An NDN Based Indoor Positioning and Navigation System

by

Sifat Ut Taki
16101064
Raiyan Hossain
16101331
Sadia Anwar
17201041

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
December 2019

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Sifat Ut Taki
16101064

Raiyan Hossain
16101331

Sadia Anwar
17201041

Approval

The thesis titled "An NDN Based Indoor Positioning and Navigation System" submitted by

1. Sifat Ut Taki (16101064)

2. Raiyan Hossain (16101331)

3. Sadia Anwar (17201041)

Of Fall, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on December 10, 2019.

Examining Committee:

Supervisor:
(Member)

Dr. Amitabha Chakrabarty
Associate Professor
Department
Institution

Head of Department:
(Chair)

Dr. Mahbubul Alam Majumder
Professor and Chairperson, Interim Dean, School of Sciences
Department of Computer Science and Engineering
Brac University

Abstract

Providing navigation services indoor is becoming a growing trend in recent years. Traditional GPS is not a viable solution for indoor use as it demands more accuracy and may be out of reach of GPS signals. Indoor positioning can be implmented using a variety of localized techniques that are not reliant on GPS. Among all the techniques, Wi-Fi RSSI fingerprinting based indoor positioning is very popular for its accuracy, reliability and ability to use the existing Wi-Fi infrastructure of a building. Large areas like shopping complexes, airports, universities, etc. already have an existing Wi-Fi network deployed. Making use of an existing network makes this technique cost effective and easy to implement. However, a large indoor area with multiple floors will require a large database of Wi-Fi fingerprint data. The performance of the localization and navigation algorithms suffer greatly due to this large database, both due to time requirements and computational complexity. This paper focuses on improving the performance of Wi-Fi RSSI fingerprinting based indoor positioning and navigation systems in large areas by making use of named data networking as the network architecture. We aim to demonstrate how named data networking can help improve the performance of both indoor localization and indoor navigation in various scenarios over traditional TCP/IP based solutions.

Keywords: Indoor Positioning; Indoor Localization; Indoor Navigation; Named Data Networking; RSSI Fingerprinting;

Acknowledgement

We would like to start off with a big and humble thank you to Dr. Amitabha Chakrabarty, our supervisor, who has constantly aided us in our efforts and has been by our side through the thick and thins of this work and continuously encouraged us to complete our work in time. We are grateful to have been able to work under his supervision. Our gratitude goes to BRAC University, and the faculty members of the Department of Computer Science and Engineering, from whom we gained the very knowledge and experience we depended on to complete our thesis. Finally, We would like to extend our gratitude to the Great Almighty. Without his grace we could not have acquired the strength, determination and intelligence required to complete our work.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AOA	Angle of Arrival
AP	Access Point
BLE	Bluetooth Low Energy
BS	Base Station
CS	Content Store
FIB	Forwarding Information Base
ICN	Information Centric Network
IPS	Indoor Positioning System
KNN	K-Nearst Neighbors
LBS	Location Based Service
$LFSR$	Linear Feedback Shift Registers
MA	Metereological Administration
$MEMS$	Micro Elctro-mechanical System
PIT	Pending Interest Table
RF	Radio Frequency
RP	Reference Point
TOA	Time of Arrival
$WKNN$	Weighted K-Nearst Neighbors

Chapter 1

Introduction

Global Positioning System (GPS) is making our life simpler. It helps us to locate our position and navigate around the city[1]. For someone unfamiliar with a place or city, Global Positioning System (GPS) is a life saver. Now a days, lots of apps are taking advantage of GPS to locate and navigate user from one place to another. However, there is a limitation of GPS. It does not work inside buildings[2]. The signal from the satellites are not strong enough to be able to accurately measure position of a person inside a building. Making matters worse, there are multi-storey buildings and it is not possible to locate which floor a person is on using GPS since smartphones have less powerful GPS receiver that cannot accurately measure altitude[3]. In modern era there are large numbers of complexes like shopping malls, airports, hospitals etc. where people can get lost easily and need to locate themselves inside that complex. Problems like finding a specific shop in a shopping mall or locating an ATM booth or a toilet, identifying the correct terminal in an airport, locating the exact ward in a large hospital complex are very common now. To tackle the situation, many have developed various indoor positioning techniques as a solution. Indoor positioning technologies come in various forms. The majority of them rely on a variety of electromagnetic signals or otherwise rely on specialized hardware. Specialized hardware are not preferable as they are very costly and are not appealing enough to consider viable in most cases. The other route is to rely on existing technologies such as Wi-Fi or Bluetooth which are readily available. However these pose a problem that their behavior is unpredictable. Thus the stopgap solution has become to maintain a database of readings from these wireless signal sources to cross reference with. However this solution also has a large drawback in the form of a large database. Especially in a real time scenario like indoor positioning with many users and where constant updates are required, continuously relying on a large database becomes a handicap in terms of computing requirements and performance. We aim to remedy this performance problem using Named Data Networking, a radical new network architecture. NDN's reliance on caching data locally on routers and it's data centric nature may be the perfect solution to eliminating indoor positioning's drawbacks and make it a cost effective and viable solution using otherwise existing technologies. It is worth noting that NDN itself relies on specialized hardware. However NDN is gaining popularity and it can be expected that hardware with default NDN capabilities will eventually become the norm going forward and will not be considered specialized hardware any more.

1.1 Indoor positioning

Indoor positioning systems are primarily based on WIFI, Bluetooth, Infrared, Ultrasonic sound, image processing etc[4]. The most promising of all of them is using prerecorded RSSI fingerprints from Wi-Fi routers and use machine learning algorithms to determine a person's location. This localization technique works exceptionally well at a lower cost since the existing Wi-Fi infrastructure of a building can be used[5]. However, this technique requires the system to manage a database of prerecorded RSSI fingerprints which is used by the algorithm to successfully determine the position. For small indoor areas the algorithms can perform fast enough but for larger areas with a massive fingerprint database, the performance of the algorithms can be very slow[6].

Navigation inside a building can be another challenge. Once the position of the user inside a building is successfully determined, it is important to navigate him to his desired location. Navigation requires running resource intensive algorithms that may take considerable amount of time based on the area of interest and computation power[7].

Both localization and navigation in an indoor environment is challenging. The problems become more evident as the indoor area gets larger. It is possible that there can be situations where the server is calculating same position or same path to a destination multiple times because multiple clients are requesting. For example, an exhibition is taking place at a building at a particular time. Visitors will enter the building and request for navigation to the exhibition room. In this case, most of the localization requests will be from the building's entrance and most of the navigation requests will be for the exhibition room. Handling the similar request over and over again is resource intensive for the server. We tried to mitigate this problem by using NDN, a new internet architecture.

1.2 Objective

- Establishing the viability of Named Data Networking (NDN) as an alternate to traditional TCP/IP based network architecture.
- Demonstrating the performance benefits of NDN over alternate established technologies.
- Applying NDN to fields where it can take the most advantage of it's working principles.
- Taking advantage of NDN to deliver large amounts of data with minimal latency.
- Demonstrating NDN's ability to vastly improve IOT technologies by applying it to Indoor Positioning Systems.
- Distributing data across various endpoints instead of a centralized place for quick access to aforementioned data.
- Improving latency and access performance for end-user devices by having frequently accessed data ready for delivery.

- Establishing Indoor Positioning as an accessible solution by basing it on pre-existing hardware and technologies.

- Implement the system in a way that is viable to end users and businesses.

1.3 Motivation

Throughout the Thesis, our primary goal has been to contribute to solving the ever growing issue of handling large amounts of data and the latency that comes along as a result of transporting said data across networks and devices. This problem is most apparent in fields like Internet of Things and Networking where large amounts of data need to be accessed frequently and repeatedly. We wanted to rely on new and promising technologies to achieve this goal. One area we found that would truly benefit from our implementation was Indoor Positioning Systems. As almost all indoor positioning technologies rely on a large database of fingerprint data and is a technology that requires real-time data, it's a prime candidate for our implementation. Mega structures like large shopping malls, Airports, Galleries and Halls are increasingly adopting indoor positioning and our implementation can lessen the pressure of increasing volumes of data and the delay that comes along with it.

1.4 Summary of contribution

We are hopeful and confident that our work will be a fruitful contribution to both indoor positioning and future networking implementations. We believe our Thesis will contribute to the following fields.

- Showcase a practical implementation of Named Data Networking and it's uses.

- Help establish NDN as a viable alternate to the aging TCP/IP network architecture.

- Demonstrate NDN's performance benefits over TCP/IP and establish it as a better solution for modern data centric networking needs.

- Improve performance of indoor positioning and navigation especially in real time use cases.

- Make indoor positioning and navigation technology more accessible from an implementation complexity and resource requirement perspective.

1.5 Thesis outline

- Chapter One is our introduction. It introduces our topic, our objectives and why we were motivated to pursue this topic.

- Chapter Two is our literature review. It briefly analyzes all the fields relevant to our work and existing work in these fields.

- Chapter Three contains details of our proposed work plan and planned implementation details.

- Chapter Four is a summary of our implementation, testing methodology and recorded results.
- Chapter Five contains some final words along with current limitations and scopes for improvement.

Chapter 2

Literature Review

Indoor Positioning technologies cannot rely on traditional Navigation Systems such as Global Positioning Systems due to being indoor. GPS does not function reliably indoors due to multiple constraints[8]. The L1 C/A standard is transmitted by all satellites and is considered the legacy mode of operation for GPS as it is freely available for public use.
According to IS-GPS-200 specification, the signal must arrive on the ground at a minimum strength of -158.5 dBW (i.e.-128.5dBm). Even the most conservative models suggest that GPS signals suffer from attenuation of up-to 2.9 dB for every meter. Some experiments suggest that the attenuation of GPS signals through structures are around 1 dB per meter[9]. To be able to reliably track inside large structures, especially closed areas like cubicles or elevators would require a strong signal ranging from around -160 dBW to -200 dBW. This requirement poses a major challenge as typically the noise of a typical receiver will range around -130 dBW[10]. These problems generally invalidate GPS as a viable solution for navigating indoor locations reliably. To this end, the focus has turned to rely on IOT systems for indoor navigation. Many research and surveys have been conducted in this field[11], [12] that explore various methods of indoor positioning.

2.1 Indoor Positioning Technologies

The technologies used for Indoor Positioning can be classified using several criteria, including based on the signal used for location. We can have the following kinds of signals:

- Radio Frequency(RF): It is a general term that references the wide range of Radio band signals that are available for public use. These are popularly used for popular communication protocols such as WiFi and Bluetooth[13].

- Audio: A generalization for human audible or inaudible(infrasonic and ultrasonic) sounds.

- Light: Light is also electromagnetic in nature similar to radio signals. however they are still categorized as a separate technology as positioning using light sources works fundamentally in a different way as opposed to radio signals. Light, like sounds, can both be perceivable by the human eye or also invisible such as infrared.

- Magnetic Fields: Some methods using Earth's natural magnetic fields along with some reliant on artificially generated magnetic fields.

Another criteria for categorization is whether or not the signal is
1. Received and analyzed.
2. Transmitted.

- The first we can classify as a passive method as in this case the system or infrastructure is generating the information or signal and the client system, which may be a mobile device with the end user.
- The other category can be called active as the client device is the source of the signal

The final categorization is based on whether the signal itself contains any metadata or any information that can help with the location.

- embedded information is included in the signal, such as WiFi signals with access point information or bluetooth information.
- no embedded information available such as a raw light source or magnetic fields.

2.2 Signal Types

Here we shall focus and explore the aforementioned types of signals used in various indoor positioning systems.

2.2.1 Radio Frequencies

- WLAN, popularly known as Wi-Fi is widely used and widely available as the most common electromagnetic signal that is available everywhere. Wi-Fi relies on electromagnetic waves to transmit data to clients. This falls under the passive category with embedded information.

  (i) If the propagation method of the antenna or device is know it can be used to measure the distance from the known base[14].

  (ii) Multiple Access Points can be used to cross reference values and solve the values using a multilateral method[15].

  (iii) Fingerprinting is a very popular method to record pre-existing values and then use those values to match new clients. This method is widely popular as it sidesteps all of Wi-Fi's irregularities and non linear strength changes.

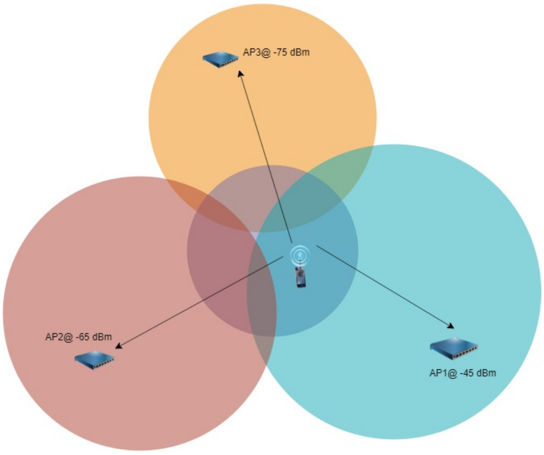


Figure 2.1: WiFi Triangulation

- Bluetooth is another readily available wireless signal source. Recently the emergence of Bluetooth Low Energy (BLE) has made the technology much more viable[16] as devices can afford to leave bluetooth actively running. Most of the methods used in WiFi fingerprinting can be applied to bluetooth as well[17]. As BLE can run on batteries for years on end, they are viable for Triangulation or Received Signal Strength (RSS) fingerprinting. One example project[18] claims high accuracy of 2 meters within a small enclosed area but is susceptible to attenuation. A more commercial example is Apple's iBeacons. The main drawback of bluetooth based systems is that bluetooth is usually used for client devices and are rarely statically positioned. As such, dedicated hardware for positioning is required.

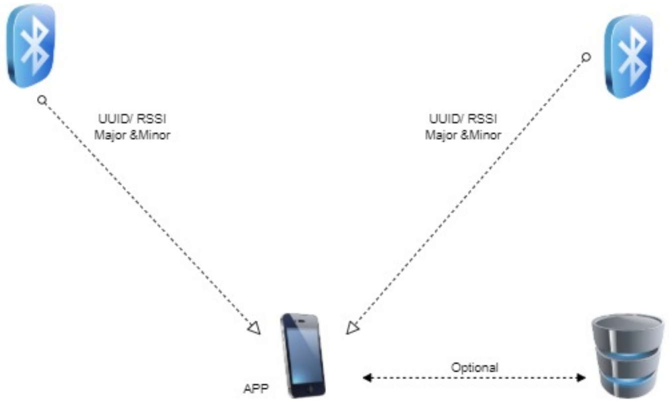


Figure 2.2: Bluetooth Beacon Based Positioning

2.2.2 Light

This section focuses on Light based systems such as passive visible environmental lights or actively generated lights for the exclusive purpose of navigation.

In most cases passive light sources are used instead of explicitly generating light sources. All places have general luminosity that varies over distances. So this technology can be applied to any place as required. Afterwards the luminosity can be recorded in a straight trajectory and then on a rerun the position of the client

can be accurately estimated. Randall et al's 'LuxTrace' [19] developed a system that claims an accuracy of up-to 21cm which is quite impressive. However, it is limited trajectories which renders it less viable.
Another approach is to use specific light sources specialized for positioning. These may be regular lights or lights that have some embedded information such as pulses. This approach is similar to RSS based methods as it estimates distances to certain sources. However this also relies on specialized hardware or infrastructure.

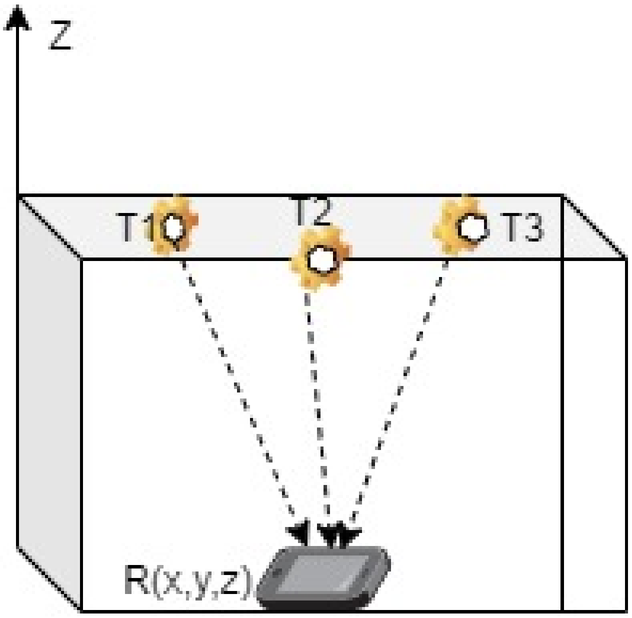


Figure 2.3: Light Source Based Positioning

Other than light sources, visual information can also be processed using cameras and other hardware. This generally falls under computer vision based systems. Cameras can either rely on specific pieces of information or just the general environment. These include Visual Odometry, which is the process of starting with an initial location and estimates the users location based on the motion of environmental objects. This process has improved but still isn't fully reliable.[20] Another way is to use specific symbols or objects as reference points, such as Signpost[21] which continuously scans bar codes set up in the environments using a phone's camera. This can provide good accuracy but it requires the codes to be within a phone's field of vision.

2.2.3 Audio

Sound can also be used, both encoded and non-encoded for positioning system. Using a human audible sound specifically generated for indoor positioning is of course not really a viable solution as this would annoy people and disrupt the environment. However, audio may be disguised as subtle variations of existing sound such as music. One method implements this through digital watermarking by playing the same music from different speakers with modulated amplitudes[22].
Passive sound based technologies on the other hand rely on sounds that are already available in the environment. this usually involves recording sounds in the environment and matching it against a database of known places to recognize a location.

As per the description, this is only viable for general locations and is not accurate enough for navigation.

2.2.4 Magnetic Fields

While some approaches using simply magnetic fields exist, most rely directly on the Earth's natural magnetic field strength and orientation for location. As magnetic fields are also subject to variation and complex principles they usually rely on fingerprinting methods as well[23].

2.2.5 Hybrids

In reality, Most implementations would benefit from using a hybrid of multiple technologies as they all have drawbacks separately. Some use both bluetooth and WiFi[24] which reports an error rate of 0.77meters, while others rely on a combination of ultrasonic sensors and compasses[25].

2.3 WiFi Access Point Based Positioning

Among all the available options of indoor positioning technologies, the one that seems to draw the most attention seems to be WiFi signal based fingerprinting. The reason is fairly obvious. WiFi Access points are almost universally accessible everywhere and there is usually enough available points to apply positioning methodologies. These methodologies can be summarised as follows.

Technology	Pros	Cons
Angle of Arrival	Moderate accuracy	Requires oriented antennas
Time of Arrival	Moderate accuracy	Base station must be synchronized
Time Difference of Arrival	Moderate performance	Low accuracy
Triangulation and Trilateration	Simple and easy to use	Requires angle
Location Fingerprinting	High accuracy	High calibration time

Table 2.1: Indoor Positioning Systems comparison

RSSI Based Positioning

In order to estimate a position, triangulation uses a triangle's geometric properties. The lateration and angulation are the two parts of triangulation. Lateration refers to the estimation of a place based on the calculation of distances from several reference points[11]. Angulation locates an entity in relation to several reference points by calculating angles. In order to identify the crossing point that determines the approximate location of the entity / individual, this strategy utilizes the arrival angle of at least 2 reference points and its accompanying rows.

Scene Analysis

This form of placement incorporates the recorded scenario interpretation framework. In some cases, the scene can vary depending on its use. For example, it is possible to differentiate between locations using any signal type, from a sensor or from the environment itself. The location can be calculated using this approach based on the similarities between scenes[26]. Scene analysis is a principle that effectively uses fingerprinting. Fingerprint is the sole characteristic of an image, or a series of features. It compares the collected features to the features already stored in the database. There are two steps to this approach. They are:

- Training Phase
- Tracking Phase

The signal intensity from APs is obtained at fixed positions, called reference points (RPs), in the training phase. This process builds the fingerprint registry to be used in the monitoring phase. Since the location of mobile users is calculated in the repository from this RPs, they should be distributed equally and homogeneously. In the monitoring process mobile users use probabilistic and deterministic algorithms to equate the RSS information for the nearby APs with RPs data set to locate a corresponding location[27].

Time Of Arrival Based Positioning

The propagation time of the signal from the target to the reference point is proportional to the length between them. In order to use the TOA technique, at least three reference points must be found in the 2D plane to triangulate. The system's downside is to sync the transmitters[27], [28] and have to mark a time stamp for a signal transmission to determine how long it has been moving.

Time Difference Of Arrival Based Positioning

By using the time difference the signal reaches multiple receptors, TDOA determines the relative position of the transmitter. Two TDOAs are indicated by three receiving units and the point of crossing defines the target location. This method also suffers from the problem of multipath.

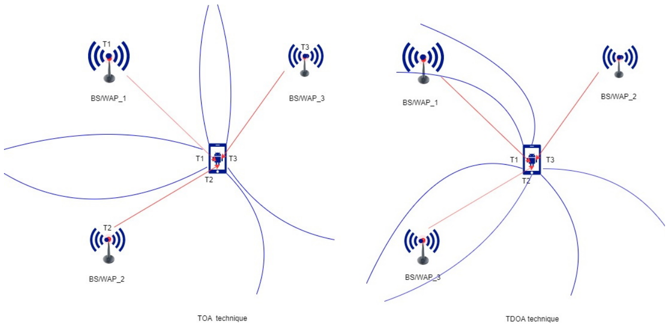


Figure 2.4: TOA vs TDOA

AOA Angle of Arrival

The intersection between the circular distances created by a base station is used to map the target point in the plane by AOA. When two or more Base Stations / AP are defined for the smartphone's AOA of the received signals and the distance from two BS / APs are known, the location of a target can then be calculated. Nevertheless, AOA suffers from reflection problems of shadows and multipath, and requires expensive and sophisticated equipment[27].

Triangulation and Trilateration

Trilateration relies on the measured distance from the device to access points or base stations. With the help of at least three access points, the location of the device may be pinpointed. Triangulation on the other hand relies on angles. It works by forming a baseline between two observation points. and uses the angles to the target object to measure it's distance from the baseline by forming triangle, as seen in the image below. However this would require sophisticated hardware if used with Access Points.

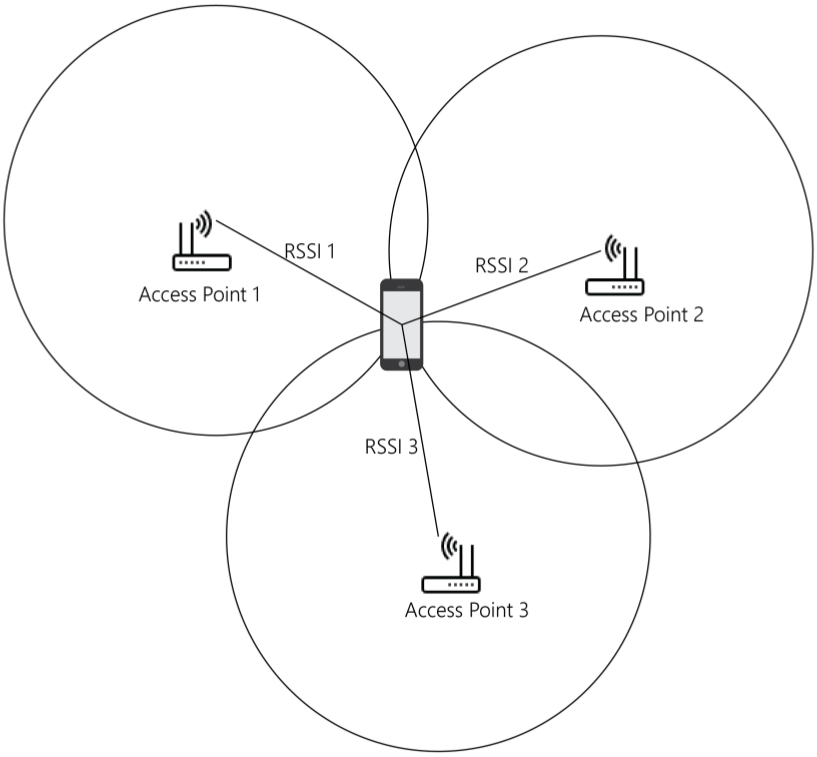


Figure 2.5: Trilateration Based Locating

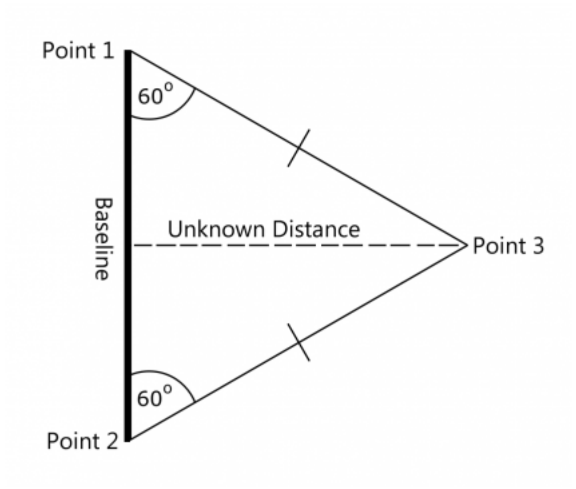


Figure 2.6: Triangulation Based Locating

RSSI Fingerprinting

A number of indoor location systems use the Received Signal Strength Indicator (RSSI), which is characterized by attenuation during propagation. While RSSI can provide accuracy of location in simple environments at a meter level, multipath dynamics and complicated environment causes sudden performance degradation. Much of the reliable RSSI positioning systems currently in use remain at room level of precision[29]. Two specific methods for calculating the goal distance are used for the signal strength obtained or approach dependent on RSS. One metric is pseudo-range dependent on the established empirical relationship of radio propagation. It uses trilateration to find smartphone locations between smartphones and multiple base stations or access points from its estimated pseudo-ranges. The second method is RSS-Fingerprinting, based on pre-stored access point RSS values in the registry. Offline and electronic approaches for calculating the position of a goal are carried out here. In the offline process, a radio map is generated to establish location, throughout contrast to current online information.

2.4 Floor Detection

Along with detecting location on a two dimensional plane, the plane's vertical altitude is also important as well. Unlike outdoor GPS, indoor positions are frequently not on the same level. This results in a need to detect the user's altitude as well. Unfortunately, the techniques used for locating on a two dimensional plane falls flat when trying to detect altitude through floors and ceilings. Thus, dedicated methods are required to detect floors for indoor positioning technologies. These technologies can be categorized into two groups.

- Explicit: When the floor or altitude is input manually
- Implicit: When it is detected

Explicit methods usually involve the user manually inputting the floor, or scanning a QR Code to detect the floor. Due to their hands on nature and difficulty of use, these methods are not evr popular.

Implicit methods involve automatically detecting the user's current floor. This method mostly consists of using a barometer sensor to use atmospheric pressure to detect altitude. Traditionally, barometers are used outdoors to measure altitude and meet positioning needs. Most modern day smartphones come with microelectromechanical system (MEMS) pressure sensors which can measure pressure with relatively high accuracy. It is, in fact, accurate enough to even detect human movement[30]. In this study, accuracy was 0.53 m. Because of the sensitivity of barometric sensors to vertical movements, a number of researchers[30], [31] have used these sensors to detect vertical motions of a human body, such as standing up, sitting down, and falling. As MEMS technology has gotten cheaper and smaller, it is now viable to embed them in common portable devices such as smartphones. Recent studies[32], [33] have proven that barometer sensors are suitable for use in harmony with other sensors and systems. In Sabatini's work[34], a sensor fusion method is presented to track vertical velocity and height based on inertial and barometric altimeter measurements.

Use in floor detection

Atmospheric pressure has been well known to decrease as the altitude increases. This effect can be represented as a formula[35]. By using a barometric sensor to measure air-pressure changes, we can calculate the altitude changes corresponding to pressure. To estimate the altitude of an object relative to a reference point, we can use atmospheric pressure's natural behavior:

$$p = p_0.exp\left(-\frac{g.M.h}{R.T_0}\right) \tag{2.1}$$

p is the pressure at altitude h, p_0 is the pressure at a reference point, M is the molar mass of dry air, g is the gravitational field strength, R is the gas constant of air, and T_0 is the temperature. Air pressure and the amount of water in air change with time; however, humidity does not significantly affect the accuracy of the system in indoor altitude estimation. Therefore, gas constant R for dry air is sufficient and so is the molar mass M of dry air. When the altitude changes slightly, we usually consider the gravitational acceleration g as constant. Another form is expressed as follows:

$$h = -\frac{R.T_0}{g.M}.ln\left(\frac{p}{p_0}\right) \tag{2.2}$$

These can be applied to detect floors assuming that the height of the floors are known. If that is not possible an alternate method is to deploy sensors on each floor individually.

2.5 Navigation

As Location Based Services (LBS) have gained popularity over the last decades[36]–[38] Navigation services along with locating services have also seen great demand. While GPS systems for outdoor navigation are fairly common and part of our daily lives, Navigating indoors and fine grained navigation have also seen a recent rise in popularity. The use cases for these types of navigation scenarios range from Airport, Hospital, Mall navigation to Vehicle Parking lot navigation and even self driving cars. Other applications also include contextual information based services such as advertising, information panels. These have gained popularity over the recent years due to advancements in Bluetooth beacons and Bluetooth Low Energy (BLE) devices, image processing based information and even augmented reality services.

2.5.1 Location

For location services, although there are many classifications from different perspectives a general categorization can be done between two groups. These groups are

- Radio Frequency (RF) based systems
- Other Systems

Radio frequency based techniques are many, which include WiFi Received Signal Strength Based location detection and all it's many variations, Bluetooth based services and many others.
Non-RF techniques include a large variety of techniques that which include Audio based systems which use audio signature fingerprinting, embedding information in existing sounds etc. It is also possible to to make use of Magnetic Fields such as Earth's magnetic fields and also artificial magnetic fields. Light, while still technically an electromagnetic wave, is not considered a radio frequency based technique due to it's completely different implementations and it can also be used in interesting ways.
The distinction between RF and Non-RF technologies are made because the RF techniques are dominant due to their general availability and their passive nature of function. Most of the Non-RF based technologies are dependent on specialized and dedicated hardware.

2.5.2 Path finding

While the previously mentioned methods can be used to locate someone while they are trying find their location on a map, or to update their location while they are on the move, they don't actually provide any means of actually guiding the the person towards their desired destination. This is where path finding or navigation algorithms come in. Before any path finding algorithm is used, the navigable region needs to be represented in a way suitable for the algorithm to operate on. A general way to do this is to assign nodes or points to the physical navigable region. Afterwards edges are used to connect the nodes and edges are assigned costs based on their distance or other constraints. This entire structure can be represented as a graph. This graph can then be traversed by path finding algorithms to find the cheapest (shortest) path. One of these algorithms is Dijkstras algorithm[39]. It is a greedy algorithm that is used to solve shortest path problems. It is still frequently used dew to it's robustness and simplicity. Other algorithms include Bellman-Ford and Floyd-Warshall. Another popular algorithm is the A-star (also stylized as A*), which is a modified implementation of Dijkstra's implementation that relies on a heuristic.

2.5.3 Standard

Most attempts to provide a standard for indoor navigation systems have generally been commercially unsuccessful due to the nature of indoor spaces. Outdoor navigation is generally simpler as vehicles drive on streets and pedestrians stick to sidewalks. However in an indoor environment there are factors like Elevators, Travellators, Escalators and narrow pathways etc. Nevertheless, some attempts to consolidate a standard have been made. One of the few existing methods are IndoorGML, Specified by the Open Geospatial Consortium. However their method only describes indoor spaces and their relationships, but not architectural features. Due to these limitations standards have gained poor market reception. It is easy to see that there is difficulty in standardizing indoor spaces as there is a larger amount of freedom involved for the users.

2.6 Named Data Networking

In 1970s, when the internet was evolving, TCP/IP was introduced as a solution for point to point connectivity. Back then it worked exceptionally well and solved the underlying problem. It has been a long time since TCP/IP was introduced and the world has changed a lot. While TCP/IP is still robust and fulfilling our needs, it has some issues. Named Data Networking (NDN) is a new internet architecture that addresses some of the issues and it is designed to replace traditional TCP/IP architecture. Recently there are a lot of researches going on with NDN [40]–[42].

2.6.1 Architecture

Named Data Networking (NDN) is the future of internet architecture. Instead of using IP to find a destination in TCP/IP architecture, NDN uses names to find data. Since it is an Information Centric Network (ICN), NDN only cares about data. It does not care where the data is coming from as long as the data is authentic and signed by the producer of that data. There are two types of data packets in NDN (Figure 2.8). One is the interest packet and the other one is the data packet. Interest packet is generated to request a data from the producer. The name in the interest packet specifies which data the consumer is looking for. This packet passes from the consumer to the producer. The data packet contains the data in response to the interest packet along with other information. This packet passes from the producer to the consumer[43]. NDN packets are very different from TCP/IP (Figure 2.7).

Interest packet

As stated earlier, there are two types of data packets in NDN. The interest packet is generated by the consumer who requests a specific data. The interest packet has the following fields[44].

Content Name: Both types of packets carry a name, which uniquely identifies a piece of data that can be carried in one Data packet. To retrieve Data, a consumer puts the name of desired data into an Interest packet and sends it to the network. Routers use this name to forward the Interest towards the data producer(s), and the Data packet whose name matches the name in the Interest is returned to the

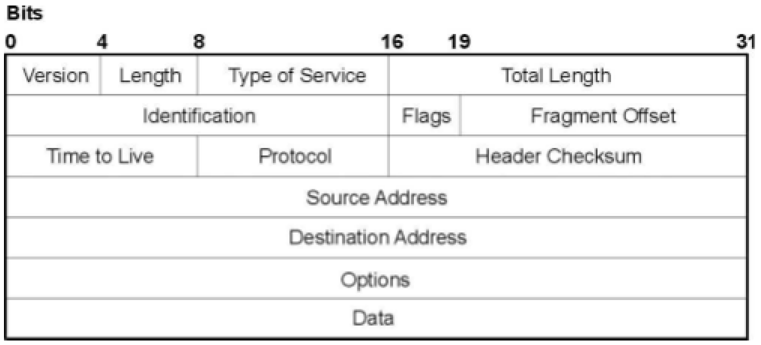


Figure 2.7: IPv4 packet

consumer.

Selector: Each Interest packet also carries a selector field which provides more specific descriptions of the desired Data

Nonce: nonce field which is a random number generated by the consumer.

Data packet

The other type of packet is the data packet. A Data packet carries the actual data, descriptions about the data, as well as a cryptographic signature that binds the data to the name This packet is generated by the producer of data in response to the request from the interest packet. This packet can come from the producer or from content store (CS). The following fields can be found in a data packet[44].

Content Name: Same name as the interest packet for which the data packet is generated.

Signature: This filed contains the key necessary to decrypt the data.

Signed Info: Contains the information about the signature like who signed the data and from where the key is retrieved from.

Data: It contains the actual data the consumer was looking for. It works as a pay load. Data can be anything like documents, image, video, audio etc.

2.6.2 Working principle

NDN is similar to TCP/IP network architecture. Instead of using IP to determine the specific destination to acquire data, it uses names to figure out what data the consumer is looking for. NDN only cares about the data. It does not care where it is coming from, as long as the data is authentic. This fundamental change leads to the shift of communication paradigm from location-centric to data-centric.

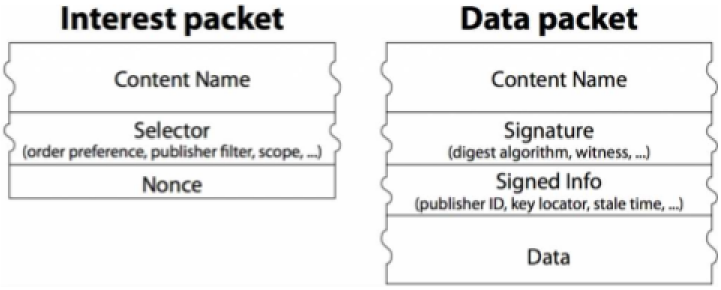

Interest packet
Content Name
Selector
(order preference, publisher filter, scope, ...)
Nonce
Data packet
Content Name
Signature
(digest algorithm, witness, ...)
Signed Info
(publisher ID, key locator, stale time, ...)
Data

Figure 2.8: NDN packets

NDN routers do not understand what the names actually mean. This allows applications to define their own naming structure and hierarchy. For example, someone wants to access a file from Brac University's file sharing system TSR. The name can be "/bracu/tsr/cse/ach/random.pdf" in NDN, where '/' is the delimiter separating the components of the name. From this hierarchy, we can see that the name is trying to access Brac University's TSR. Inside TSR, it is accessing Computer Science and Engineering department's instructor's file system. Finally, "random.pdf" is the file it is looking for. We can further specify the naming structure by adding segment and version. "/bracu/tsr/cse/ach/random.pdf/2/1" indicates that it is indeed looking for the first chunk of second version of the file "random.pdf". The applications can be designed to take advantage of this structure.
Another advantage is that it can be aggregated. the name "/bracu/tsr/" can be distribute among all the department's file system the same way IP prefixes are distributed for routing.
Three major data structures are maintained to implement the Interest and Data forwarding functions (Figure 2.9):

- Forwarding Information Base (FIB)
- Pending Interest Table (PIT)
- Content Store (CS)

Forwarding Information Base (FIB)

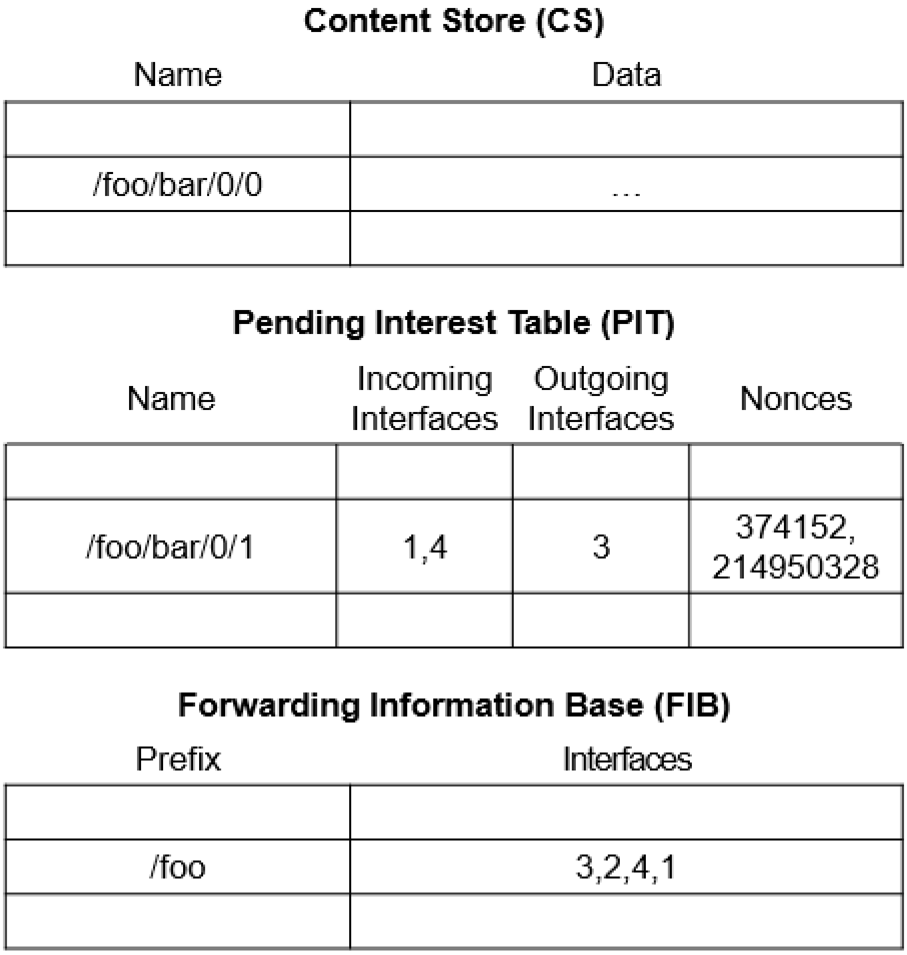
Forwarding Information Base (FIB) holds all the information about routing. The information is about the name prefixes and their next hop, similar to IP prefixes. The difference is that instead of IP addresses, we have names[45].

Pending Interest Table (PIT)

When a data packet is being retrieved, during that time all the incoming interest packets for the same data are kept in the PIT. once the data packet arrives, all the interests in the PIT are satisfied[45].

Content Store (CS)

Once an interest is satisfied, the data is cached inside the Content Store (CS) of the NDN router. This way it can satisfy the interest without reaching for the producer. This saves both time and resources[45].



Content Store (CS)

Name	Data
/foo/bar/0/0	...

Pending Interest Table (PIT)

Name	Incoming Interfaces	Outgoing Interfaces	Nonces
/foo/bar/0/1	1,4	3	374152, 214950328

Forwarding Information Base (FIB)

Prefix	Interfaces
/foo	3,2,4,1

Figure 2.9: NDN data structure

2.6.3 NDN router

The routers we use in TCP/IP internet architecture are designed to route packets based on destination IP addresses. There is a routing table inside the router containing all the routing information necessary for routing. When a packet arrives, the router just sends the packet to it's next hop based on the routing table.
An NDN router works similar to a traditional router. Once a data arrives, it first looks at its Content Store(CS) if there is ant entry with exact same name. If there is, it sends the data. If not, it will look at the PIT if there is already any awaiting request using the same name. If there is then the router will add the interest to the PIT as well and wait for the data. if there isn't any awaiting request in the PIT with the same name then it will send the interest packet to the next hop based on the information available in the FIB (Figure 2.10). this process is similar to TCP/IP routers. The difference is the routing is done based on name instead of IP address[45], [46].

2.6.4 Drawbacks of TCP/IP

Not optimized for content distribution

TCP/IP was designed to interconnect devices at two end point for conversation. However, today's internet is mostly used for content distribution. Most of the time it does not matter where the data is coming from as long as the data is authentic. That is why nowadays we are seeing more and more decentralized server system.

Weak security

Another problem with TCP/IP is the security. TCP/IP only secures the the connection between to end points. if one the points are compromised, this security method can no longer be effective. TCP/IP can not secure the actual data. Only securing the connection might not be enough for every situation. If the data is compromised, there is no way to detect in TCP/IP.

Not suitable for "unconventional" routing algorithm

TCP/IP routing can cause looping in certain routing algorithms. Routing algorithms that use greedy approach which means selecting the nodes closer to the destination. This type of algorithms may cause looping in TCP/IP. For example hyperbolic routing do not work on TCP/IP internet architecture.

2.6.5 Why NDN is better

Better architecture design

NDN not only aims to solve the issues with TCP/IP but also provide additional features suitable for modern internet. In TCP/IP internet architecture, every endpoint is defined by an IP address. Accessing that endpoint requires that specific IP address. NDN on the other hand defines data not endpoints. Every data has a specific name and is accessed by that name. To explain in simple words, TCP/IP asks who do you want to connect. Once connected, you can retrieve whatever data

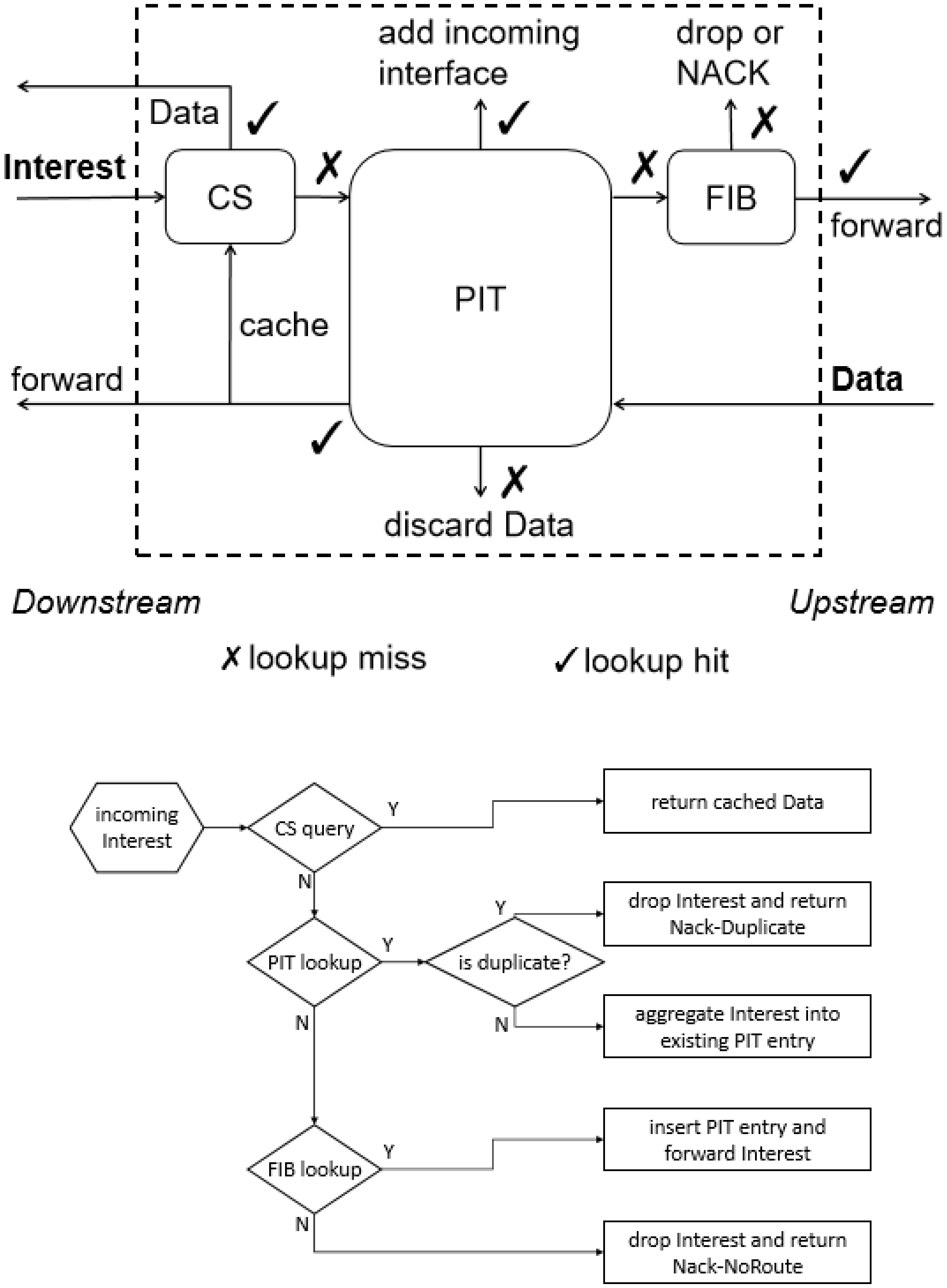


Figure 2.10: NDN working principle

you want. NDN asks which data you need. Once the data is determined, NDN will provide the data regardless of where it is coming from.

Better content handling

Today's internet is mostly about content distribution. NDN focuses on content rather than on connection. Providing the appropriate data to the one looking for is a faster approach than connecting to the data provider first then supply the data. It is because the data may have been already accessed by someone sitting closer than the actual data provider. Retrieving the data from the closer node is always faster than fetching them from the origin.

Better security

One of the major drawbacks of TCP/IP is its security. TCP/IP secures the connection as a security measure where NDN secures the data with encryption. If the endpoints security is compromised, securing the connection is not useful. By securing the the data, NDN provides better security than TCP/IP.

Works with most of the routing algorithms

Names in NDN are hierarchical and unique. NDN works with all the conventional routing algorithms like link state. Looping can be prevented as it allows free distribution to all the connected nodes as opposed to single-path forwarding constraint of traditional TCP/IP. As a result greedy algorithms like hyperbolic routing will also work with NDN.

2.6.6 Additional features of NDN

Along with all the improvements, NDN brings some new features to the table. These features are believed to help shape the standard for the modern internet.

New forwarding strategy

NDN routers has a different and intelligent forwarding strategy. When there are multiple requests for the same data, NDN router only sends the first request and wait for the data. Meanwhile all the same requests are kept in waiting. NDN router recognize same requests by looking at the name. Since NDN is designed to request data by using a unique name, same requests can be easily identified just by looking at the name it contains. Once the data arrives, NDN router satisfies all the requests waiting for the result. This way the server only had to produce the data once.

Content cache

NDN routers can cache the already requested data in the memory called Content Store(CS). By doing this, any future request for that data can be satisfied by the NDN router. The request will no longer need to reach the server. This can drastically reduce server load. It is also beneficial for tackling packet loss as the lost packet can be retrieved from the closest NDN router's cache unlike TCP/IP where the lost packet needs to be retrieved from the server again.

2.7 Our Approach

As WiFi RSSI based approaches are the most popular among all the aforementioned Indoor Positioning technologies, We have decided to use RSSI fingerprinting in our implementation. Since our aim is to make use of NDN to improve the performance of already existing technologies, we have decided to work with the RSSI Fingerprinting approach. As mentioned in the previous chapter, RSSI Fingerprinting relies on a vast pool of fingerprinting data to locate a user. It's benefits are numerous, such as it eliminates the need to model RSSI strength-distance relations by simply storing the readings for comparison. The biggest drawback is the frequent need to access this large database. We can reduce this latency relying on NDN. Our proposed implementation is as follows.

Technologies We Will Use

- For the network infrastructure we will use NDN. The fingerprint data will be stored on a MongoDB database. NDN will also be used to fetch floor detection and navigation route information.
- To determine location we will use the K-Nearest Neighbors classification algorithm. It will be used to take the users current readings and classify it as it's closest Reference Point (RP).
- For the navigation we will use the A* algorithm to navigate the user to their desired location.
- Detecting the current floor will rely on Barometer sensors of the users' mobile devices.

2.8 TCP-IP vs NDN in indoor positioning

TCP-IP

TCP-IP is the generally used approach in current network architectures. It transports data in series of bytes, and that is generally suitable for mass exchange of data. the IP header contains the bulk of the information required for routing and navigation. It contains the sender's IP address and the recipient's address as well. It also contains the packet's time to live and other information such as priority, recommends other conventions such as UDP, ICMP etc. The TCP section contains port information, checksum for integrity and other options. As a whole, TCP-IP provides a robust method of secure point to point communication. It's behavior is location centric as it focuses heavily on the source of the data rather than the content of the data.

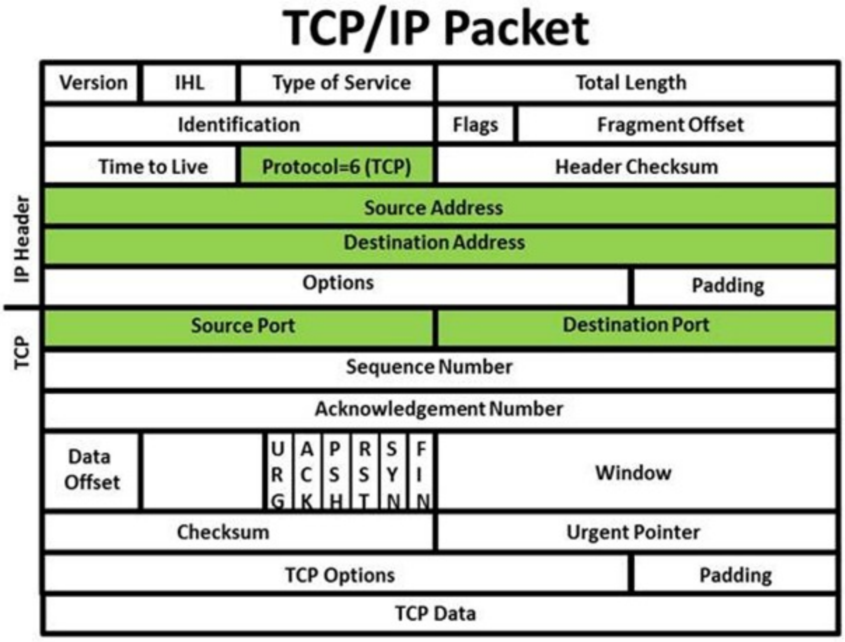


Figure 2.11: A TCP-IP Packet

Named Data Networking

NDN was developed to take advantage of the way data is distributed across the web. It's core ideology is to care about the content of the data instead of the source, and to deliver it by the most convenient means possible. It identifies data by their id or 'name', hence the name of the architecture. To store data elsewhere other than the primary source means some ability to store or cache data is required, along with a method to keep track of requests for that data, it's expiry etc. This role is performed by NDN routers. An NDN router contains a Pending Interest Table (PIT). Whenever a request (interest packet) is received, the router checks it's PIT to see if the same request already exists. If it does, it simply adds the client to the table, otherwise it adds the client to the table and forwards the interest pack. Later on, when the data is received, it forwards the data to everyone in the PIT. Additionally, it also caches the data locally so that if the same interest is received in the future, the router itself can satisfy the interest without even requesting the source.

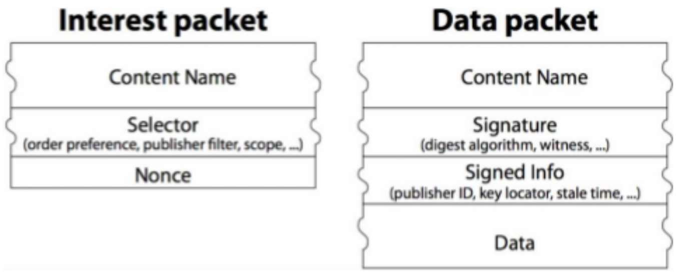

Interest packet
Content Name
Selector
(order preference, publisher filter, scope, ...)
Nonce
Data packet
Content Name
Signature
(digest algorithm, witness, ...)
Signed Info
(publisher ID, key locator, stale time, ...)
Data

Figure 2.12: NDN Packets

Usage in indoor positioning

As detailed before, RSSI Fingerprinting based methods rely on a database of fingerprint data. If we look at the properties of NDN and TCP-IP, we can see that NDN's benefits shoul theoretically have a very positive effect on the RSSI fingerprinting method. The database of reference points and the RSSI readings at those points can easily be cached on NDN routers. Additionaly other data such as the map layout, can be stored on the routers. Or if the calculations are done on the server side, the result data can be cached on the NDN routers. The benefits will gradually increase over time as more and more people use it. More results will get cached on the routers over time and very few requests will actually need to hit the servers.

Chapter 3

Localization and Navigation using NDN

3.1 Methodology

Based on our previous analysis, we have decided to apply NDN to RSSI Fingerprinting based localization to improve it's performance. The process will involve multiple steps. The steps involve,

- Detecting the current floor of the structure based on Barometric Pressure.
- Fetching the map of the current floor.
- Reading the current RSSI values of the user's current location.
- Using the prerecorded RSSI Fingerprint database with KNN to locate the user on the map.
- Picking a destination on the map and generate a route to it using the A* algorithm.

All the steps that require network communications are implemented using NDN. These steps include Floor Detection, Fetch Map, Self Location and Generating Route. They all stand to gain from the performance benefits of NDN. As we know, over NDN data may not need to be fetched from the server if the same data was requested before. This condition applies to all of the scenarios. For example,

Floor Detection

When a user requests to detect the current floor with their current barometer readings, these results will be stored locally on NDN routers' Content Stores (CS) and when later on another user tries to detect floor with the same readings the request will not need to travel all the way to the server.

Fetch Map

Fetching map will almost always find the required data in the CS. As there are only a few floors the same maps will be repeatedly requested frequently. As such most map fetches will be satisfied by the nearest router.

Self Location

Locating oneself involves sending the current RSSI readings from the routers selected for navigation and sending them to the servers for calculating the current location using Weighted KNN. But this is an expensive process as Weighted KNN requires time and resources to run. However, over time users will frequently request the same locations with the same readings, and requests will also be sent during navigation to update the users location on the map. Eventually the CS of the nearby NDN Routers will include most of the common RSSI readings. this will result in many of the Locating requests to be satisfied by the NDN Routers themselves rather than running the Weighted KNN algorithm on the server. This will greatly reduce the required time for self location.

Navigation

Navigation is done by specifying a destination and generating the shortest path to it from the current location using the A* algorithm. Similar to the aforementioned locating system, over time these navigation tasks will become repeated and the NDN routers will be able to satisfy the navigation requests and will not rely on the servers to run the A* algorithm.

As we can see, NDN can improve the performance and speed, along with the computing resource requirements of all the steps of the Locating and Navigation process. More detailed discussions of the various steps involved along with our implementation of NDN are discussed in further detail below.

3.2 Implementation

To implement this we chose to use BCS Computer City, a computer hardware accessories market in Dhaka, Bangladesh. First we went to the location and chose a floor for mapping.

Mapping

We first mapped a complete floor of the structure using Autocad. We also noted down locations of escalators, elevators and shop names throughout the floor.

Selecting access points

Afterwards we went around the map and selected access points to record values. These access points' RSSI values will later be used to record fingerprinting data.

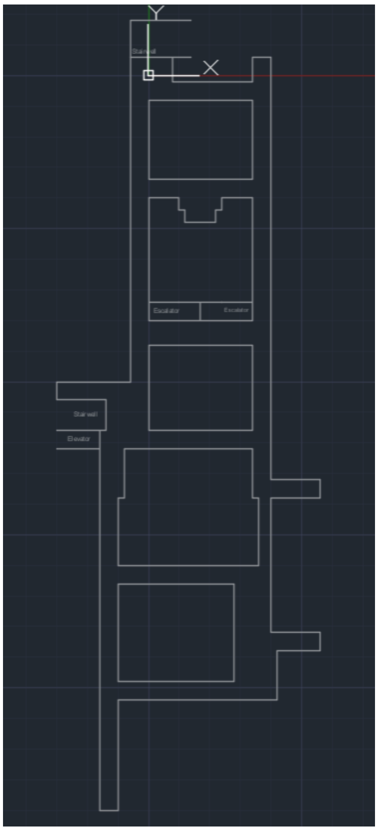


Figure 3.1: Map of BCS Computer City

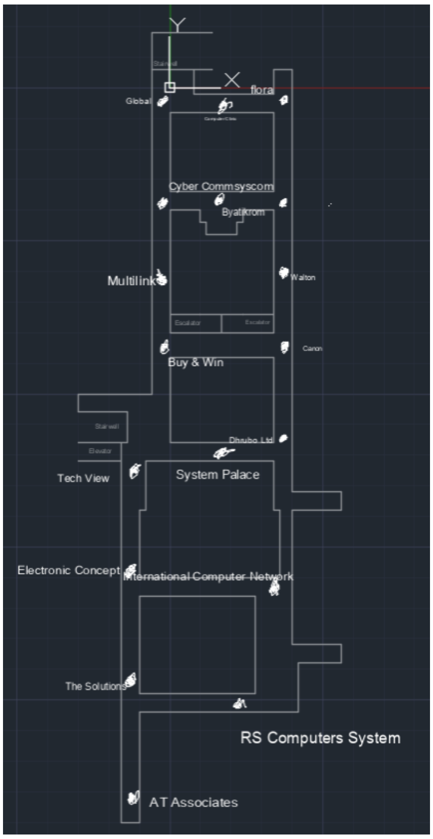


Figure 3.2: Access point selection

Recording fingerprint data

Afterwards we walked through the entire map recording fingerprint data at regular intervals. The RSSI values from the APs we picked previously were recorded and stored in the database.

Testing performance

As our goal is to compare Indoor positioning performance between TCP/IP and NDN, we needed to record performance data and compare them. After the database of fingerprint data was completed, we started testing performance using both TCP/IP and NDN and recorded both results for future comparison. Theoretically NDN caches data after it has been requested once in the content store (CS) and afterwards subsequent requests should be satisfied much quicker than TCP/IP. As true NDN routers are not commercially available, we implemented NDN's functionality over TCP/IP by setting up personal computers as NDN Routers.

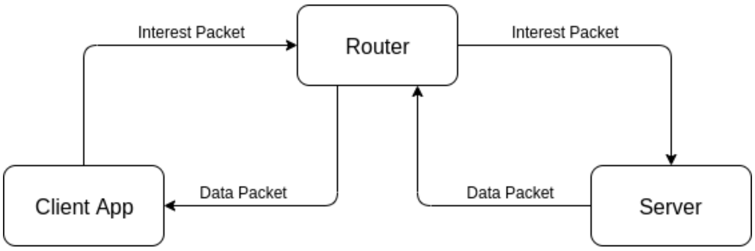


Figure 3.3: NDN Workflow

The NDN Routers decide on how to treat the interest packets using the following criteria.

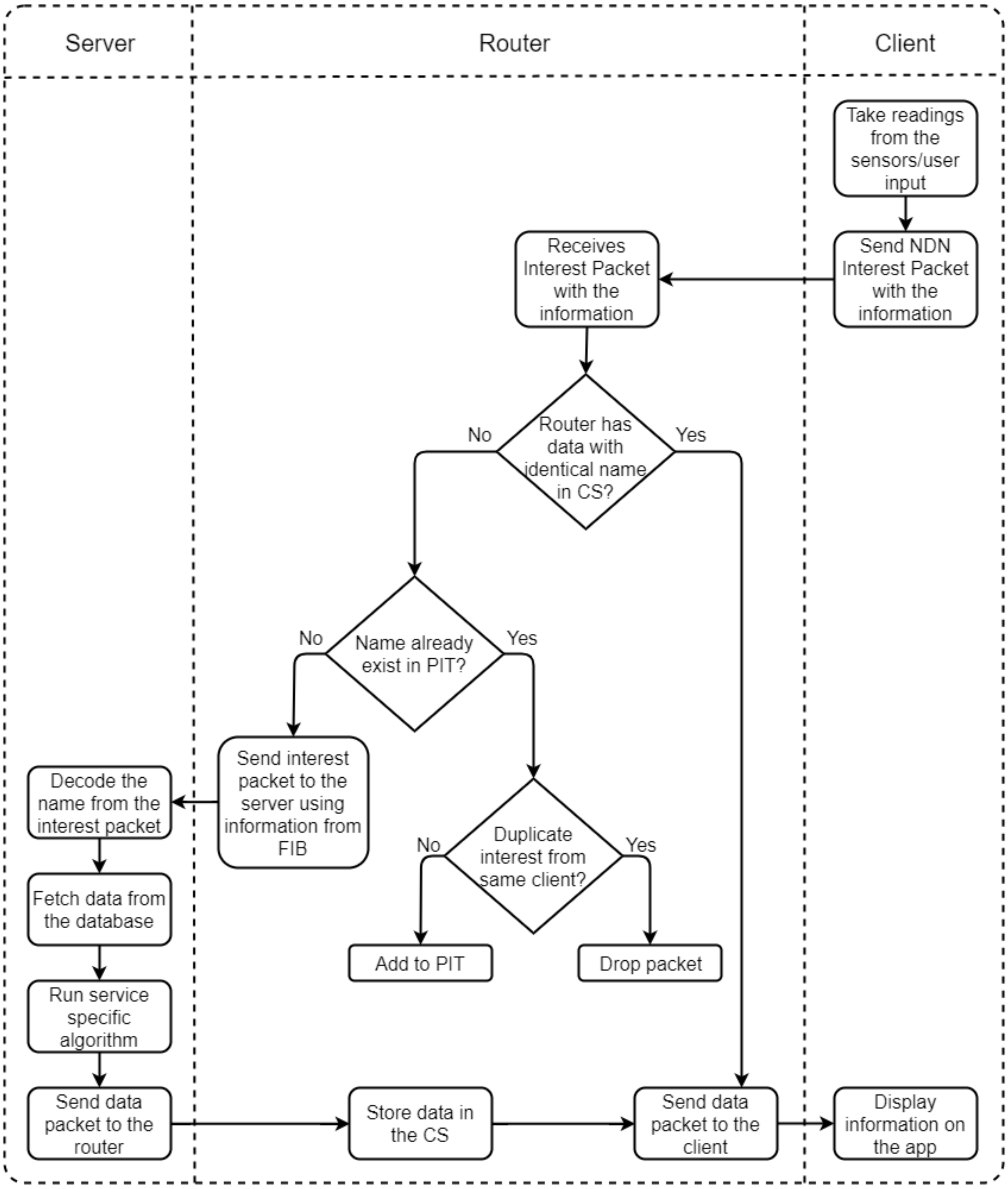

Server
Router
Client
Take readings from the sensors/user input
Send NDN Interest Packet with the information
Receives Interest Packet with the information
Router has data with identical name in CS?
No
Yes
Name already exist in PIT?
No
Yes
Send interest packet to the server using information from FIB
Duplicate interest from same client?
No
Yes
Add to PIT
Drop packet
Decode the name from the interest packet
Fetch data from the database
Run service specific algorithm
Send data packet to the router
Store data in the CS
Send data packet to the client
Display information on the app

Figure 3.4: General overview of the process

We tested the following steps using NDN and TCP/IP and collected their performance data.

- Floor Detection: We set up a mobile device on the ground floor to act as the reference barometric pressure sensor. afterwards we tested the floor detection performance using the smartphone app. As pressure varies a lot, we required a few attempts before we started to see result gains from NDN.
- Fetch Map: Fetching the map was benefited the most from NDN as it started to show promising results after a single attempt. The map is static and there are only a few floors so this is expected.
- Locate Me: After the map was loaded we tried to locate ourselves. As RSSI values vary quite a lot, we needed quite a lot of attempts before we started to see promising results from NDN over TCP. However it should be noted that even without CS caching NDN performed just as well as TCP/IP
- Navigation: The final step was picking a destination to navigate to. We tried TCP/IP performance and also used the same source and destination for NDN repeatedly for testing. This resulted in great performance gains using NDN, while it still performed just as well as TCP/IP under regular circumstances where the task was unique.

3.3 Floor Detection

As mentioned in the previous chapter, we will be detecting the current floor using barometric pressure sensors that are available in almost any modern smartphone researchers have recently drawn attention to improve performance using pressure information gathered from a smartphone barometer. When the floor changes it causes the pressure to change as well. In addition, the difference of pressure is sufficiently large between floors which help to distinguish between floors[47], [48]. However, the prediction of floors by pressures is sometimes difficult. In the first place, pressures are not stable. With changing weather conditions it oscillates and changes continuously. The absolute pressure value measured on a particular floor at a given time is not used for detecting the floor at a later stage[49]. Secondly, to meet constantly changing conditions, it is difficult to use the comparison stress indicators at the local national MA (Meteorological Administration). The stress calculated on a territorial MA is often not equivalent to that of a tower, even if it is assessed on the building's surrounding MA. Thirdly, even when the pressure is measured at the same location the pressure values measured on different smartphones vary greatly from one another[50], [51]. Even the barometer sensors in the same smartphone version have the same problem. In the same smartphone type, the spectrum of errors of barometer sensors is too large to be used for floor detection. Due to pressure instability and inconsistency, certain reference data are required for accurate floor detection. The pressure value of a pre-installed reference device in a certain position where the altitude in the building is known, and a pressure difference between the user's smartphone and the reference device is necessary. The next figure shows a schematic method illustration. The process is of three stages. In the first phase, pre-installed test equipment in a target building tests the pressure level. For each floor of the building, the estimated pressure rate is determined in the second step. The difference between the pressure levels of the pre-installed reference device and the Smartphone consumer is measured in the last step and after the pressure difference is measured, the floor is identified.

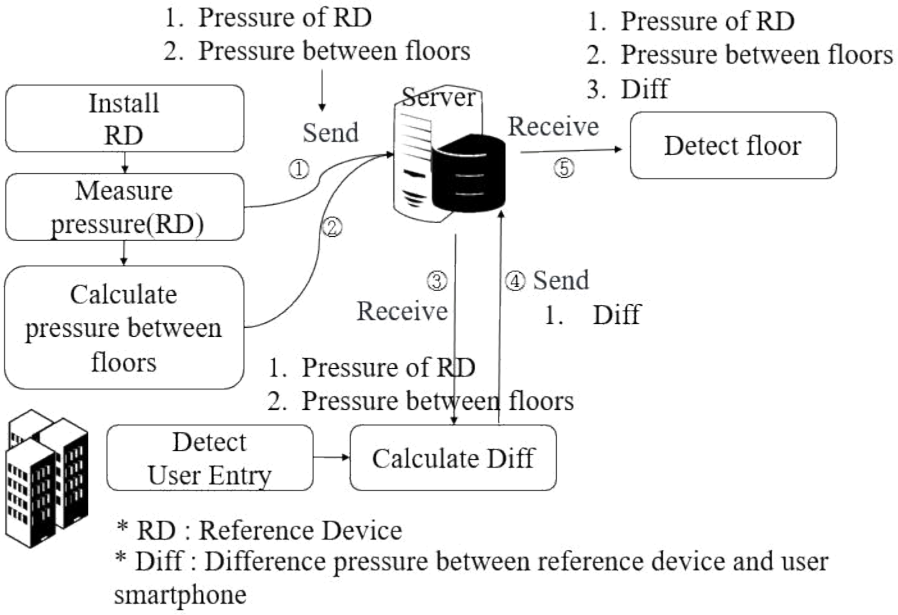


Figure 3.5: Schematic view of floor detection system

Step 1: installation of reference device

In order to install a reference system and measure floors by pressure level we require the altitude of each floor and the height of the ground level of a target tower. We can obtain them from a CAD file from the building, normally open to the public in many countries. If an embedded barometer sensor is used, any device can play the role of the reference device. The air pressure can be consistently measured with the reference instrument, and the floor can be detected despite changes in the weather conditions which causes pressure to fluctuate.

Step 2: Calculating pressure at each floor

It is well known that if the water level pressure is known, the altitude can be calculated at that point. Equation 1 transforms pressure data into altitude which is currently widely used.[52] The temperature is not taken into account here since temperature is used to calculate the exact altitude as a corrective factor. However, the difference in height between each floor is used rather than using an exact altitude. The pressure of each floor can thus be calculated using the below equation without taking into account the temperature factor.

$$h = \left(1 - \sqrt[5.255]{\frac{p}{1013.25}}\right) \times \frac{288.15}{0.0065} \tag{3.1}$$

where h is the altitude and p is the pressure. When the pressure is changed by about 0.1pha, the height will change by about 0.837m. Therefore, the equation can be rewritten as.

$$p = \frac{0.1 \times h}{0.837} \tag{3.2}$$

Using this equation, the pressure of each floor can be determined if it's height is known. The predicted values of pressure were known to be the real pressure value for that floor. The pressure value, which the equation calculates, was subtracted from the value of the current floor pressure to predict the pressure value of the floor above. The magnitude of the pressure value for the existing floor is added to the pressure level measured in the equation to estimate the lower floor pressure in accordance with the present floor.

Step 3: Floor detection

Correction of inter layer pressure

The pressures measured in various smartphones using pressure sensors are usually different, even if they are measured simultaneously on location. There is a consistent difference. In other terms, the gap persists almost everywhere even when a smartphone calculates a pressure level greater than another calculated smartphone. Therefore, every smartphone (Up) to a reference system (Rp) can be adjusted to cope up with the difference. Adjustment is made by adding or removing (Bpx) the difference between Rp and Up. The below equation determines the shift.

$$Bpx_{adjust} = \begin{cases} Bpx_{original} - Diff, & if\, Rp > Up \\ Bpx_{original} + Diff, & if\, Rp \leq Up \end{cases} \tag{3.3}$$

Adjustment of Bpx measured from a smartphone may be used for the detection of floors regardless of smartphone models.

Setting of critical range

Once step 3(Floor detection)-A is completed, a critical pressure range is established. The critical range is the height from the above floor to a certain height where the person carries a smartphone. The smartphone pressure value is then checked to detect the floors. The floor is described as the floor in which the device is situated if the pressure level belongs to the particular range of a specific floor. Depending on the location of the reference device installed and the deployed floor height, the critical range changes. A smartphone is usually held in a 0 cm 180 cm range. Therefore the critical distance may be defined within the range of-60 cm / 120 cm if the reference unit is positioned at a height of approximately 60 cm from the floor. It is a disadvantage since a reference device is required to be installed in a building. However, considering that the costs of installing a reference device are very low and that no other method can detect floors as reliably, the proposed method should be taken into account in improving indoor positioning systems. Nonetheless, the user's smartphones can be used as a reference tool instead which will be more successful.

3.4 Self location methodology

For self location we are going to be using the K-Nearest Neighbor algorithm. K-Nearest Neighbor is essentially an algorithm to classify something into one of many classes based on the properties of the aforementioned thing and the properties of the previous members of the predefined classes. If, for example,the algorithm is provided the properties of a fruit, it would classify the fruit as a specific fruit such as Apple or Orange based on the database it has of other already classified fruits by comparing their properties. As such KNN is a 'Supervised' algorithm as it needs to rely on pre-existing data. The algorithm determines the distance of nearby neighbors using the 'Euclidean Distance' method.

Weighted KNN

Weighted KNN is a variant of KNN where the members of the K Nearest group are assigned weights based on their distance or 'closeness'[53]. This results in the nearer among the neighbors to have a larger vote in the classification process.

3.4.1 Determining K

K refers to the number of 'Neighbors' that are taken into consideration or has a vote in determining the class of the item in question. When KNN starts to classify an item, firstly it starts to look at all the neighbors where the closest neighbors are those who have data that most closely matches that of the current item. But for the algorithm to function it has to set a limit on how many neighbor's it will take into consideration. That number of neighbors is referred to as K[54]. Picking a suitable value for K is referred to as parameter tuning and it is a very important step to get desired results from KNN. If we refer to Figure below, we can see that if the K is set to '3' the new object gets classified as a triangle whereas if set to '5' it gets classified as a circle.

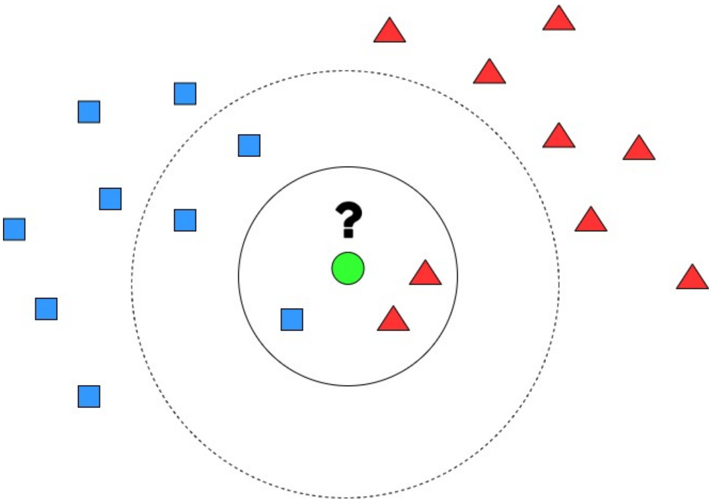


Figure 3.6: KNN Classifier

If K is set to number too low, the result will mostly be random. If K is set to a value too large it will simply take a long time to process. One good practice is to pick the Square Root of the total number of datapoints sqrt(n) and then convert it to an odd number by adding or subtracting one if necessary to avoid classification confusions[55].

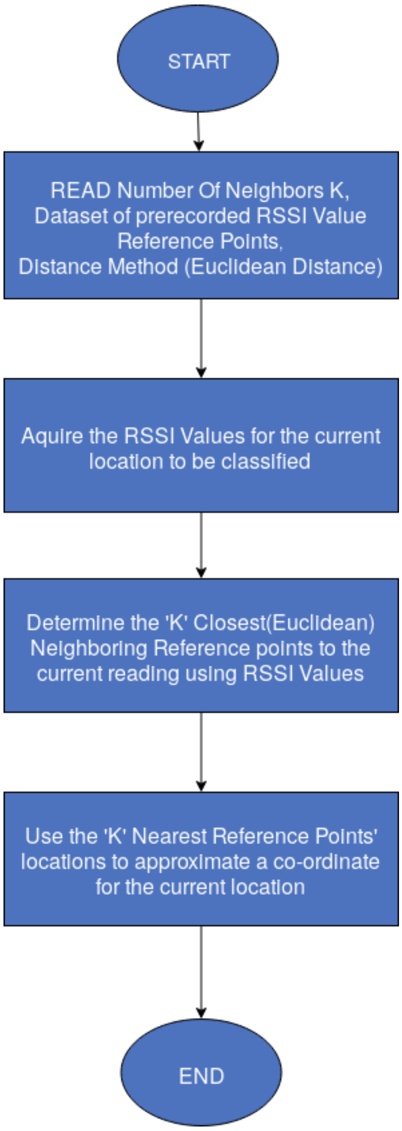

START
READ Number Of Neighbors K,
Dataset of prerecorded RSSI Value
Reference Points,
Distance Method (Euclidean Distance)
Aquire the RSSI Values for the current location to be classified
Determine the 'K' Closest(Euclidean) Neighboring Reference points to the current reading using RSSI Values
Use the 'K' Nearest Reference Points' locations to approximate a co-ordinate for the current location
END

Figure 3.7: Location detection using KNN flowchart

3.4.2 Algorithm

This is a simple implementation of the Weighted KNN Algorithm.

```
Algorithm 1: The Weighted KNN Algorithm
input  : D the set of reference points, x the current location
output: Location of x
begin
    for each reference point y belongs to D do
        calculate the distance D (y,x) between y and x ;
    end
    initialize N = ∅ where N is the subset of D with the k closest members ;
    for 1 to k do
        insert into N the closest co-ordinate n from D;
        assign weight to n;
    end
    from k nearest neighbors N of the co-ordinate x calculate the location of
      of x using ;
    $c_x = arg\,max \sum_{y \subset \mathsf{N}} I(c = class(y))$ ;
end
```

3.4.3 KNN's suitability for our work

Whether or not to use KNN for a certain purpose depends on a few conditions. KNN works best when some conditions are full-filled[54]. For KNN to work efficiently, some properties of the data should be

1. Labelled: The data should be labelled, which means the classes should be identifyable in some way.
2. Noise Free: The classes should be of the same type. For Example, All the classes should be Fruits, Animals etc.
3. Small Dataset: KNN is a Lazy learner so it works best with relatively less complicated data.

If we compare these conditions to our use case, we can see that these conditions are all satisfied.
Our data is labelled, We are trying to classify our current Access Point Readings as one of the prerecorded reference points. So all the classes are a single RP.
The data is also noise free. All the classes have similar data and they are all reference points
Although there is a large dataset, it's still relatively small for KNN
Our data is stored as Reference Points with a set of RSSI values of Access Points stored within them. When a person tries to locate themselves, their device takes some RSSI readings from the valid predetermined access points. then when the WKNN algorithm is run on the RSSI readings, they are compared to the RSSI readings of Reference Points stored in the database. The RP whose AP RSSI values most closely match the device's current readings is resultant Classification, or we could say the person is at that Reference Point. Here the RP's are the classes the

current reading is being classified as. We have set K = 4. As it is apparent, Each RP or Class has only one set of values. So the question arises whether we needed K-Nearest Neighbors instead of Nearest Neighbor. The reason is that we are using multiple reference points to calculate a floating location instead of classifying the current location as a single RP. Thus we use the multiple points to classify the current location as a value between multiple classes with the help of KNN.

3.5 Navigation Approach

Due to the complex structure of buildings and infrastructures, Navigating between indoor areas plays an important role in the indoor positioning and navigation system. Finding the position of the desired destination will not help, as there are many routes in the structure, including hotels or malls. Therefore, it is very important to determine the optimal path to the destination so that users can get to the destination quickly.

3.5.1 Path finding Scenarios

Previously, numerous studies have been extensively conducted on indoor navigation schemes utilizing mobile technologies such as Wi-Fi, Bluetooth. Indoor targeting, triangulation, fingerprinting and cell-ID strategy methods are the three ways used for indoor positioning[56]. Today, citizens use location-based service that is also regarded as Location Based Network[57], [58]. LBS cover all wireless media services that supply user-specific information. In order to locate the appropriate site, such facilities need indoor engineering for position determination (path finding). Global Navigation System (GPS) is widely used for global positioning. The use of indoor settings by GPS is however restricted, because it relies on satellite signal[59].
Indoor navigation systems that use wireless signal such as Wi-Fi or Bluetooth are progressively of interest[60], [61]. in order to meet these challenges. The dynamic configurations of environments, constant human movement and inner complex structure of walls, means that most indoor positioning solutions are ineffective. An A* algorithm can be applied to seek the route and enhance the quality and exactness to improve the precision of the positioning system utilizing wireless communications.

3.5.2 Proposed Algorithm: A*

It is an algorithm for finding the shortest route, which seeks the least optimal route by looking for the lowest-cost(low length, shortest time etc.) It will visit all possible paths that lead to the target node and takes into account the ones that lead to the target node the fastest. It is an extension of Dijkstra's algorithm. Working of A* algorithm: The algorithm was built for weighted graphs: it creates a route tree from a reference node in the network and extends the routes once in a while to one of its paths to the destination node. A* should decide which of its partial routes should be expanded to one or more longer routes for every stage of its main loop. It is based on a value estimation (total weight) that is always accessible at the goal node. To addition, A* tends to pick the route that minimizes the below equation

$$G(n) = a(n) + h(n) \tag{3.4}$$

Where n is the last node on the path, a (n) is the value of the route from the beginning node to n and h(n)is a heuristic route that calculates the lowest cost from n to the desired destination. In order for the algorithm to determine the exact shortest route the heuristic feature must be allowed. This also ensures that the actual cost to reach the nearest destination node from node n must never be overestimated. The below figure (Figure:3.8) is an algorithm flowchart that provides a description of how it operates.

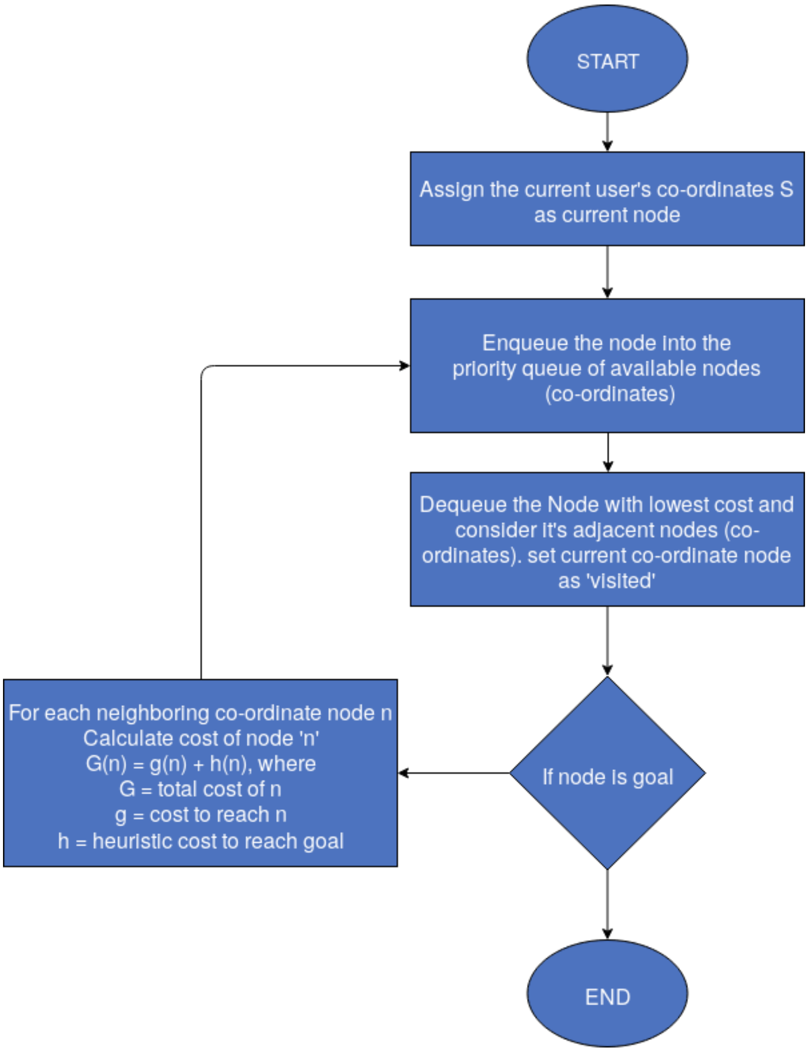

START
Assign the current user's co-ordinates S as current node
Enqueue the node into the priority queue of available nodes (co-ordinates)
Dequeue the Node with lowest cost and consider it's adjacent nodes (co-ordinates). set current co-ordinate node as 'visited'
If node is goal
For each neighboring co-ordinate node n
Calculate cost of node 'n'
G(n) = g(n) + h(n), where
G = total cost of n
g = cost to reach n
h = heuristic cost to reach goal
END

Figure 3.8: Navigation using A* algorithm flowchart

3.5.3 A* algorithm's advantages over Dijkstra's algorithm

As As previously mentioned an A* algorithm is an expansion of Dijkstra's algorithm[62]. A* algorithm can be considered as the heuristic reliant version of Dijkstra algorithm. It chooses which vertex to explore next upon the value of G(n)=a(n)+h(n) which makes it known as the "best first search" algorithm. The 'h' determines the estimate cost using a heuristic whereas 'a' determines the current cost of the path. The A* algorithm is complete and optimal, the results are quicker and more effective[63]. The original algorithm by Dijkstra is slower in comparison to A* because Dijkstra constantly tries to improve the node's initial approximation (cost). Because of this, hitting the goal node takes longer. Whereas A*'s heuristic based approach has a higher probability of finding the desired goal quicker in most cases[64]. A* mainly improves its performance by using heuristic function that will prioritize the nodes based on their estimated distance from the goal which will result in a faster path detection rather than Dijkstra that prefers to explore all possible nodes.

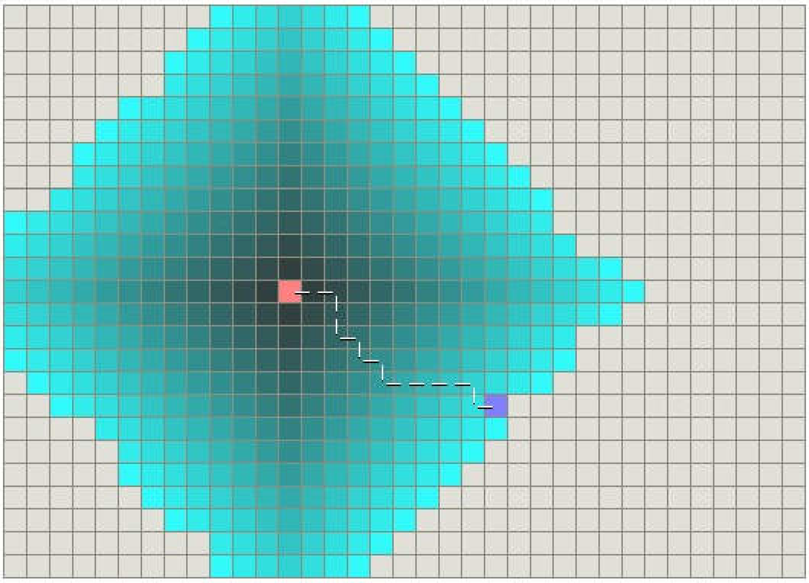


Figure 3.9: Path finding using Dijkstra's Algorithm

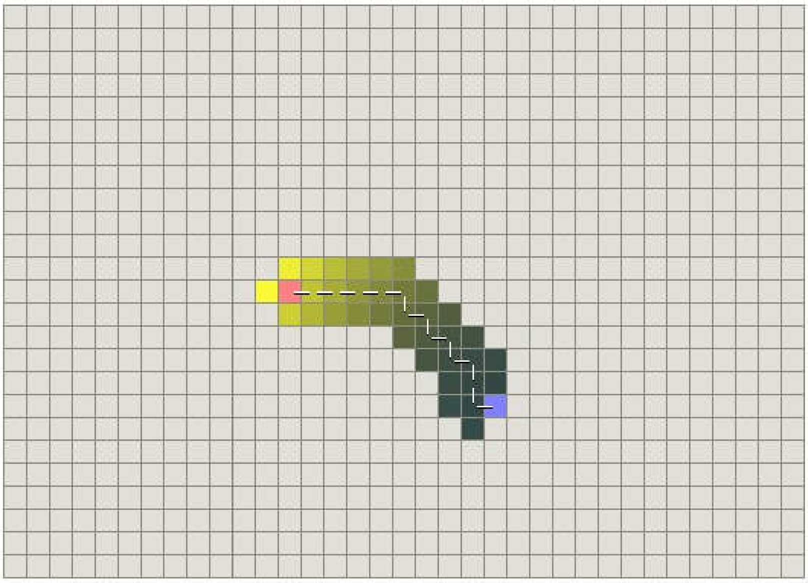


Figure 3.10: Path finding using A* Algorithm

3.5.4 Algorithm

Here is a simple implementation of A*

Algorithm 2: A* Algorithm

```
input : Start location s, Destination (Goal) node g, Navigation Nodes
Closed set = visited co-ordinates;
Open set = available co-ordinate nodes for expansion;
Extract-Min-F = function to determine total cost of node ;
begin
    A* (s, g);
    Closed set ⟵ empty set;
    Open set ⟵ include s;
    Extract-Min-F = G [s ] = 0, H [s ]=H_calc[s, g ] ;
    while Open set $\neq \emptyset$ do
        CurrentNode ⟵ Extract-Min-F(Open set) ;
        if CurrentNode == g then
            return Route ;
        for each Neighbor node N of CurrentNode do
            if N is in Closed Set then
                do Nothing;
            else if N is in Open Set then
                calculate N's G, H and F;
                if G [N on the Open Set] > calculated G [N ] then
                    RELAX(N, Neighbor in Open Set, w) ;
                    N's parent = CurrentNode —& add N to Open Set ;
            else
                calculate N's G, H and F;
                N's parent = CurrentNode —& add N to Open ;
            end
        end
    end
end
```

3.5.5 Suitability for our implementation

As we can see A* is suitable for our requirement of navigating to a destination location. We also saw the performance difference between the two algorithms. The only distinction between the two algorithms is that A* gives a faster direction by using a heuristic , while Dijkstra investigates all possible paths. A* produces better performance by using heuristics to efficiently expand only the routes that are closer to the destination. For these reasons A* was chosen as the method for navigation.

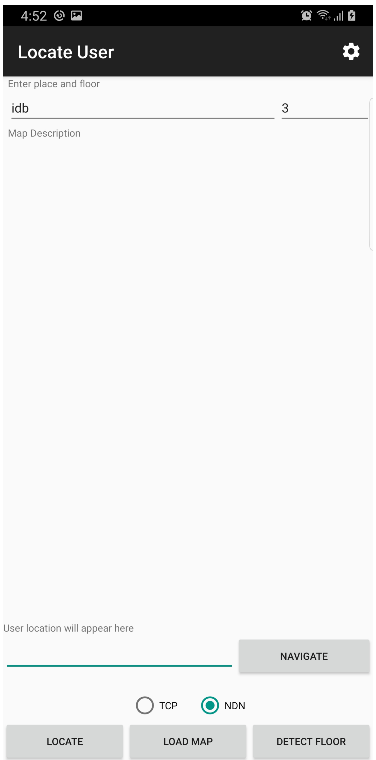
3.6 Networking implementation using NDN

3.6.1 Floor Detection

Floor detection is based on two barometers, one on the client and one on the server. The client device records it's current pressure in hPa and sends it to the server. There is already information about the building stored in the database. The server use those information to calculate the client's floor. To retrieve this information using NDN we used the following example naming structure,

$$”/ips/idb/floor\_detection/1015.4”$$

Names in NDN are hierarchical. the /ips indicates the floor detection service. The /idb references the specific structure which in this case is the BCS Computer City. The next part floor_detection is the floor detection service and the value after it is the pressure reading from the client. When the server satisfies the given request for the first time, the floor information for that specific air pressure is stored in the router's Content Store (CS). Whenever there are other requests with same air pressure value, the name for NDN interest packet will also be the same thus the router can send the data packet from its cache without contacting the server.



idb 3

○ TCP ● NDN

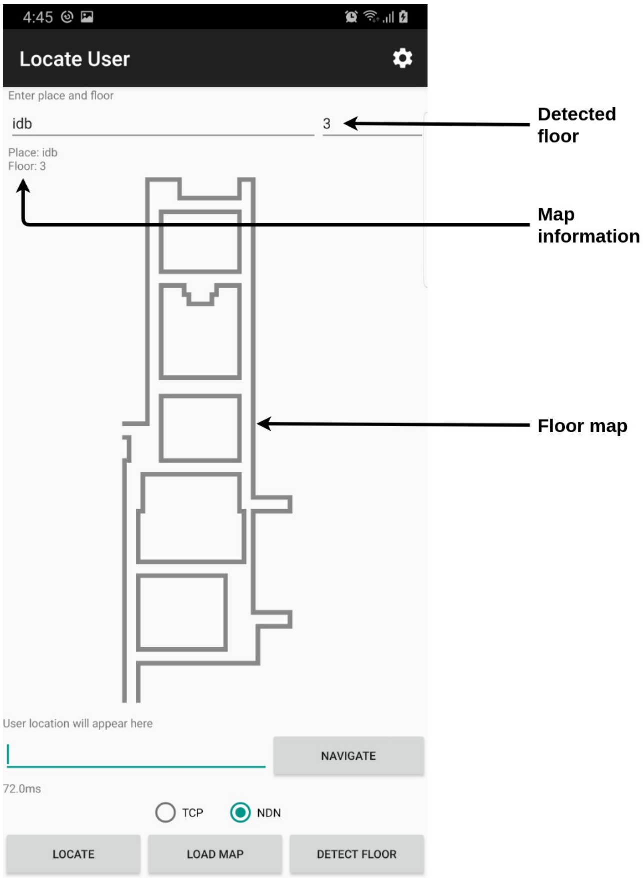
Figure 3.11: Floor Detection

3.6.2 Map Service

After detecting the floor successfully, the next step is to generate the map in client’s device. The map we used was very simple as our intention was to compare the performance of NDN. Following is an example of how the name for loading map was structured.

”*/ips/idb/load_map/2*”

The ”/ips/idb/” part is identical to floor detection name. Following part /load_map indicates the map delivery service. Finally, ”2” is the floor number for which the map was generated. The NDN data packet contained all necessary coordinate information to generate the map. This data packet also contained the information of the APs in that floor and the sequence which is very important for positioning.



4:45

Locate User

Enter place and floor

idb 3

Place: idb
Floor: 3

User location will appear here

72.0ms

○ TCP ● NDN

Figure 3.12: Load Map

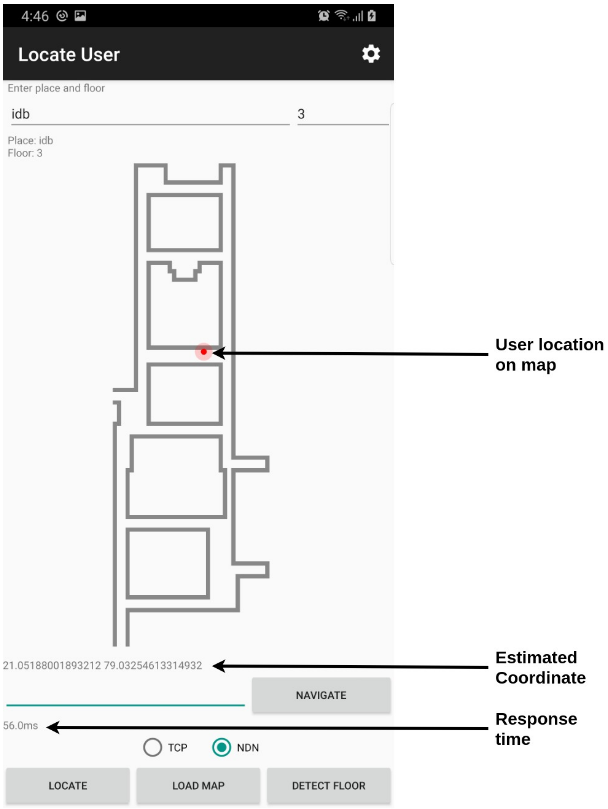
3.6.3 Positioning

The client first measures the signal strangths from predetermined APs in it’s vicinity and then requests that data. This information is sent to the server. The server then accesses the database containing the information of all the Reference Points and their corespondent RSSI values from the Access Pointss. Based on the observed RSSI from the client and existing RSSI from the database, we used WKNN algorithm to estimate user’s location. The name for the interest packet was constructed in the following manner,

$$"/ips/idb/position/3/-43\_-32\_-84\_-57\_-55..."$$

Here ”/ips/idb/” part is identical to floor detection name. The next part /position indicates the service to detect position of the user. After that, ”/3” is the floor number. The last part contains all the observed RSSI values from the client device. The values are in a specific order so the server can understand which RSSI value is for which AP. They are separated using ”_”.

Similar to floor detection, after satisfying the client device with the position information for that specific name, cache will be stored in the NDN router’s CS. Future interest packets with exact same RSSI values will be satisfied by the NDN router itself instead of contacting hte server. Now it is difficult to match all the RSSI values for all the individual APs even if the user is standing in the exact same spot. However this service will be heavily used since locations are constantly updated even on the move. So eventually after several requests there can be some interest packets with identical names.



Enter place and floor

idb 3

Place: idb
Floor: 3

21.05188001893212 79.03254613314932

56.0ms

○ TCP ● NDN

User location on map

Estimated Coordinate

Response time

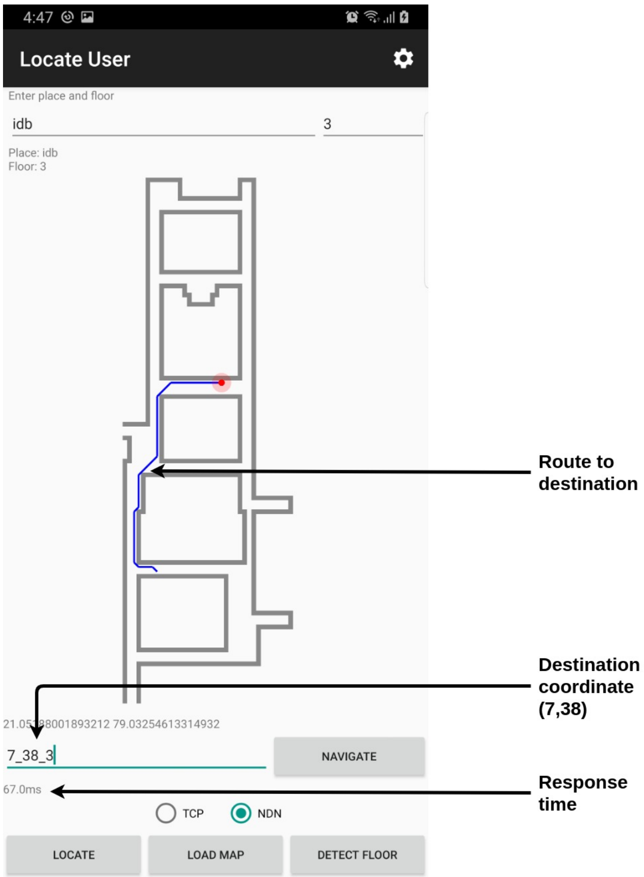
Figure 3.13: Locate Me

3.6.4 Navigation

Finally the last step is for the user to navigate to the destination. The user picks a destination on their device and the request for navigation service is sent to the router. The server calculates the smallest possible route using A-Star algorithm. The routing information is sent to the client device. Following is an example name we used for navigation,

”/ips/idb/navigation/15_80_2_27_119_2”

Again ”/ips/idb/” carried the same meaning as floor detection and positioning. The following part was indicating the service name which is navigation service. In the last part, the first three values told the current position of the user. Current X coordinate is 15, Y coordinate is 80 and 2 is current floor. The last three values were used for the destination co-ordinates. Destination X coordinate is 27, Y coordinate is 119 and 2 is destination floor.



Enter place and floor

idb 3

Place: idb
Floor: 3

21.05188001893212 79.03254613314932

7_38_3

67.0ms

Route to destination

Destination coordinate (7,38)

Response time

Figure 3.14: Navigation

Chapter 4

Result Analysis

4.1 Result Graph of Different Modes

As we have implemented our NDN based indoor positioning and navigation system in IDB bhaban, we were able to gather data from our proposed methodology. We measured the time it takes to receive data for the client from the server. The time was measured in nanoseconds and then converted to milliseconds. Since our client is an android application, we were able to use *System.nanoTime()* to measure precise time to execute one method. We measured each of the modes 20 times using NDN and TCP/IP protocol. For NDN, we took two different measurements, one for cached(best case scenario) other for no cached(worst case scenario). We then plotted the measured times in graphs and compared both protocol for each of the modes.

4.1.1 Result Graph of Floor Detection

To detect floor a request was sent from the client's device to the server carrying the air pressure data from the barometer sensor. To compare our proposed NDN based system with traditional TCP/IP architecture, We ran our test in two different scenarios. One for the worst case scenario, where there was no cache stored in NDN router. This gave us an idea of how NDN based system would perform for different requests compared to TCP/IP.

Second for the best case scenario, when the data was already fetched once and stored in the NDN router's content store. For this we stood in a specific floor putting the client device in same elevation and recorded the time. This way we were able to test the NDN's performance for similar requests. Since this time it was sending similar requests over and over again, data was being retrieved from the router resulting in much faster response time. This gave us the idea about performance we can expect when people from the same floor and similar elevation will be requesting for the floor information.

Floor Detection Without NDN Cache

To achieve the result, we went to different floors and put the device at different elevation so each of the requests are different. This way we made sure the requests are not identical and not hitting NDN router's content store(cache). Time we got was almost similar to traditional TCP/IP (Figure 4.1). For each attempts, TCP/IP ranged within around 120ms to 230 ms where as NDN ranged around 100ms to 240ms. This was our worst case test.

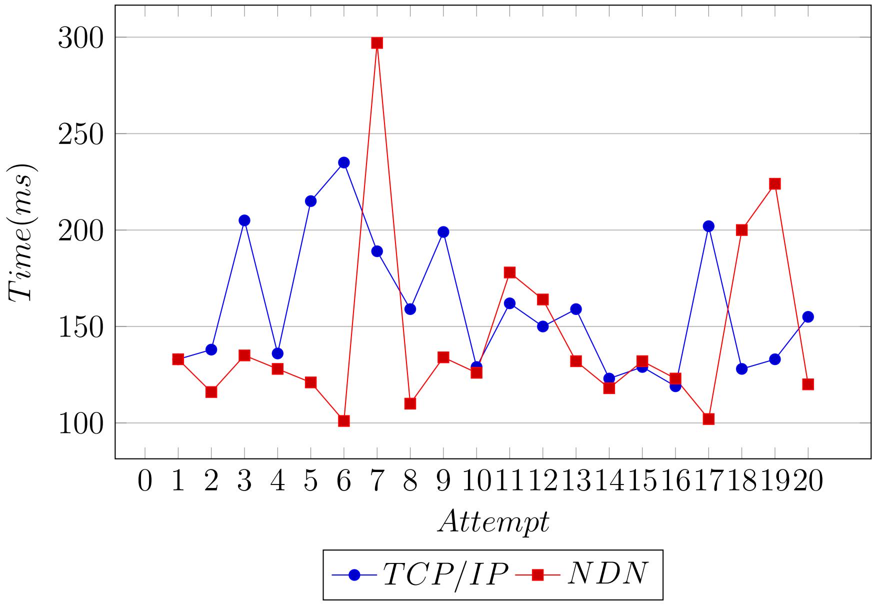


Figure 4.1: Floor detection time without NDN cache

Floor Detection with NDN Cache

By analyzing the results (Figure 4.2) we can see that when we stood in same location with our device, which was our best case scenario, response time is much faster after the first request. It is because after the first request, they were being retrieved from the NDN router. there is one spike in the graph due to slight change in air pressure during the test. Otherwise all the results were between around 50ms to 75ms.

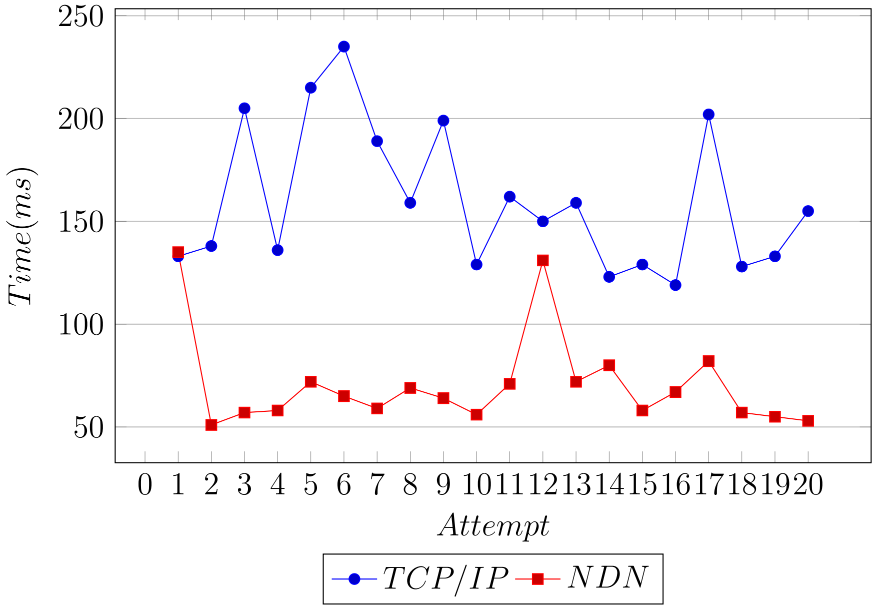


Figure 4.2: Floor detection time with NDN cache

4.1.2 Result Graph of Positioning

For the calculation of the users location in a particular floor, the received RSSI values of predefined access points needed to be sent to the server. For this mode we also compared NDN, both with and without cached, with traditional TCP/IP architecture. When testing NDN without cached data, moved around in a floor so that we get different RSSI values making each request to the server different.

Positioning Without NDN Cache

When testing NDN without cached data, moved around in a floor so that we get different RSSI values making each request to the server different. Response time was similar to TCP/IP(Figure 4.3). TCP/IP's response time ranged within 160ms to 300ms most of the time where as NDN's response time ranged within 180ms to 280ms most of the time with some exceptions.

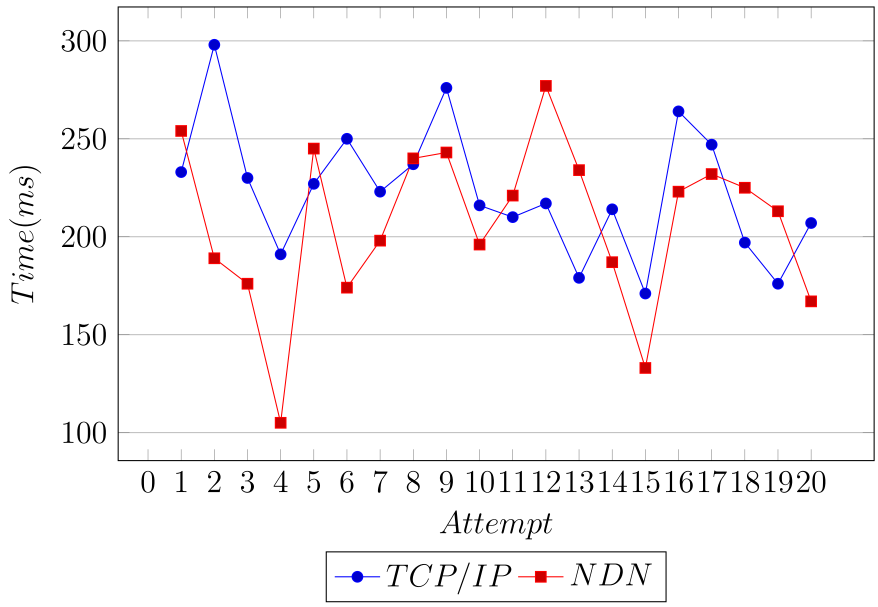


Figure 4.3: Positioning time without NDN cache

Positioning With NDN Cache

It was challenging for us to test cached NDN performance as it is very difficult to get all the RSSI value from the access points same everytime, even if we stand in exact position as the RSSI fluctuates with other variables. We did get some readings that hit the cache. From the graph (Figure 4.4) we can see that in our best case scenario, it was hitting cache only 10 times. Others were not hitting cache and requests were processed by the server instead of the router. When it hit the cache, we got around 50ms to 80ms of response time.

4.1.3 Result Graph of navigation

Navigation works by sending two location points from the client to the server. One is the current position of the client which is retrieved by using the positioning mode,

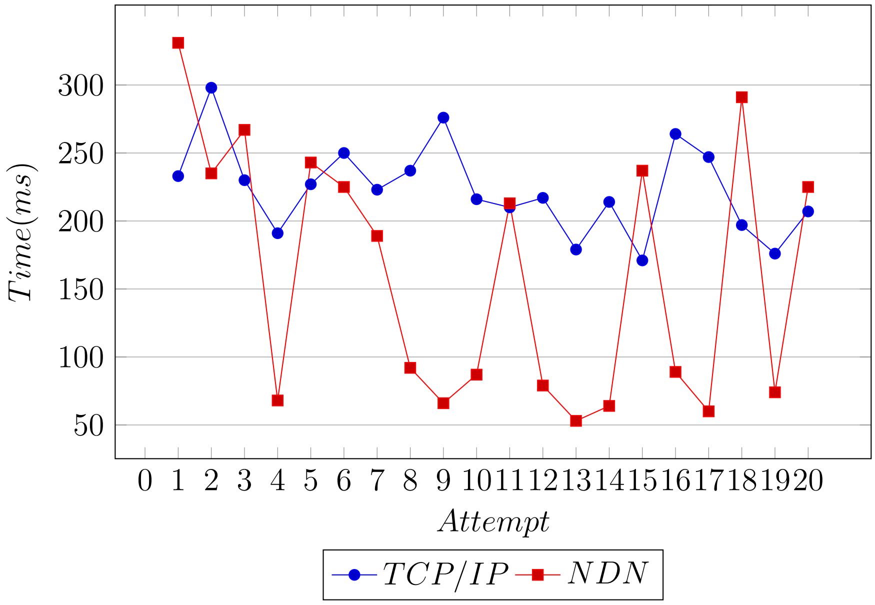


Figure 4.4: Positioning time with NDN cache

other is the destination point set by the client. To test this mode, we decided to opt for the longest routes possible in IDB.

Navigation Without NDN Cache

This test was done by sending different destination from same position of the device so that each of the requests are different. For this test we took coordinate (1,1) as our current position which is in the furthest south-west corner and for destination we chose (27,119), (27,118), (27,117), (27,116), (27,115), (27,114), (27,113), (27,112), (27,111), (27,110), (28,119), (28,118), (28,117), (28,116), (28,115), (28,114), (28,113), (28,112), (28,111) and (28,110) which are in the furthest north-east corner of the building. Result shows that response time is equivalent to TCP/IP (Figure 4.5). In our worst case test in navigation service, TCP/IP's response time ranged within 150ms to 250ms most of the time where as NDN's response time ranged within 150ms to 300ms most of the time with some exceptions.

Navigation With NDN Cache

This test was done by sending same destination from same position of the device so that each of the requests are identical. or this test we took coordinate (1,1) as our current position and (27,119) as our destination for all the tests. Result shows that response time after the first request is much quicker as the data is being retrieved from NDN router's content store (Figure 4.6). In the first attempt, 192ms from NDN and 150ms from TCP/IP. After that since NDN was getting data from the cache, we got much less response time compared to TCP/IP, about 50ms to 70ms.

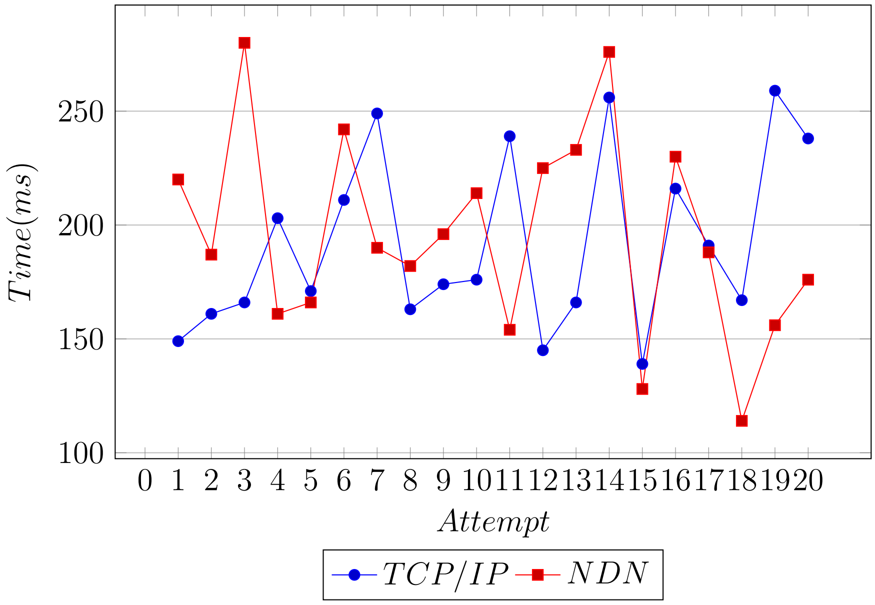


Figure 4.5: Navigation time without NDN cache

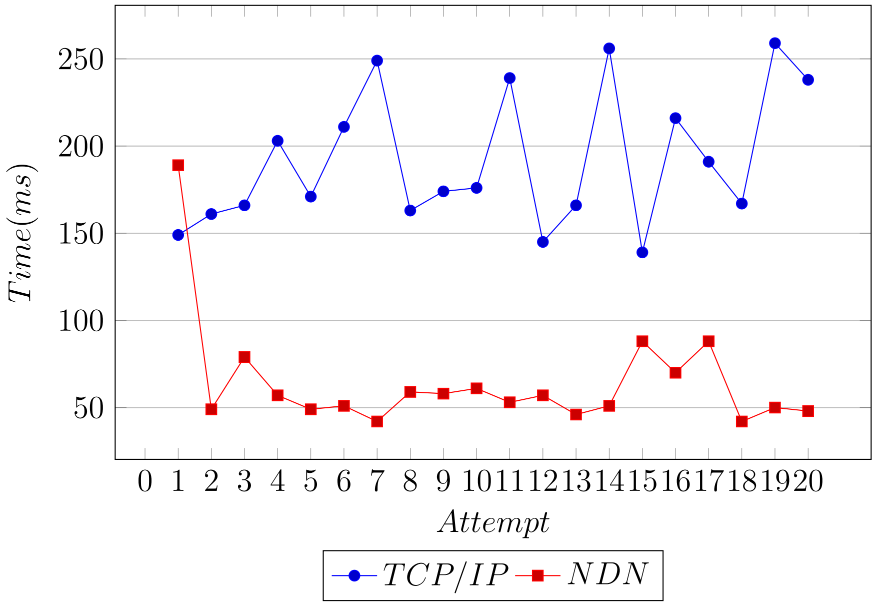


Figure 4.6: Navigation time with NDN cache

4.2 Overall Comparison of NDN with TCP/IP

From the results we can see that our NDN based system performs just as good as TCP/IP when no cache is hitting but outperforms TCP/IP when hitting cache. On average we got 159.9ms response time when detecting floor using TCP/IP, 144.7ms when using NDN with deferent elevation and 70.6ms when using NDN with same floor and same elevation. For positioning, on average we got 223.2ms response time using TCP/IP, 159.4ms with NDN while standing on the same location and 206.6ms with different locations. For navigation, on average we measured 191.9ms response time when using TCP/IP, 195.9ms response time when using NDN with different destination and 64.4ms response time when using NDN with same destination (Figure 4.7).

By taking the average time for each of the modes we can see that we got 77.5% better response time in floor detection, 33.4% better response time in positioning and 99.5% better response time in navigation than TCP/IP for similar requests (Figure 4.8).

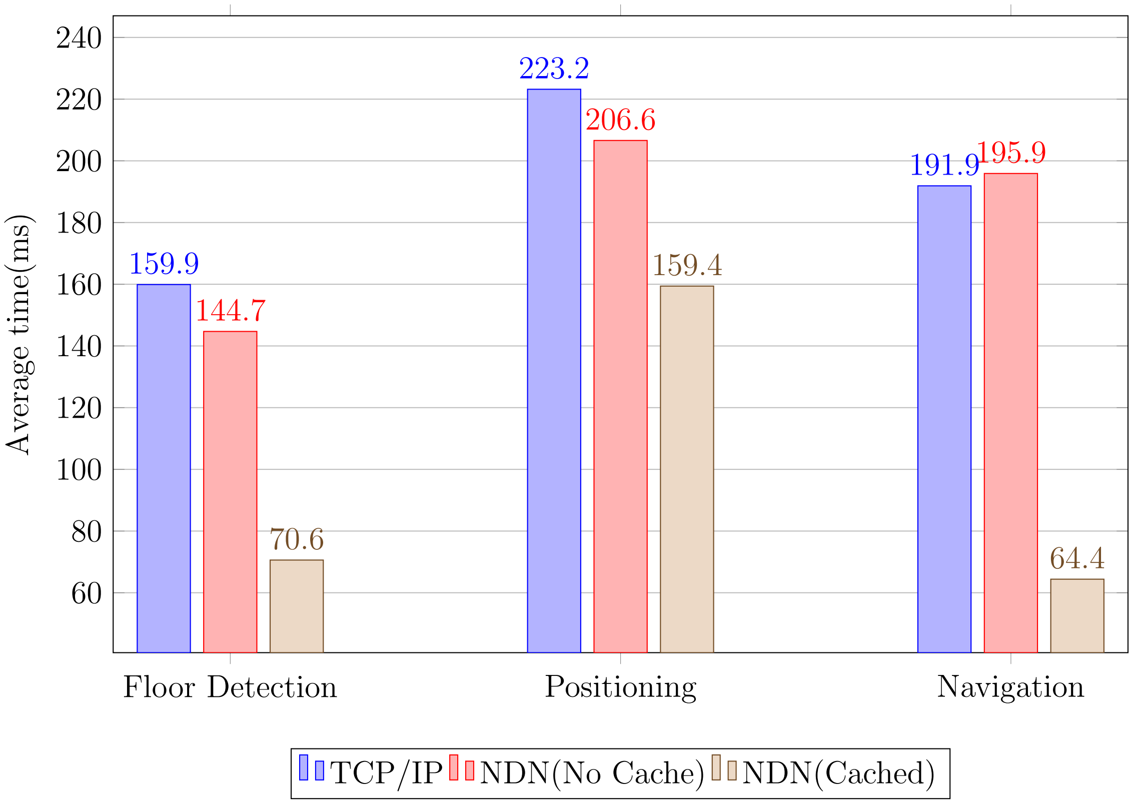


Figure 4.7: Average time for each modes

4.3 Limitations

Even though the performance and benefits that NDN brings are great and very useful, there are obvious limitations when implementing this system.

- NDN is still in its early stage. Developments are still on going.
- Although NDN can run natively above Ethernet, there isn't a global scale native NDN network yet. Devices that are connected to other networks will not be able to communicate via NDN
- The advantages in indoor positioning system can only be achieved if the system is connected to an NDN network.
- As a completely different internet architecture, existing applications will not work. Existing applications have to be reworked to work with NDN.
- New NDN compatible routers are needed or existing routers' firmware need to be updated to take advantage of our system.

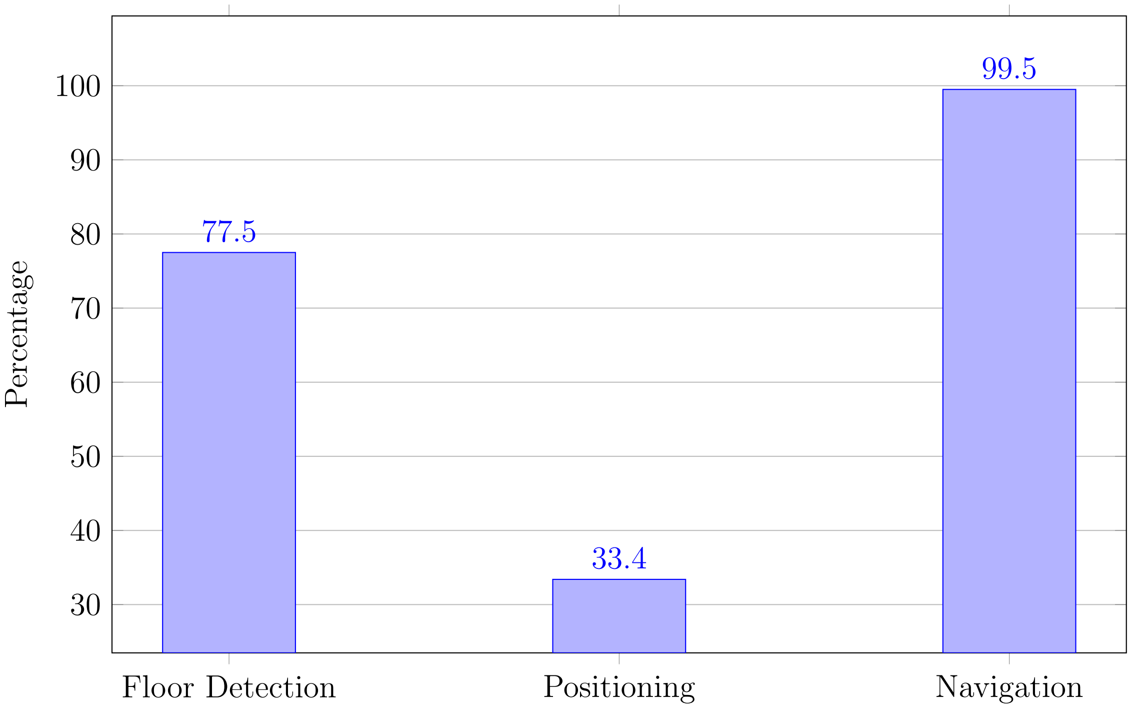


Figure 4.8: Improvement over TCP/IP

Chapter 5

Conclusion

Indoor Positioning technologies have become a trendy topic in recent days and it is expected to grow even further in the near future. With the advent of larger and larger mega structures like shopping malls, hospitals, airports etc navigating indoors has become quite an issue that cannot be solved by GPS. Hence indoor navigation solutions have been introduced to tackle this growing need. However indoor navigation relies on a plethora of conditions which are difficult to satisfy. The WiFi RSSI Fingerprint based method suffers from it's reliance on a large dataset of recorded fingerprint data which is both difficult to load and difficult to process.

This paper aims to mitigate and circumvent these issues with the use Named Data Networking, a promising new network architecture. NDN provides various advantages over the TCP/IP Protocol. One of the advantages being it's data oriented nature, where it doesn't depend on the source of the data rather the data itself. Another benefit it enjoys is it's ability to cache data on routers themselves. This approach mitigates or circumvents many of the issues associated with RSSI Fingerprinting Based locating and also navigation techniques. Maps of floors can be cached on an NDN router for future use, Location detection task results can be cached on the NDN router and even navigation task results can be cached on the router as well. Our result shows that by using NDN, the existing system can be more efficient. With the efficiency and performance increment from our NDN based system, we believe that now it will be possible to implement Wi-Fi RSSI fingerprinting based indoor positioning and navigation system in larger indoor areas where previously it was not feasible

Not only the performance but also ease of implementation is also a factor. For the solution to be viable the implementation needs to be affordable and easy to maintain. While NDN hardware is not yet widely available it is expected to become popular soon, Our implementation only relies NDN Routers, A barometric sensor for reference during floor detection and a server for calculations and algorithms when the data is not already cached on content stores.

The obtained performance results are promising. Our results are on par with TCP/IP performance when the data needs to be fetched from the servers, whereas it outperforms TCP/IP, sometimes by hundreds of milliseconds, when the data is available on a content store. And this gap will only increase the more hops are added to TCP/IP routers.

This approach may be applied to other public locations as well. As this relies on Wi-Fi access points it can be set up with minimal investment in the future with NDN routers. Candidate locations could be Bus, Train stations, Public galleries, Movie Theatres, Convention Grounds etc

Currently our system suffers from some limitations. One problem is that all devices may not have barometer sensors. This is not a major problem as MEMS based barometer sensors are getting increasingly cheaper and thus are almost universally available on most smartphones. A larger concern is the relative infancy of NDN technology itself. Our implementation is completely reliant on NDN Routers, but these were simulated over TCP/IP networks. True NDN Routers may not be available commercially in the recent future.

Further work can be applied to many aspects. Some candidates can be finding an alternate way for floor detection instead of barometric sensors, dynamically select access points, or even employing a method other than RSSI Fingerprinting. However, our current focus is to restructure the workflow to increase overlap of data requests so more interest packets match data on the content stores more frequently, thus improving performance even further.

Bibliography

[1] B. W. Parkinson, P. Enge, P. Axelrad, and J. J. Spilker Jr, *Global positioning system: Theory and applications, Volume II.* American Institute of Aeronautics and Astronautics, 1996.

[2] J. Werb and C. Lanzl, “Designing a positioning system for finding things and people indoors”, *IEEE spectrum*, vol. 35, no. 9, pp. 71–78, 1998.

[3] C. Bauer, “On the (in-) accuracy of gps measures of smartphones: A study of running tracking applications”, in *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, ACM, 2013, p. 335.

[4] R. Mautz, “Indoor positioning technologies”, 2012.

[5] L. Petrou, G. Larkou, C. Laoudias, D. Zeinalipour-Yazti, and C. G. Panayiotou, “Demonstration abstract: Crowdsourced indoor localization and navigation with anyplace”, in *Proceedings of the 13th international symposium on Information processing in sensor networks*, IEEE Press, 2014, pp. 331–332.

[6] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz, “Wireless rssi fingerprinting localization”, *Signal Processing*, vol. 131, pp. 235–244, 2017.

[7] P.-Y. Gilliéron, D. Büchel, I. Spassov, and B. Merminod, “Indoor navigation performance analysis”, in *Proceedings of the 8th European Navigation Conference GNSS*, 2004.

[8] V. Zeimpekis, G. M. Giaglis, and G. Lekakos, “A taxonomy of indoor and outdoor positioning techniques for mobile location services”, *ACM SIGecom Exchanges*, vol. 3, no. 4, pp. 19–27, 2002.

[9] R. WATSON, G. LACHAPELLE, R. KLUKAS, S. TURUNEN, S. PIETILÄ, and I. HALIVAARA, “Investigating gps signals indoors with extreme high-sensitivity detection techniques”, *Navigation*, vol. 52, no. 4, pp. 199–213, 2005. DOI: 10.1002/j.2161-4296.2005.tb00363.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2161-4296.2005.tb00363.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2161-4296.2005.tb00363.x.

[10] J. B.-Y. Tsui, *Fundamentals of global positioning system receivers: a software approach.* John Wiley & Sons, 2005, vol. 173.

[11] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.

[12] S. Gezici, “A survey on wireless position estimation”, *Wireless personal communications*, vol. 44, no. 3, pp. 263–282, 2008.

[13] Y. J. Guo, *Advances in mobile radio access networks*. Artech House, 2004.

[14] J. Koo and H. Cha, “Localizing wifi access points using signal strength”, *IEEE Communications letters*, vol. 15, no. 2, pp. 187–189, 2010.

[15] C.-H. Lim, Y. Wan, B.-P. Ng, and C.-M. S. See, “A real-time indoor wifi localization system utilizing smart antennas”, *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 618–622, 2007.

[16] C. Gomez, J. Oller, and J. Paradells, “Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology”, *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, 2012.

[17] E. Dahlgren and H. Mahmood, “Evaluation of indoor positioning based on bluetooth smart technology”, *Master of Science Thesis in the Programme Computer Systems and Networks*, 2014.

[18] S. Feldmann, K. Kyamakya, A. Zapater, and Z. Lue, “An indoor bluetooth-based positioning system: Concept, implementation and experimental evaluation.”, in *International Conference on Wireless Networks*, vol. 272, 2003.

[19] J. Randall, O. Amft, J. Bohn, and M. Burri, “Luxtrace: Indoor positioning using building illumination”, *Personal and ubiquitous computing*, vol. 11, no. 6, pp. 417–428, 2007.

[20] F. Fraundorfer and D. Scaramuzza, “Visual odometry: Part ii: Matching, robustness, optimization, and applications”, *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[21] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg, “Indoor positioning and navigation with camera phones”, *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 22–31, 2009.

[22] Y. Nakashima, R. Kaneto, and N. Babaguchi, “Indoor positioning system using digital audio watermarking”, *IEICE TRANSACTIONS on Information and Systems*, vol. 94, no. 11, pp. 2201–2211, 2011.

[23] H. Hellmers, A. Norrdine, J. Blankenbach, and A. Eichhorn, “An imu/magnetometer-based indoor positioning system using kalman filtering”, in *International Conference on Indoor Positioning and Indoor Navigation*, IEEE, 2013, pp. 1–9.

[24] P. Kriz, F. Maly, and T. Kozel, “Improving indoor localization using bluetooth low energy beacons”, *Mobile Information Systems*, vol. 2016, 2016.

[25] H.-S. Kim and J.-S. Choi, “Advanced indoor localization using ultrasonic sensor and digital compass”, in *2008 International Conference on Control, Automation and Systems*, IEEE, 2008, pp. 223–226.

[26] O. Costilla-Reyes and K. Namuduri, “Dynamic wi-fi fingerprinting indoor positioning system”, in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2014, pp. 271–280.

[27] A. Disha, “A comparative analysis on indoor positioning techniques and systems”, *International Journal of Engineering Research and Applications*, vol. 3, no. 2, p. 1790, 2013.

[28] M. T. Rashid, M. S. S. Chowdhury, and M. F. Nawal, “Composite indoor localization and positioning system based on wi-fi repeater technology”, in *2016 IEEE Region 10 Conference (TENCON)*, IEEE, 2016, pp. 2130–2133.

[29] Z. Yang, Z. Zhou, and Y. Liu, “From rssi to csi: Indoor localization via channel response”, *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, p. 25, 2013.

[30] F. Massé, A. K. Bourke, J. Chardonnens, A. Paraschiv-Ionescu, and K. Aminian, “Suitability of commercial barometric pressure sensors to distinguish sitting and standing activities for wearable monitoring”, *Medical engineering & physics*, vol. 36, no. 6, pp. 739–744, 2014.

[31] C. Bollmeyer, T. Esemann, H. Gehring, and H. Hellbrück, “Precise indoor altitude estimation based on differential barometric sensing for wireless medical applications”, in *2013 IEEE International Conference on Body Sensor Networks*, IEEE, 2013, pp. 1–6.

[32] G. Retscher, “Location determination in indoor environments for pedestrian navigation”, in *2006 IEEE/ION Position, Location, And Navigation Symposium*, IEEE, 2006, pp. 547–555.

[33] J. Seo, J. G. Lee, and C. G. Park, “Bias suppression of gps measurement in inertial navigation system vertical channel”, in *PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No. 04CH37556)*, IEEE, 2004, pp. 143–147.

[34] A. Sabatini and V. Genovese, “A sensor fusion method for tracking vertical velocity and height based on inertial and barometric altimeter measurements”, *Sensors*, vol. 14, no. 8, pp. 13 324–13 347, 2014.

[35] M. N. Berberan-Santos, E. N. Bodunov, and L. Pogliani, “On the barometric formula”, *American Journal of Physics*, vol. 65, no. 5, pp. 404–412, 1997.

[36] K. W. Kolodziej and J. Hjelm, *Local positioning systems: LBS applications and services*. CRC press, 2017.

[37] D. Zhang, F. Xia, Z. Yang, L. Yao, and W. Zhao, “Localization technologies for indoor human tracking”, in *2010 5th International Conference on Future Information Technology*, IEEE, 2010, pp. 1–6.

[38] F. Dovis, R. Lesca, G. Boiero, and G. Ghinamo, “A test-bed implementation of an acquisition system for indoor positioning”, *GPS solutions*, vol. 14, no. 3, pp. 241–253, 2010.

[39] E. W. Dijkstra *et al.*, “A note on two problems in connexion with graphs”, *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[40] D. Gao, Y. Rao, C. H. Foh, H. Zhang, and A. V. Vasilakos, “Pmndn: Proxy based mobility support approach in mobile ndn environment”, *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 191–203, 2017.

[41] D. Saxena and V. Raychoudhury, “Design and verification of an ndn-based safety-critical application: A case study with smart healthcare”, *ieee transactions on systems, man, and cybernetics: systems*, vol. 49, no. 5, pp. 991–1005, 2017.

[42] K. Lei, S. Zhong, F. Zhu, K. Xu, and H. Zhang, “An ndn iot content distribution model with network coding enhanced forwarding strategy for 5g”, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2725–2735, 2017.

[43] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, *et al.*, "Named data networking", *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[44] A. Afanasyev, I. Moiseenko, L. Zhang, *et al.*, "Ndnsim: Ndn simulator for ns-3", *University of California, Los Angeles, Tech. Rep*, vol. 4, 2012.

[45] S. DiBenedetto, C. Papadopoulos, and D. Massey, "Routing policies in named data networking", in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, ACM, 2011, pp. 38–43.

[46] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking", *ACM SIGCOMM computer communication review*, vol. 42, no. 3, pp. 62–67, 2012.

[47] J.-s. Jeon, Y. Kong, Y. Nam, and K. Yim, "An indoor positioning system using bluetooth rssi with an accelerometer and a barometer on a smartphone", in *2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, IEEE, 2015, pp. 528–531.

[48] X. Shen, Y. Chen, J. Zhang, L. Wang, G. Dai, and T. He, "Barfi: Barometer-aided wi-fi floor localization using crowdsourcing", in *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*, IEEE, 2015, pp. 416–424.

[49] Y. Bai, W. Jia, H. Zhang, Z.-H. Mao, and M. Sun, "Helping the blind to find the floor of destination in multistory buildings using a barometer", in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2013, pp. 4738–4741.

[50] B. Li, B. Harvey, and T. Gallagher, "Using barometers to determine the height for indoor positioning", in *International Conference on Indoor Positioning and Indoor Navigation*, IEEE, 2013, pp. 1–7.

[51] R. Ichikari, L. C. M. Ruiz, M. Kourogi, T. Kurata, T. Kitagawa, and S. Yoshii, "Indoor floor-level detection by collectively decomposing factors of atmospheric pressure", in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2015, pp. 1–11.

[52] T. Willemsen, F. Keller, and H. Sternberg, "Concept for building a mems based indoor localization system", in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2014, pp. 1–10.

[53] W. Liu and S. Chawla, "Class confidence weighted knn algorithms for imbalanced data sets", in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2011, pp. 345–356.

[54] L. E. Peterson, "K-nearest neighbor", *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[55] J. Raikwal and K. Saxena, "Performance evaluation of svm and k-nearest neighbor algorithm over medical data set", *International Journal of Computer Applications*, vol. 50, no. 14, 2012.

[56] M.-S. Choi and B. Jang, "An accurate fingerprinting based indoor positioning algorithm", *International Journal of Applied Engineering Research*, vol. 12, no. 1, pp. 86–90, 2017.

[57] B. Jiang and X. Yao, “Location-based services and gis in perspective”, *Computers, Environment and Urban Systems*, vol. 30, no. 6, pp. 712–725, 2006.

[58] C. Steinfield, “The development of location based services in mobile commerce”, in *E-life after the dot com bust*, Springer, 2004, pp. 177–197.

[59] R. Matos, D. F. Santos, J. E. Sanguino, and A. Rodrigues, “A gps-based mobile coordinated positioning system for firefighting scenarios”, in *2006 Proceedings of the First Mobile Computing and Wireless Communication International Conference*, IEEE, 2006, pp. 209–214.

[60] U. Joo, “Localization methods for location services”, *Korean Institute of Industrial Engineers*, pp. 441–446, 2009.

[61] H. Kim, “Wireless lan-based lbs services”, *Telecommunications Review*, vol. 16, no. 2, pp. 188–202, 2006.

[62] M. Nosrati, R. Karimi, and H. A. Hasanvand, “Investigation of the*(star) search algorithms: Characteristics, methods and approaches”, *World Applied Programming*, vol. 2, no. 4, pp. 251–256, 2012.

[63] X. Liu and D. Gong, “A comparative study of a-star algorithms for search and rescue in perfect maze”, in *2011 International Conference on Electric Information and Control Engineering*, IEEE, 2011, pp. 24–27.

[64] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, “Path planning with modified a star algorithm for a mobile robot”, *Procedia Engineering*, vol. 96, pp. 59–69, 2014.