mode, the NIC passes all traffic to the CPU, even the ones not intended to it. On the other hand, the monitor mode also passes the traffic to the CPU but in this case, the Wireless NIC does not have to associate with an access point or ad-hoc network. Monitor mode is used for geographical packet analysis, observing of widespread traffic; especially for unsecure channels (such as through WEP). This mode is useful during the design phase of Wi-Fi network construction to discover how many Wi-Fi devices are already using spectrum in a given area and how busy are the Wi-Fi channels in that area. This can help in planning better Wi-Fi networks and reduce interference with other Wi-Fi devices [53]. But the monitor mode does not allow transmission of packets and is limited to one channel. So, in our experimentation, we create a new virtual interface using 'ifconfig', set it to monitor mode on the channel on which the target node is transmitting the DATA frames. This interface is different from the one that is used to transmit frames over the air.

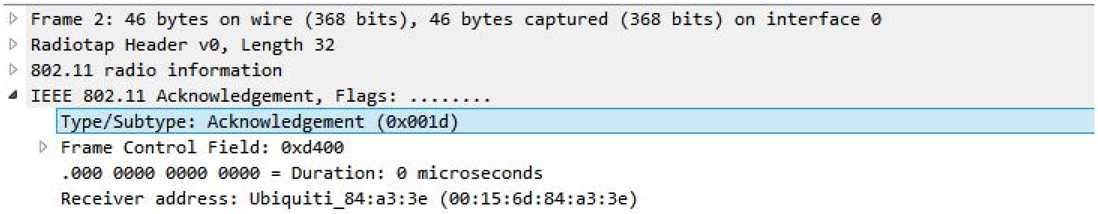
The target and reference nodes are placed at 10, 20 and 30 m distances apart and the Wireshark packet capture program (tshark on FreeBSD) is run to sniff packets transmitted and received by the target during the period it communicates with the reference nodes to collect *65536 (2^{16})* RTT samples. This is repeated for LOS and NLOS conditions along the hallway (Fig. 5.8 and 7.5) and the multiroom (Fig. 7.7) scenario where the nodes are placed in different rooms. Figures 7.10, 7.11, 7.12 show a sample wireshark packet trace of DATA and ACK frames.

Since the wireless nodes are operating in IEEE 802.11g mode, the transmission of every DATA frame is preceded by the reception of ACK frame for the previous DATA frame. Filtering the packets on wireshark based on time of capture, we can see this pattern in Figure 7.12 (see packets 1 through 20). If the ACK frame is not received within a time out period, the DATA frame is retransmitted (usually at PHY layer not captured by Wireshark) and if those retransmissions



```
▷ Frame 1: 104 bytes on wire (832 bits), 104 bytes captured (832 bits) on interface 0
▷ Radiotap Header v0, Length 28
▷ 802.11 radio information
◢ IEEE 802.11 QoS Data, Flags: ........
    Type/Subtype: QoS Data (0x0028)
  ▷ Frame Control Field: 0x8800
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Ubiquiti_84:a3:3b (00:15:6d:84:a3:3b)
    Destination address: Ubiquiti_84:a3:3b (00:15:6d:84:a3:3b)
    Transmitter address: Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e)
    Source address: Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e)
    BSS Id: 8e:e4:fb:8b:f5:0d (8e:e4:fb:8b:f5:0d)
    .... .... .... 0000 = Fragment number: 0
    0011 0001 1011 .... = Sequence number: 795
  ◢ Qos Control: 0x0000
      .... .... .... 0000 = TID: 0
      [.... .... .... .000 = Priority: Best Effort (Best Effort) (0)]
      .... .... ...0 .... = EOSP: Service period
      .... .... .00. .... = Ack Policy: Normal Ack (0x0000)
      .... .... 0... .... = Payload Type: MSDU
      0000 0000 .... .... = TXOP Duration Requested: 0 (no TXOP requested)
▷ Logical-Link Control
▷ Internet Protocol Version 4, Src: 192.168.0.4, Dst: 192.168.0.5
▷ User Datagram Protocol, Src Port: 31328 (31328), Dst Port: 5001 (5001)
▷ Data (12 bytes)
```

Figure 7.10: Wireshark DATA packet trace

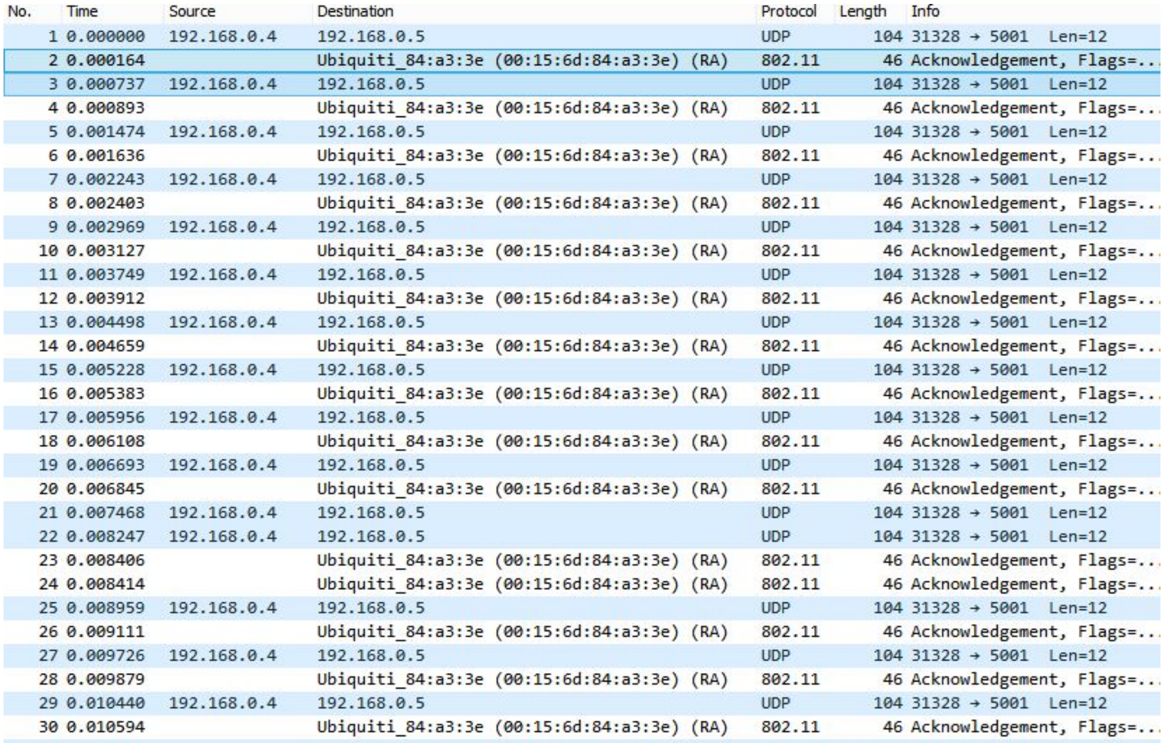


```
▷ Frame 2: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface 0
▷ Radiotap Header v0, Length 32
▷ 802.11 radio information
◢ IEEE 802.11 Acknowledgement, Flags: ........
    Type/Subtype: Acknowledgement (0x001d)
  ▷ Frame Control Field: 0xd400
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e)
```

Figure 7.11: Wireshark ACK packet trace

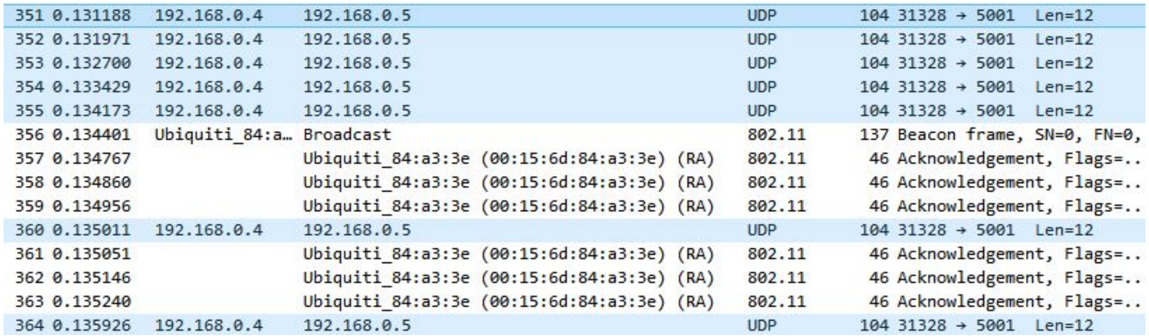
exceed a certain threshold, it is discarded and a new DATA frame is transmitted (see packet 21 through 24 in Fig.7.12). Analyzing the time of packet capture, we see that the approximate time between a DATA and ACK frame capture is roughly 160-180 μs and that between two DATA frames is about 700-800 μs.

From the packet trace collected during the time period in which the target node communicates with the reference nodes to collect *65536 (2^{16})* RTT samples, we count the number of occurrences of DATA+ACK frame sequences (like packets 1 to 20) which have a time difference of less than 200 μs, while neglecting all the multiple DATA followed by multiple ACK sequences (Fig. 7.13). The multiple DATA and multiple ACK sequences usually occur because the ACK was not received within a timeout period which triggers a low level (PHY layer) retransmission.



```
No.   Time      Source       Destination                                Protocol  Length  Info
    1 0.000000  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
    2 0.000164               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
    3 0.000737  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
    4 0.000893               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
    5 0.001474  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
    6 0.001636               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
    7 0.002243  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
    8 0.002403               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
    9 0.002969  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   10 0.003127               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   11 0.003749  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   12 0.003912               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   13 0.004498  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   14 0.004659               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   15 0.005228  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   16 0.005383               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   17 0.005956  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   18 0.006108               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   19 0.006693  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   20 0.006845               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   21 0.007468  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   22 0.008247  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   23 0.008406               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   24 0.008414               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   25 0.008959  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   26 0.009111               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   27 0.009726  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   28 0.009879               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
   29 0.010440  192.168.0.4  192.168.0.5                                UDP          104  31328 → 5001  Len=12
   30 0.010594               Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=...
```

Figure 7.12: DATA and ACK frame sequence



```
351 0.131188  192.168.0.4     192.168.0.5                                UDP          104  31328 → 5001  Len=12
352 0.131971  192.168.0.4     192.168.0.5                                UDP          104  31328 → 5001  Len=12
353 0.132700  192.168.0.4     192.168.0.5                                UDP          104  31328 → 5001  Len=12
354 0.133429  192.168.0.4     192.168.0.5                                UDP          104  31328 → 5001  Len=12
355 0.134173  192.168.0.4     192.168.0.5                                UDP          104  31328 → 5001  Len=12
356 0.134401  Ubiquiti_84:a…  Broadcast                                  802.11       137  Beacon frame, SN=0, FN=0,
357 0.134767                  Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=..
358 0.134860                  Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=..
359 0.134956                  Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=..
360 0.135011  192.168.0.4     192.168.0.5                                UDP          104  31328 → 5001  Len=12
361 0.135051                  Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=..
362 0.135146                  Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=..
363 0.135240                  Ubiquiti_84:a3:3e (00:15:6d:84:a3:3e) (RA)  802.11        46  Acknowledgement, Flags=..
364 0.135926  192.168.0.4     192.168.0.5                                UDP          104  31328 → 5001  Len=12
```

Figure 7.13: Multiple DATA and ACK frame sequence

The number of occurrences of DATA+ACK frame sequence indicates the number of DATA packets that are ‘ACK’nowledged immediately without any retransmission or some retransmissions lesser than the threshold. We define the ratio of this to the total number of DATA packets in the Wireshark capture as the *‘DATA frames Goodput Ratio’*.

$$DATA\ Frame\ Goodput\ Ratio = \frac{no.\ of\ (DATA + ACK)\ frame\ sequences}{total\ no.\ of\ DATA\ frames} \tag{7.2}$$

Figures 7.14 shows the goodput ratio for different distances in a LOS, NLOS and multiroom scenarios (Fig. 5.8, 7.5, 7.7).

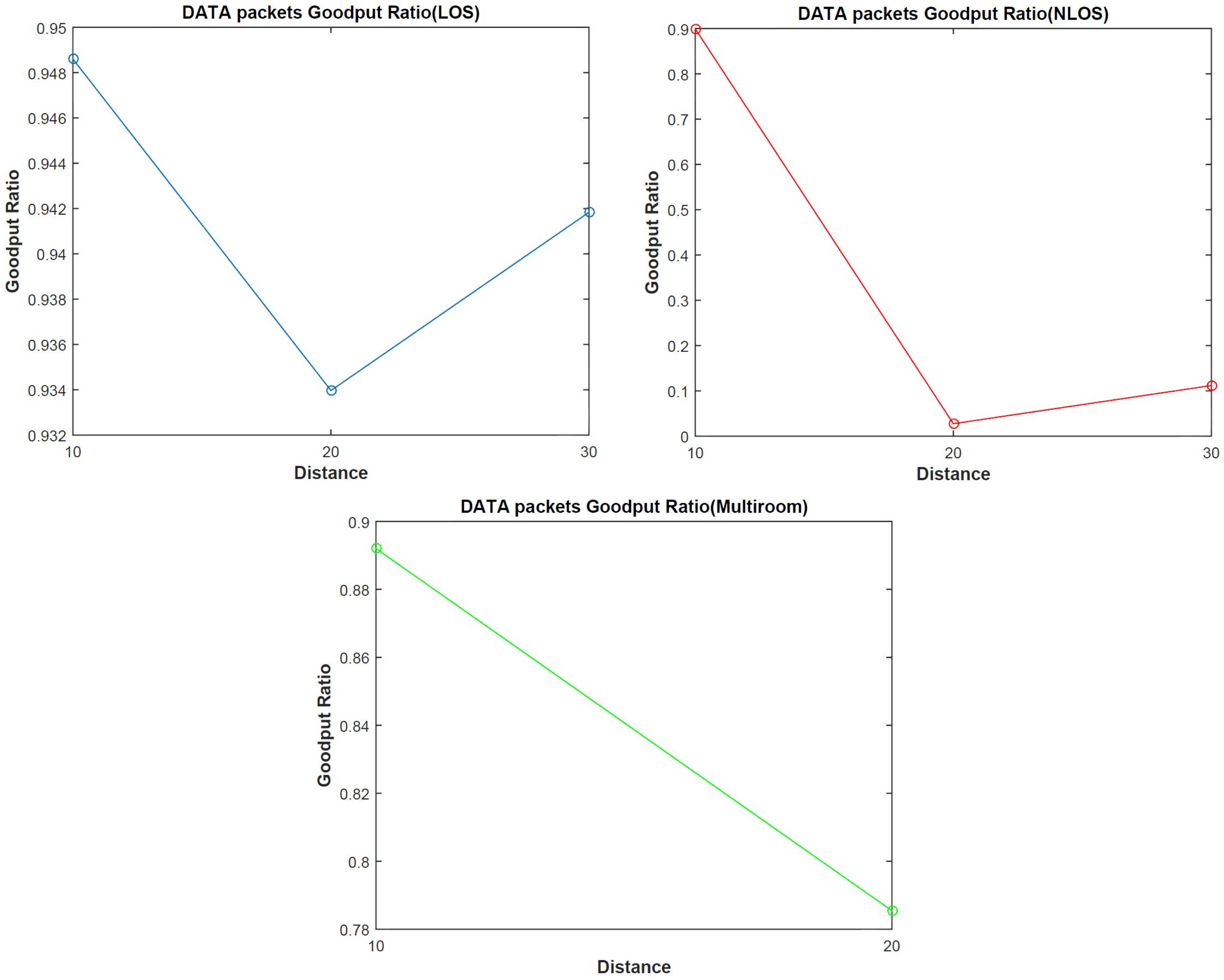


Figure 7.14: DATA frames Goodput Ratio (a) LOS (b) NLOS (c) Multiroom

In Section: 5.2.2, we discussed different types of RTT data probability distributions in different environments. Some have dominant modes (peaks) located at 96-98 μs or 276-277 μs or 343-344 μs. This behavior is due to the packet retransmissions which may increase the processing times and since we measure the RTT at the driver layer, these get added to the measured RTT value.

As mentioned in Section: 7.1, we use the mean of a 10 μs window which encloses this dominant mode to distinguish between different types of RTT distributions. Figure 7.1 below shows the 10 μs window. Figure 7.15 shows the data packets goodput ratio for different 10 μs window mean. The graph shows that as the mean increases, the goodput ratio decreases and this presents an intuitive explanation for the quantum shifts seen in the distribution of the measured RTT data. If the goodput ratio is small, it indicates that DATA frames are not followed by ACK frames which means there is retransmissions causing high average of RTT data.

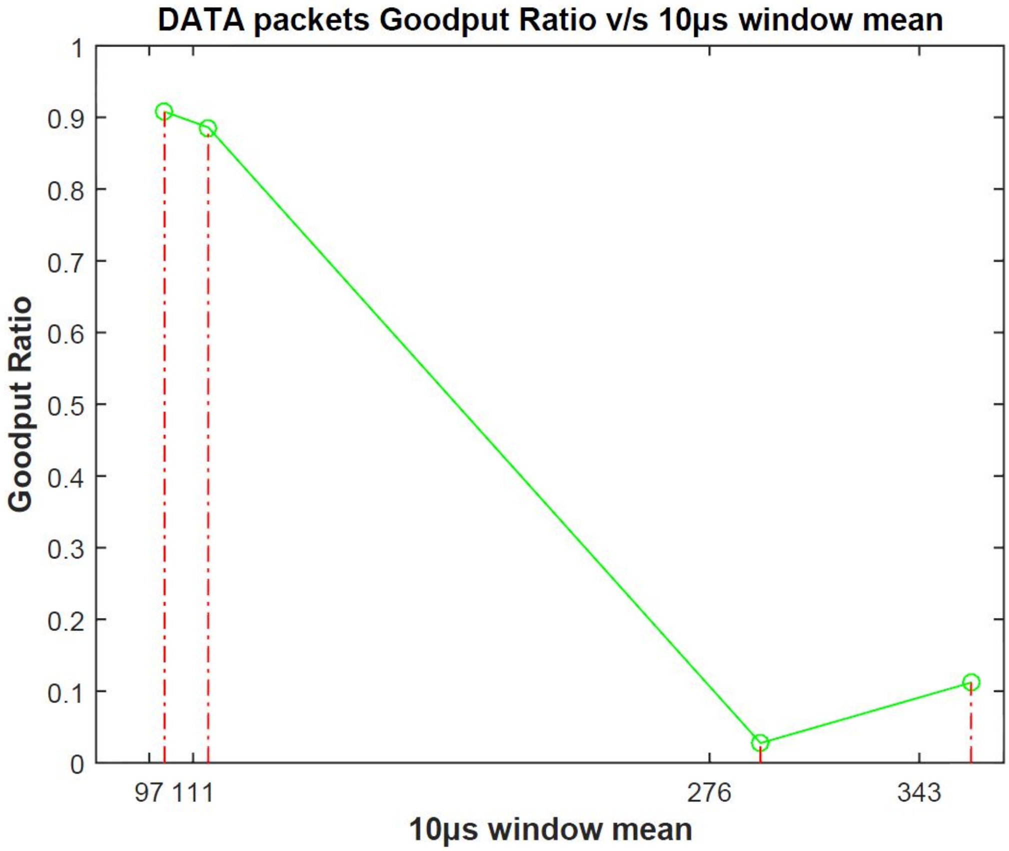


Figure 7.15: Goodput ratio v/s 10 μs mean

Chapter 8

CONCLUSION

8.1 Conclusion

In this thesis work, we describe our indoor localization system which involves a location unaware target node communicating with in-range reference peer nodes opportunistically over Wi-Fi to determine its position coordinates. The target and reference nodes operate in IEEE 802.11g mode at 2.4 GHz. It involves 3 stages: the first is the measurement of the Round-Trip Time (RTT) between transmission of a *DATA* frame and the reception of an *ACK* frame. This time is measured at the driver layer of the OS kernel. The second stage is ranging where the distance between target and reference nodes is predicted. Literature defines several ranging methods like RSS, AOA, TOA, but we used the TOA/RTT ranging due to its better accuracy compared to RSS and it does not need special hardware as AOA. We propose a new algorithm called Statistical Time-of-Arrival ranging to perform ranging between target and reference nodes. RTT values are severely affected by multipath making it difficult to determine the range. Our Statistical TOA algorithm has two parts: an offline stage where RTT measurements are made in different indoor environments to build a reference database for each distance. In the online stage, RTT samples measured in real-time is matched against these stored reference databases to predict the range. This way we can deal with multipath effectively as they will be included in the databases. The third and final stage is called position estimation which takes as input the position coordinates of reference peer nodes and their predicted range to estimate position coordinates of the target. Position estimation utilizes Linear Matrix Inequality (LMI) technique to estimate the position and takes Center-of-Gravity (CoG) of a cluster of estimated target positions to improve the accuracy.

The main contributions of this thesis work includes:

- Field Testing - RTT measurements were made in different indoor environments and at different distances.
- Analysis - RTT measurements were analyzed and compared with frame timings in IEEE 802.11g standard. Reference Database was built each distance.
- Statistical TOA algorithm - developed an algorithm to predict the range between target and reference nodes

RTT samples measured was found to have discrete/quantum shifts in the dominant modes (peaks) in the distribution. Some had dominant peaks at 96-98 μs, 276-277 μs or 343-344 μs. We think that this is due to the packet retransmission causing increased processing times. Using Wireshark packet capture, we defined Goodput ratio as the percentage of data packets that are acknowledged immediately without retransmissions or some retransmissions. Goodput ratio was found to decrease as the peaks shifted towards 276 μs proving our intuition.

Reference databases were built by creating a decision tree as in Figure 7.4 and choosing a subset of RTT samples collected using Bhattacharyya coefficient.

We used Bhattacharyya coefficient as the statistical distance measure to match the real-time RTT samples against stored reference databases. We showed how this performs better than the Euclidean distance. We performed several experiments in the real world to test the ranging accuracy. The overall average error was found to be 3.08 metres in the best case. The prediction strategy is such that if the actual range is say 14 metres, the predicted range is 10 metres instead of 20 metres since it reduces the error. Thus, a more relevant accuracy indicator would be the percentage of accurate range prediction which was found to be approximately 71%.

8.2 Future Work

RTT measurements were made during the field testing stage in different indoor environments and a reference database was built using Bhattacharyya coefficient to select a subset of these measurements. The distribution functions of RTT samples were analyzed to choose appropriate parameters to choose reference databases. This was detailed in Section: 7.1. In the future this process can be left to a machine learning algorithm that creates a decision tree on its own with a few training parameters. The parameters might change if the databases have to be refreshed and machine learning algorithms could learn over time and take data driven decisions, thus making it more autonomous.

Also, RTT samples include frame processing times which can vary if more applications are running on the nodes. To eliminate this processing time from affecting the measured RTT, timestamps need to be collected when the frame is just transmitted (over the air) and just received. This would eliminate the processing time anomalies. But, this would require access and understanding of the firmware code (PHY layer) in the wireless NIC which is usually proprietary. There is a need to build or choose a device where the PHY layer is accessible to the developer.

The result of the position estimation stage using LMI can be fed back to the Statistical TOA ranging stage to correct range results or eliminate old unused reference databases.

Appendix A

Wireshark Packet Capture data

LOS/NLOS samples	DATA frames #	ACK frames #	Total # of frames	DATA+ACK Sequence #	Goodput Ratio
LOS-10m_1	85939	85948	172226	81741	0.9512
LOS-10m_2	86618	86623	173579	82691	0.9547
LOS-10m_3	96691	96705	193802	90876	0.9399
NLOS-10m_1	35871	35873	173292	32557	0.9076
NLOS-10m_2	91822	91834	184043	82782	0.9015
NLOS-10m_3	95686	95696	191788	85048	0.8888

Table A.1: Hallway-10m

LOS/NLOS samples	DATA frames #	ACK frames #	Total # of frames	DATA+ACK Sequence #	Goodput Ratio
LOS-20m_1	88443	88450	177253	82714	0.9352
LOS-20m_2	93119	93126	186682	87579	0.9405
LOS-20m_3	89477	89485	179314	82870	0.9262
NLOS-20m_1	97622	97630	195645	2934	0.0301
NLOS-20m_2	99063	99074	198515	2772	0.0280
NLOS-20m_3	94922	94933	190241	2322	0.0245

Table A.2: Hallway-20m

LOS/NLOS samples	DATA frames #	ACK frames #	Total # of frames	DATA+ACK Sequence #	Goodput Ratio
LOS-30m_1	93927	93934	188260	87297	0.9294
LOS-30m_2	88274	88284	176888	84602	0.9584
LOS-30m_3	87082	87091	174530	81656	0.9377
NLOS-30m_1	439896	439949	881639	30658	0.0697
NLOS-30m_2	210141	210166	421120	24995	0.1189
NLOS-30m_3	295821	295860	592847	43438	0.1468

Table A.3: Hallway-30m

Multiroom samples	DATA frames #	ACK frames #	Total # of frames	DATA+ACK Sequence #	Goodput Ratio
10m_1	90826	90835	182353	81016	0.8920
10m_1	94147	94160	189049	83072	0.8824
10m_1	104046	104058	208907	88173	0.8474

Table A.4: Maintenance Room-10m

Multiroom samples	DATA frames #	ACK frames #	Total # of frames	DATA+ACK Sequence #	Goodput Ratio
10m_1	96782	96791	193965	88536	0.9148
10m_1	92683	92694	185755	85234	0.9196
10m_1	99884	99895	200186	92211	0.9232

Table A.5: Server Room-10m

Multiroom samples	DATA frames #	ACK frames #	Total # of frames	DATA+ACK Sequence #	Goodput Ratio
20m_1	92949	92958	186292	78318	0.8426
20m_1	96414	96422	193241	82310	0.8537
20m_1	96414	96422	193241	82310	0.8537

Table A.6: Measurements Lab 2326-20m

Multiroom samples	DATA frames #	ACK frames #	Total # of frames	DATA+ACK Sequence #	Goodput Ratio
10m_1	90573	90583	181939	82452	0.9103
10m_2	85519	85529	171817	78142	0.9137
10m_3	97543	97551	195999	85830	0.8799
10m_4	100534	100544	201900	84135	0.8369
20m_1	112662	112655	225809	84271	0.7480
20m_2	124106	124105	249203	89833	0.7238
20m_3	118273	118284	237337	80849	0.6836

Table A.7: Lab 2319-10m, 20m

Bibliography

[1] In: *Wi-Fi Location-Based Services 4.1 Design Guide*. Cisco Systems Inc., 2008. Chap. 2.

[2] *"About FreeBSD"*. URL: `{https://www.freebsd.org/about.html}`.

[3] F. Aherne, N. Thacker, and P. Rockett. "The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data". In: *Kybernetika* 34.4 (1998), pp. 363–368.

[4] N. Alsindi, X. Li, and K. Pahlavan. "Analysis of Time of Arrival Estimation Using Wideband Measurements of Indoor Radio Propagations". In: *IEEE Transactions on Instrumentation and Measurement* 56.5 (2007), pp. 1537–1545. ISSN: 0018-9456. DOI: `10.1109/TIM.2007.904481`.

[5] *"Atheros 802.11n PCI/PCI-E devices (ath9k)"*. URL: `{https://wiki.debian.org/ath9k#supported}`.

[6] B. Peterson and C. Kmiecik and R. Hartnett and P. Thompson and J. Mendoza and H. Nguyen. "Spread Spectrum Indoor Geolocation". In: *Navigation* 45.2 (1998), pp. 97–102. ISSN: 2161-4296.

[7] P. Bahl and V. N. Padmanabhan. "RADAR: an in-building RF-based user location and tracking system". In: *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*. Vol. 2. 2000, 775–784 vol.2. DOI: `10.1109/INFCOM.2000.832252`.

[8] M. Bshara et al. "Fingerprinting Localization in Wireless Networks Based on Received-Signal-Strength Measurements: A Case Study on WiMAX Networks". In: *IEEE Transactions on Vehicular Technology* 59.1 (2010), pp. 283–294. ISSN: 0018-9545. DOI: `10.1109/TVT.2009.2030504`.

[9] M. Burton. *White paper:"802.11 Arbitration"*. Tech. rep. Durham,NC: Certified Wireless Network Professional Inc., 2009. URL: `https://www.cwnp.com/uploads/802-11_arbitration.pdf`.

[10] S.-H. Cha. "Comprehensive survey on distance/similarity measures between probability density functions". In: *International Journal of Mathematical Models and Methods in Applied Science* 1.4 (2007), pp. 1–8.

[11] Y. Chen and H. Kobayashi. "Signal strength based Indoor geolocation". In: *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333)*. Vol. 1. New York, NY, USA, 2002, pp. 436–439. DOI: `10.1109/ICC.2002.996891`.

[12] A. Daund and P.K. Vyas. "Wireless Broadband Access with the Application of IEEE 802.11b based Wi-Fi Model". In: *International Journal of Computer Applications* 154.5 (2016), pp. 39–44. ISSN: 0975-8887. DOI: `10.5120/ijca2016912143`.

[13] L. Doherty, K. S. J. pister, and L. El Ghaoui. "Convex position estimation in Wireless Sensor Networks". In: *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*. Vol. 3. 2001, 1655–1663 vol.3. DOI: `10.1109/INFCOM.2001.916662`.

[14] E. Elnahrawy, Xiaoyan Li, and R. P. Martin. "The limits of localization using signal strength: a comparative study". In: *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004*. 2004, pp. 406–414. DOI: `10.1109/SAHCN.2004.1381942`.

[15] FreeBSD Documentation. *"Wireless Networking"*. URL: `{https://www.freebsd.org/doc/handbook/network-wireless.html}`.

[16] Darren Handler. "An Island of Chaos Surrounded by a Sea of Confusion: The E911 Wireless Device Location Initiative". In: *Virginia Journal of Law and Technology* 10.1 (2015).

[17] *Hardware Overview*. URL: {https://wiki.freebsd.org/dev/ath_hal(4)/HardwareOverview}.

[18] J. He et al. “A Testbed for Evaluation of the Effects of Multipath on Performance of TOA-Based Indoor Geolocation”. In: *IEEE Transactions on Instrumentation and Measurement* 62.8 (2013), pp. 2237–2247. ISSN: 0018-9456. DOI: 10.1109/TIM.2013.2255976.

[19] J. Hightower and G. Borriello. “Location systems for ubiquitous computing”. In: *IEEE Computer* 34.8 (2001), pp. 57–66. ISSN: 0018-9162. DOI: 10.1109/2.940014.

[20] I. Guvenc and C. C. Chong. “A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques”. In: *IEEE Communications Surveys Tutorials* 11.3 (2009), pp. 107–124. ISSN: 1553-877X. DOI: 10.1109/SURV.2009.090308.

[21] “IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”. In: *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)* (2012), pp. 1–2793. DOI: 10.1109/IEEESTD.2012.6178212.

[22] *“iperf tutorial”*. URL: {https://openmaniak.com/iperf.php}.

[23] J. Ash and L. Potter. “Sensor network localization via received signal strength measurements with directional antennas”. In: *Proceedings of the 2004 Allerton Conference on Communication, Control, and Computing*. 2004, pp. 1861–1870.

[24] K. Wendlandt and M. Berhig and P. Robertson. “Indoor localization with probability density functions based on Bluetooth”. In: *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*. Vol. 3. 2005, 2040–2044 Vol. 3. DOI: 10.1109/PIMRC.2005.1651798.

[25] K. Kaemarungsi and P. Krishnamurthy. “Modeling of indoor positioning systems based on location fingerprinting”. In: *IEEE INFOCOM 2004*. Vol. 2. 2004, 1012–1022 vol.2. DOI: 10.1109/INFCOM.2004.1356988.

[26] E.D Kaplan. *"Understanding GPS:Principles and Applications"*. Artech House, 1996, pp. 4477–4479.

[27] *"Kernel Density Estimators"*. URL: {`http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0405/MISHRA/kde.html`}.

[28] R. Khanduri, S. S. Rattan, and A. Uniyal. "Understanding the Features of IEEE 802.11g in High Data Rate Wireless LANs". In: *International Journal of Computer Applications* 64.8 (2013), pp. 1–5.

[29] Abhishek Arunkumar Kulkarni. "Self-localization of target nodes using opportunistic communication with reference nodes, statistical time-of-arrival, grid method, linear matrix inequality and center-of-gravity". MA thesis. Auburn University, 2016.

[30] L. Cong and W. Zhuang. "Hybrid TDOA/AOA mobile user location for wideband CDMA cellular systems". In: *IEEE Transactions on Wireless Communications* 1.3 (2002), pp. 439–447. ISSN: 1536-1276.

[31] L. M. Ni and Yunhao Liu and Yiu Cho Lau and A. P. Patil. "LANDMARC: indoor location sensing using active RFID". In: *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).* 2003, pp. 407–415. DOI: `10.1109/PERCOM.2003.1192765`.

[32] L. Xinrong and K. Pahlavan and M. Latva-aho and M. Ylianttila. "Comparison of indoor geolocation methods in DSSS and OFDM wireless LAN systems". In: *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No.00CH37152).* Vol. 6. 2000, 3015–3020 vol.6.

[33] L. Yip and K. Comanor and J.C Chen and R.E Hudson and K. Yao and L. Vandenberghe. "Array Processing for Target DOA, Localization, and Classification Based on AML and SVM Algorithms in Sensor Networks". In: *Proceedings of the 2Nd International Conference on Information Processing in Sensor Networks*. IPSN'03. Palo Alto, CA, USA: Springer-Verlag, 2003, pp. 269–284.

[34] X. Li et al. "Indoor positioning within a single camera and 3D maps". In: *2010 Ubiquitous Positioning Indoor Navigation and Location Based Service*. 2010, pp. 1–9. DOI: 10.1109/UPINLBS.2010.5653577.

[35] H. Liu et al. "Survey of Wireless Indoor Positioning Techniques and Systems". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007), pp. 1067–1080. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.905750.

[36] L. Mainetti, L. Patrono, and I. Sergi. "A Survey on Indoor positioning systems". In: *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. 2014, pp. 111–120. DOI: 10.1109/SOFTCOM.2014.7039067.

[37] K. Pahlavan et al. "Taking Positioning Indoors: Wi-Fi Localization and GNSS". In: *Inside GNSS* 5.3 (2010).

[38] S. Zhou and J. K. Pollard. "Position measurement using Bluetooth". In: *IEEE Transactions on Consumer Electronics* 52.2 (2006), pp. 555–558. ISSN: 0098-3063. DOI: 10.1109/TCE.2006.1649679.

[39] S. Saha et al. "Location determination of a mobile device using IEEE 802.11b access point signals". In: *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003*. Vol. 3. 2003, 1987–1992 vol.3. DOI: 10.1109/WCNC.2003.1200692.

[40] David Schneider. *"New Indoor Navigation Technologies Work Where GPS Can't"*. URL: {http://spectrum.ieee.org/telecom/wireless/new-indoor-navigation-technologies-work-where-gps-cant}.

[41] S. Se, D. Lowe, and J. Little. "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks." In: *I. J. Robotic Res.* 21.8 (2002), pp. 735–760.

[42] Song Gao. "toad Architecture". This author is a part of our research group at Auburn University.

[43] Stephen D. Strowes. "Passively Measuring TCP Round-trip Times". In: *Commun. ACM* 56.10 (Oct. 2013), pp. 57–64. ISSN: 0001-0782. DOI: 10.1145/2507771.2507781. URL: http://doi.acm.org/10.1145/2507771.2507781.

[44] Mark Sullivan. *"A Brief History of GPS"*. URL: {http://www.pcworld.com/article/2000276/a-brief-history-of-gps.html}.

[45] R. D. Tingley and K. Pahlavan. "Space-time measurement of Indoor Radio Propagation". In: *IEEE Transactions on Instrumentation and Measurement* 50.1 (2001), pp. 22–31. ISSN: 0018-9456. DOI: 10.1109/19.903874.

[46] F. Tramarin et al. "Performance assessment of an IEEE 802.11-based protocol for real-time communication in agriculture". In: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. 2014, pp. 1–6. DOI: 10.1109/ETFA.2014.7005304.

[47] V. Otsason and A. Varshavsky and A. LaMarca and E. de Lara. "Accurate GSM Indoor Localization". In: *Proceedings of the 7th International Conference on Ubiquitous Computing*. UbiComp'05. Tokyo, Japan: Springer-Verlag, 2005, pp. 141–158. ISBN: 978-3-540-28760-5.

[48] Jeremy G. Vanantwerp and Richard D. Braatz. "A tutorial on linear and bilinear matrix inequalities". In: *Journal of Process Control* 10 (2000), pp. 363–385.

[49] Wikipedia. *"Berkeley Software Distribution"*. URL: {https://en.wikipedia.org/wiki/Berkeley_Software_Distribution}.

[50] Wikipedia. *"IEEE 802.11g-2003"*. URL: {https://en.wikipedia.org/wiki/IEEE_802.11g-2003}.

[51] Wikipedia. *"iperf"*. URL: {https://en.wikipedia.org/wiki/Iperf}.

[52] Wikipedia. *"Location Based Services"*. URL: {https://en.wikipedia.org/wiki/Location-based_service}.

[53] Wikipedia. *"Monitor Mode"*. URL: {https://en.wikipedia.org/wiki/Monitor_mode}.

[54] Wikipedia. *"Statistical Distance"*. URL: {https://en.wikipedia.org/wiki/Statistical_distance}.

[55] Wikipedia. *"Wireless Ad Hoc Network"*. URL: {https://en.wikipedia.org/wiki/Wireless_ad_hoc_network}.

[56] Wikipedia. *"Wireless Sensor Network"*. URL: {https://en.wikipedia.org/wiki/Wireless_sensor_network}.

[57] Wikipedia. *"Wireshark"*. URL: {https://en.wikipedia.org/wiki/Wireshark}.

[58] Wikipedia. *"Worcester Cold Storage and Warehouse Co. fire"*. URL: {https://en.wikipedia.org/wiki/Worcester_Cold_Storage_and_Warehouse_Co._fire}.

[59] X. Wang and H. Zhou and S. Mao and S. Pandey and P. Agrawal and D. M. Bevly. "Mobility improves LMI-based cooperative indoor localization". In: *2015 IEEE Wireless Communications and Networking Conference (WCNC)*. 2015, pp. 2215–2220. DOI: 10.1109/WCNC.2015.7127811.

[60] X. Wang and L. Gao and S. Mao. "BiLoc: Bi-Modal Deep Learning for Indoor Localization With Commodity 5GHz WiFi". In: *IEEE Access* 5 (2017), pp. 4209–4220. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2688362.

[61] X. Wang and L. Gao and S. Mao and S. Pandey. "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach". In: *IEEE Transactions on Vehicular Technology* 66.1 (2017), pp. 763–776. ISSN: 0018-9545. DOI: 10.1109/TVT.2016.2545523.

[62] X. Wang and L. Gao and S. Mao and S. Pandey. "DeepFi: Deep learning for indoor fingerprinting using channel state information". In: *2015 IEEE Wireless Communications and Networking Conference (WCNC)*. 2015, pp. 1666–1671. DOI: 10.1109/WCNC.2015.7127718.

[63] Xuyu Wang and Shiwen Mao and Santosh Pandey and Prathima Agrawal. “CA2T: Cooperative Antenna Arrays Technique for Pinpoint Indoor Localization”. In: *Procedia Computer Science* 34 (2014), pp. 392 –399. ISSN: 1877-0509. DOI: `http://dx.doi.org/10.1016/j.procs.2014.07.044`.

[64] Ting Yang, Qing Yang, and A. Lim. “Driver layer approach to time-of-arrival ranging in IEEE 802.11g networks”. In: *2012 IEEE Consumer Communications and Networking Conference (CCNC)*. 2012, pp. 240–244. DOI: `10.1109/CCNC.2012.6181094`.

[65] Z. Gu and E. Gunawan. “Radiolocation in CDMA Cellular System Based on Joint Angle and Delay Estimation”. In: *Wireless Personal Communications* 23.3 (2002), pp. 297–309. ISSN: 1572-834X. DOI: `10.1023/A:1021283401901`.

[66] Z. Xiang and S. Song and J. Chen and H. Wang and J. Huang and X. Gao. “A Wireless LAN-based indoor positioning technology”. In: *IBM Journal of Research and Development* 48.5.6 (2004), pp. 617–626. ISSN: 0018-8646. DOI: `10.1147/rd.485.0617`.