Relative Node Position Discovery in Wireless Sensor Networks

vorgelegt von

Computer Engineer, M.Sc.
Mustafa Onur Ergin
geb. in Ankara

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Rolf Niedermeier
Gutachter:	Prof. Dr.-Ing. Adam Wolisz
Gutachter:	Prof. Dr. Thiemo Voigt
Gutachter:	Prof. Dr.-Ing. Andreas Willig

Tag der wissenschaftlichen Aussprache: 19. Dezember 2017

Berlin 2018

Abstract

Internet of Things (IoT) systems, including Wireless Sensor Networks (WSNs), are getting integrated into virtually all aspects of life faster than before. These systems are used from agricultural monitoring and actuation to manufacturing plants. Such wireless networks play a critical role in home and health applications as well as security and asset tracking. The knowledge of physical position of the nodes is important for many applications of WSNs, but this information is often not readily available. In the past, a plethora of different solutions has been proposed that focus on recovering the positions of the sensor nodes, often at the cost of high complexity, preconfiguration, training or limited accuracy.

The precision and computational complexity of such "positioning" algorithms is still a big issue. However, there are cases where the objects are placed in one of a few possible predetermined positions, especially indoors. In those cases, the set of potential locations of the objects is limited and computing the relative positions of those objects in relation to each other might be sufficient to determine their real positions and a precise location in the x, y, z coordinates is not necessary.

This thesis focuses on determining the relative positions of nodes in a WSN by utilizing only readily available Received Signal Strength (RSS) information that is provided by the radio chips they are equipped with. Contrary to common belief, the "closeness" information can be extracted with high confidence among one sender and multiple receiver nodes.

An RSS sampling technique for extracting closeness information is introduced as the initial step of position discovery. This technique utilizes the frequency diversity features of the radio modules. Combining the frequency diversity with statistical reasoning allowed us to demonstrate how RSS information can be used for detecting closer nodes to a transmitting node with high confidence, where this information cannot be extracted by connectivity information. The closeness information can then be utilized to discover node positions by having only one or few nodes at known places on a grid-like setting.

For relative node position discovery, two types of grid settings were considered: one-dimensional and two-dimensional. The only prerequisite for introduced position discovery algorithms is the knowledge of one reference node at one head of a one-dimensional grid and the knowledge of two reference nodes at two corners of a two-dimensional grid.

In this study, a successful position discovery is defined as mapping all nodes to the cells of the grid perfectly, otherwise a result is considered unsuccessful even when only two nodes are mapped to swapped cells. The proposed techniques result in up to 100 % successful position discovery in the repeated real-world experiments and in the simulations.

For self-detecting whether a run of position discovery was successful, reliability analyses are developed. Using these analyses each result is mapped into one of the $high$, $medium$ or low reliability categories. Over 99 % of the results that are assigned to $high$ reliability category were perfectly correct and the unsuccessful computations could be assigned to the remaining categories.

It has been discussed that the suggested frame of procedures has significant advantages over other systems that are commonly used for indoor position discovery, such as accuracy, time-complexity and independence from an infrastructure. The results are compared to Multi-Dimensional Scaling (MDS) and fingerprinting-based systems.

Zusammenfassung

Die Geschwindigkeit, mit der sich Internet-of-Things (IoT) Systeme, wie z.B. kabellose Sensornetzwerke (Wireless Sensor Networks - WSN) in nahezu allen Bereich des täglichen Lebens ausbreiten steigt rasant an. In der Landwirtschaft und der Industriellen Fertigung werden durch solche Systeme zunehmend Monitoring Anwendungen realisiert. Smart-Home und Gesundheitsapplikationen werden durch sie überhaupt erst möglich und auch in den Bereichen Sicherheit und Asset-Management spielen sie eine immer wichtigere Rolle. In vielen dieser Anwendung sind die physikalischen Positionierungsdaten einzelner Sensorknoten von entscheidender Bedeutung, obwohl sie nicht immer einfach abzuleiten sind. Darum wurden in der Vergangenheit eine Vielzeit unterschiedlicher Mechanismen entwickelt, die - oftmals mit hohem Aufwand und Trainingsbedarf, hoher Komplexität, aber geringer Genauigkeit - die Positionierung der im System beteiligten Knoten ermitteln können.

Die mangelnde Genauigkeit und der hohe Aufwand, um die Positionierung von Sensorknoten in einem System zu bestimmen sind noch immer eine der größten Herausforderungen beim Design von Sensornetzwerken. Bei bestimmten Anwendungen, insbesondere im Indoor-Bereich, ist eine komplexe Berechnung hingegen gar nicht nötig, da sich die Knoten lediglich über eine limitierte Anzahl vordefinierter Positionen verteilen. In solchen Fällen kann es ausreichen, die relative Positionierung der Knoten zueinander zu bestimmen, während die genaue Bestimmung der x, y, z-Raumkoordinaten der Knoten nicht nötig ist.

Die vorliegende Arbeit beschäftigt sich mit der Bestimmung relativer Positionierungs-Koordinaten unter Sensorknoten eines kabellosen Sensornetzwerks durch die Nutzung der empfangenen Signalstärke (Received Signal Strength - RSS). Alle heute üblichen WSN-Chipsets liefern bereits einen entsprechenden Indikator (den RSSI). Im Widerspruch zur vorherrschenden Meinung, können Informationen zum Abstand zweier Knoten mit hoher Zuverlässigkeit von diesem Wert abgeleitet werden, selbst in Szenarios mit nur einem Sender und mehreren Empfängern.

Als initialen Schritt zur Ableitung von Positionierungsdaten aus dem RSSI wird hier ein eine RSS Sampling Methode präsentiert, die sich auf Frequenzdiversitätseigenschaften genutzter Funkmodule stützt. Die Kombination aus dieser Diversitätsbetrachtung und statistischen Argumentationsketten erlaubt es uns, mit hoher Zuverlässigkeit genaue Aussagen über die Nähe verschiedener Knoten zueinander zu machen, was über die reine Analyse von Verbindungsinformationen nicht gelingen kann. Diese Information der Nähe einzelner Knoten zueinander wird in einem zweiten Schritt dann genutzt, um die Lokalisierung auf einem Gitter (Grid) möglicher Positionen zu bestimmen.

Zwei verschiedene Arten von Grids werden in dieser Arbeit betrachtet: eindimensionale sowie zweidimensionale Versionen. Einzige Voraussetzung für jegliches Setup ist dabei die Kenntnis über die Positionierung eines Referenzknotens am Kopf- oder Endpunkt des eindimensionalen Grids, bzw. zwei Referenzknoten in zwei Ecken des zweidimensionalen Grids.

Wir gehen in dieser Arbeit davon aus, dass ein erfolgreicher Positionieringdurchgang die Grid-Positionierung aller Knoten richtig bestimmt. Bei jeder Abweichung, selbst wenn nur zwei Knoten auf vertauschten Plätzen lokalisiert wurden, wird der Positionierungsversuch als nicht erfolgreich gewertet. Die hier vorgestellten Techniken zeigen Erfolgsraten von bis zu 100 % in verschiedentlich wiederholten Feldversuchen und Simulationen.

Um systemintern entscheiden zu können, ob ein Positionierungsversuch erfolgreich durchgeführt werden konnte, werden die Systeme zusätzlich mit einem Zuverlässigkeitsanalyse-Tool ausgestattet, das jede erfolgte Positionierung in eine der Zuverlässigkeitskategorien $high$, $medium$ und low einstuft. Dabei waren über 99 % der Versuche, die mit $high$ bewertet wurden tatsächlich erfolgreich.

Zusammenfassend kann festgehalten werden, dass die hier vorgestellten Mechanismen signifikante Vorteile gegenüber gängigen Methoden zur relativen Positionierung von Sensorknoten im Indoor-Bereich haben, dazu zählen die hohe Genauigkeit, die geringe Komplexität und Berechnungskosten sowie die Unabhängigkeit von einer Netzinfrastruktur. Insbesondere vergleichen wir die hier vorgestellten Mechanismen mit dem Multi-Dimensional-Scaling (MDS) Ansatz sowie mit auf Fingerprinting basierenden Systemen.

Acknowledgments

This work has greatly benefited from knowledge exchange and discussions with several scientists. For acknowledging their support, I preferred to use the pronoun "we" instead of "I" throughout my thesis.

First of all, I would like to thank my advisor *Prof. Adam Wolisz* for welcoming me in the Telecommunication Networks Group, and providing support for many years, which resulted in this thesis. The technical discussions, pushing the boundaries of human imagination for advancing science, as well as the inspiring conversations that we have had with *Prof. Wolisz* has been a unique experience, for which I would like to express my gratitude once more. I would also like to thank *Prof. Andreas Willig* and *Prof. Thiemo Voigt* for having accepted reviewing my dissertation and for having travelled all the way from New Zealand and Sweden respectively to attend my defense. I also thank *Prof. Rolf Niedermeier* for having chaired my defense.

I am deeply grateful to *Dr. Vlado Handziski* for the whole time I worked on my research. He was always encouraging and he was always ready to discuss and advise. I would like to acknowledge his help once again for all the long nights of work as he supported me in any way he could, be it fixing the figures or editing literature surveys or proofreading, of course in addition to the intellectual support.

The journey through a doctoral study is never easy. But if you are lucky, you have people around who make this journey much more enjoyable and I had that luck. I would like to explicitly thank *Dr. Mathias Bohge* for making Germany a better place for me and handing me that book (just to have a look at), which eventually inspired me for utilizing radio frequency diversity. *Tacettin Ayar* is a wonderful friend and he has been a great colleague to me, for which I am specially thankful. And of course *Dr. Jan Hauer*, my co-philosopher! In addition to your intellectual support, thank you very much for all the deep-philosophical discussions that we had, whether during a coffee'n'cake break, or during a sea-food dinner or at a jazz concert.

Furthermore, I would like to express my special gratitudes to all of my colleagues at the university for all the good time that we have had: *Petra Hutt*, *Georgios Ainatzes*, *Sven Spuida*, *Marina Rodríguez Aliberas*, *Manoj Rege Ravindra*, *Hieu Le*, *Thomas Menzel*, *Niels Karowski*, *Dr. Sven Wiethölter*, *Dr. Osama Khader*, *Dr. Konstantin Miller*, *Dr. Arash Behboodi*, and by no means is this a complete, nor an ordered list. Thank you folks! I am also greatly thankful to my new colleagues at the Fleetboard's Innovation Hub for all their mental support at the final phase of writing my dissertation and to all the colleagues and friends who supported me with the proofreading of my thesis!

Finally, I would like to thank my girlfriend and my whole family for encouraging and supporting me patiently throughout this journey.

Contents

List of Figures

List of Tables

Acronyms

AoA Angle of Arrival

AP Access Point

BLE Bluetooth Low Energy

CPU Central Processing Unit

GNSS Global Navigation Satellite System

GPS Global Positioning System

IEEE Institute of Electrical and Electronics Engineers

IoT Internet of Things

LBS Location Based Services

LSN Linear Sensor Networks

LoS Line-of-Sight

LR-WPAN Low-Rate Wireless Personal Area Network

MDS Multi-Dimensional Scaling

NLoS Non-Line-of-Sight

OSI Open Systems Interconnection

PbL Pick-by-Light

PNSD Probabilistic Node Sequence Discovery

PRR Packet Reception Rate

QR code Quick Response Code

RFID Radio-Frequency Identification

RF Radio Frequency

RSS Received Signal Strength

RSSI Received Signal Strength Indication

TDoA Time Difference of Arrival

TKN Telecommunication Networks group at Technical University Berlin

ToA Time of Arrival

ToF Time of Flight

TWIST TKN Wireless Indoor Sensor Network Testbed

USB Universal Serial Bus

USNO U. S. Naval Observatory

UWB Ultra-Wideband

WBAN Wireless Body Area Network

WGS World Geodetic System

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Networks

WSN Wireless Sensor Network

Chapter 1

Introduction

Location has been one of the most important information, if not the most. Everything in the nature requires this information; plants detect the position of the sun, animals mark and find their territories, humans determine it proactively. Discovering one's own or others' location is crucial for survival. We humans have developed numerous ways to compute the location information. It began with observing the positions of big landmarks, such as mountains and water bodies. The sun was of big help, hence the directions were defined: East, West, North and South. Later our ancestors created their own landmarks, such as drawings and carvings. Once the early humans started to migrate from one place to another, they realized they could use particular stars as reference points to figure out where they were.

With time, more precise measurement systems were created. The distances were first measured in relative units, like feet and arm lengths, then in more standardized measures such as meters. Along the way, imaginary landmarks were defined instead of physical ones, namely parallels and meridians, which would help one communicate their location on the Earth.

In the middle ages, determining their own geographical location was (and still is) crucial for the sailors. Some used polarization with sunstones to find their positions with respect to the sun. Later the magnetic compass was developed to compute their relative bearings. Next, they developed dead reckoning methods to compute their current locations some time after they left their last-known positions. Then they calculated the absolute locations of landmarks, or they built their own landmarks, sea marks and lighthouses at known locations to develop piloting methods with their compasses. This enabled the development of triangulation methods for calculating location fixes, with respect to a few known points.

Fast-forwarding to the modern age; artificial satellites were built and placed on one of Earth's orbits to form the Global Positioning System (GPS), which have been helping us to pinpoint our global location with down to five meter precision [4]. GPS consists of twenty-four satellites that sit on six orbits of the Earth. Users scan for signals from the visible GPS satellites and process the arrival time delays of the signals from four of the satellites to compute their own latitude, longitude and altitude relative to the World Geodetic System (WGS) and time relative to the U. S. Naval Observatory (USNO) time. With development of mobile communication systems, we integrated the GPS into our cellular telephony devices. Today, techniques to deduce the geographic or relative locations of mobile or stationary objects are defined under the term

"positioning", which is also referred as "localization". Some researchers argue that these two terms differ in the absoluteness or relativeness of the calculated position information [5], but most often they are used interchangeably.

Precise determination of location has enabled new services, such as geographic tracking of objects or humans, assisted navigation or determining presence. These services very often utilize GPS localization, which works only when direct Line-of-Sight (LoS) to at least four satellites is possible. This limitation encourages the scientific community to search for alternative positioning systems that would also work where such satellite visibility is not present.

Alternative systems have been developed for the cases when the GPS is not available; such as vision based systems with video cameras, ranging using cellular and other Radio Frequency (RF) signals or connectivity based positioning systems [6]. However, these systems, which strictly depend on an outdoor infrastructure, do not work inside of buildings reliably. Furthermore, camera-based systems require certain types of light conditions, RF-based ranging requires LoS and indoor deployments are often too dense for connectivity-based systems. Hence the research in localization is split into two main categories: *outdoor positioning* and *indoor positioning*. This work finds its place in the latter one.

1.1 Indoor Positioning

1.1.1 Why is it needed

Many of us spend most of their time and keep most of their belongings indoors. Very often we interact with other people indoors, use indoor appliances such as copiers and coffee machines, use phones and computers that are attached to indoor infrastructures. Also, when we enter a new building, finding certain things or other people can be challenging, due to the lack of known landmarks. We need to understand floorplans and orient ourselves accordingly, if they are provided to us. Navigating inside a new building is a time consuming, non-trivial task.

Another important aspect is business. Most of the time shopping is done inside buildings, like a mall or a gallery. Commercial goods are stored behind the walls. Packaging and storing of organic and non-organic items and other inventory processing happen inside buildings. These necessities add significant value to development of indoor positioning systems.

There is also an increasing demand for Location Based Services (LBS) [7]. For example, in a museum a hand-held device can provide us specific information about an artifact, before which we are standing. Customized announcements can be made, or individualized messages can be delivered to people at specific locations without necessarily knowing their identities, such as for advertising purposes. Likewise, shops can provide time and location specific offers to the people who visit them. In a grocery store, indoor navigation systems can help us collect the items on our shopping list faster.

Very recently, a smartphone game, which rewards its players based on outdoor navigating activities, has been a big success [8]. A system, which enables indoor navigation, may help indoor versions of that particular game to be developed, or may help developers create other such games. These systems can even be integrated into existing games to provide new features.

Location information has also a big role in the Industry 4.0 realm [7,9]. At each stage of the

production cycle, this information will enable easier access to facility components, maintenance workers and machinery that move and interact with each other, as well as humans. The precision of mobility in robotics can also be increased. Collecting and processing of data can be improved.

The need for indoor positioning systems are beyond industrial and convenience domains. In cases of emergencies, it can save lives of first responders, such as firefighters, by helping them to navigate towards their targets and then —just as importantly— aiding them navigate their ways out from a collapsing building.

These use cases and applications are only limited with human imagination. The need for determining positions of people and objects is obvious and often vital. The research, like in this work, is currently continuing at a global scale and new ideas and systems are emerging almost daily.

1.1.2 Where can it be used: Some Practical Examples

We perform indoor position discovery more often than we think, though manually. For example when we enter a plane's fuselage, we need to locate our seats by comparing the number on our boarding cards to the row and seat numbers. We do the same at a theater or an opera. It is not only the guests who need to find the locations of these items, but also the people who are maintaining them. Some objects require regular maintenance and they are often removed from and reinstalled to their places. It is important for the maintainers to figure out where one particular seat is located among all others for various reasons, such as providing customer specific service, safety or post-service data analysis. Easy retrieval of position information is also critical for our peers in the society who are hard of hearing or seeing.

The information on the location of the seats can be also used to facilitate safety practices. If we know which seat is where in a plane fuselage, the seat belt checks can be done remotely with the wireless capable electronics attached to the seats, which will decrease operational time overhead per flight. After a tragic incident of a plane crush, the seats that are recovered post-accident can be analysed to increase flight safety, if the configuration of the seats can be identified prior to the beginning of a flight, especially if the seats and their locations change in between flights.

Many commercial products are stored indoors in warehouses that contain big numbers of similar looking containers. The containers may need to be individually identified, which is a time consuming process. Manual search can be replaced by an automatized indoor positioning technique. Today, systems that utilize self-navigating robots for fetching items in the storage rooms exist [10]. The operation of these robots depends on a database, in which the positions of all the items are pre-recorded.

These needs and principles can be applied to many different scenarios, from hospital beds to classroom seats or to devices in a call center, where many of the similar looking items exist. The fast identification of object positions can be crucial. Particularly, the advancements in Industry 4.0 create new application areas and needs for indoor positioning systems, such as inside mines or production plants.

1.1.3 Challenges in Indoor Positioning Systems

Positioning techniques that are designed for outdoor use utilize landmarks or signals from other sources. Outdoor positioning mostly relies on either Time of Arrival (ToA) measurements or signal attenuation calculations at LoS. Implementing such systems is very challenging indoors because the LoS is often hard to achieve due to the partitioning of buildings. Also the level of multipath fading is higher inside buildings. Most common challenges that arise with indoor positioning systems are listed below. It must be noted that the way these challenges is handled might vary across different systems and not all of these challenges may be relevant for each type of indoor positioning system.

Unavailability of Satellite-based Systems

Due to its availability, applicability and simplicity at the industrial and end-user systems, the GPS has become the de facto standard for outdoor positioning globally. Among other satellite-based systems GLONASS (Global Orbiting Navigation Satellite System) [11], Galileo [12] and Beidou [13] are popular in Russia, Europe and China respectively. These Global Navigation Satellite Systems (GNSSs) are integrated in land, nautical and air vehicles, in mobile telephony systems, in security and tracking devices and they can even be attached to wild animals [14]. The satellite signals, however, can attenuate to the levels that they become useless when they penetrate through the walls and ceilings of most buildings. Because of this reason indoor positioning systems rely on other passive or active RF or optical systems, and they often require specific infrastructures such as access points, Radio-Frequency Identification (RFID) tags, Quick Response Codes (QR codes) or sound emitters.

Accuracy

The equipment that indoor positioning systems rely on are often imprecise. Utilizing Wi-Fi Access Points (APs) is common but they are known to provide inaccurate and inconsistent results. For example 5 meter positioning accuracy may be sufficient for most outdoor positioning systems but such precision can be useless indoors, output of which might appear in a different room of a building. Therefore, special hardware designed for more accurate indoor positioning is required, which increases the costs substantially. When we consider the potential density of indoor objects, whose positions are desired to be determined, the installation costs can rise very quickly.

Need for infrastructure

Many of the indoor positioning systems rely on the existence of an infrastructure. The infrastructure may already exist in the building, for example Access Points for Wi-Fi network connectivity can be used for fingerprinting systems. Installation of extra hardware, such as ultra-sound transmitters, QR codes or Bluetooth Low Energy (BLE) beacons, may also be needed. Two big problems arise with equipping a building with such infrastructure for an indoor positioning system:

1. Retrofitting an infrastructure costs time and money.
2. The infrastructure cannot be easily moved or changed, otherwise the system will need to be reconfigured (e.g. retrained) accordingly.

Need for additional hardware attachments

Particularly the systems that rely on ranging between devices require additional wireless interfaces, or attachments, like ultra-sound and chirp-based ranging systems, on the devices that need to be positioned. Additional need for hardware implicitly increases the setup and operational costs. Maintenance of such systems may become infeasible by time due to technological developments and supply of the special hardware may become scarce.

Need for training

Fingerprinting systems are very popular, yet they are the most demanding ones in terms of need for preconfiguration. They require extensive offline training phases which take a lot of time and manual labor. Some ranging systems require their parameters to be adjusted to suit their environment. There are also systems that require the precise floorplan of their environment, which may not always exist and need to be produced.

The mechanisms that are developed in this thesis handle all these challenges successfully.

1.2 Goals of the Thesis

In outdoor situations, we are mostly interested in finding out locations relative to the Earth's coordinate system, therefore we use global look-up maps for matching our locations to addresses or known locations. In contrast, each building has its own coordinate system and each building is different in size and shape. Indoor environments are also very different from outdoors in terms of wireless signal propagation characteristics. Under controlled environments some systems achieve centimeter-grade precision, however they require Line-of-Sight, which is often not possible to have, and multipath is often too high [15]. Indoor positioning starts with the knowledge of the environment that the object to position is in, and the location information is expected to be expressed relative to that particular environment.

This assumption motivated this work for creating a solution for the indoor position discovery problem to handle the challenges that are explained in Section 1.1.3 better than the present state-of-the-art systems.

We will approach the indoor positioning problem from a different perspective, which is assigning objects to their potential positions, rather than finding cartesian coordinates of the objects. We have observed that there is a big enough class of applications, in which the sought objects could occupy only some predetermined positions, which could be represented as a regular grid. For example, staff of a call-center use only one of suitably equipped desks during their work, experimenters in a lab use one of the laboratory benches, patients waiting for a radiological screening use one of few clothes-changing cabins, passengers of an airplane occupy

one of the seats in predefined positions inside the fuselage et cetera. In fact, for such applications the location problem can be reduced to the problem of determining the mapping between a set of objects and a set of candidate positions. Such a mapping problem can be addressed in one-dimensional, two-dimensional or three-dimensional settings. In the scope of this work, one-dimensional and two-dimensional settings will be addressed.

A newly emerging sub-class of WSNs is the Linear Sensor Networks (LSN), in which the nodes of the network are placed in a linear shape in space [16]. These kinds of networks involve nodes that perform sensing and measurement in conditions where the geographical multi-dimensionality is not relevant. For example above and under ground pipeline control and sensing, under water pipeline sensing, railroads and subway monitoring, powerline monitoring, border and coast monitoring and roadway driver alerting type of applications [17]. The special case of these settings allow and often require special techniques for their common problems, such as routing efficiency, network reliability, security and location management [18]. In this class of applications, exact distance between the nodes are not needed but their positions relative to each other is the important information. In many industrial settings, such as warehouse management, the merchandise is stored in strictly regular grids. Pick-by-Light (PbL) is a modern and efficient way of speedy access to those items, however the installation and maintenance costs are high [19]. A self-discovering multi-hop wireless network of PbL setup will also benefit greatly by a configuration-free position discovery system.

Therefore, the object position discovery problem that we are tackling is an object-to-position mapping problem, rather than computing cartesian coordinates. This problem of indoor positioning field assumes that the candidate positions are predefined, such as the cubicles of a working space, shelves of a warehouse or seats in an auditorium. The objects, whose positions need to be recovered, are assumed to be enhanced with wireless communication capabilities. These objects may or may not have Line-of-Sight to each other.

Throughout this thesis, the objects, whose positions need to be discovered, are assumed to be part of a WSN and we will be referring to them as *"nodes"* when we explain our approaches. The position discovery algorithms that are proposed in this work produce a mapping between a set of N nodes and a set of N positions. We use a binary metric for measuring their performances: the mapping of nodes to positions is either perfectly correct or it is not correct. A $correct$ (or $true$ or $success$) result means that N nodes could be placed in N positions without any error. An $incorrect$ (or $false$ or $fail$) result means the assignment of nodes to positions was not perfectly correct.

The system that is designed in this work is free from an infrastructure and a training phase. We have designed and tested our algorithms for the common communication radio modules for WSNs, therefore they are free from special ranging or other dedicated positioning hardware attachments. We assumed that the proposed systems will work inside any type of indoor environments, and hence an aid from GNSS is considered to be absent. We targeted to achieve an indoor positioning system that creates an assignment of objects to their predefined potential positions. Therefore we are interested in not an accuracy metric that is expressed in a coordinate system, but a precise assignment of the nodes to their positions.

The mechanisms that one can utilize through common communication radios chip are: presence information, ToA, Packet Reception Rate (PRR) and RSS. Presence information re-

quires installing properly labelled radio tags to predetermined locations and it is used to alert whenever an object with a compatible radio interface enters the proximity. Such a system is out of our interest as it requires a costly infrastructure. Time of Arrival based positioning systems require fast Central Processing Units (CPUs) that can measure the Time of Flight. RF signals travel at the speed of light, and one meter accuracy requires approximately 3 ns time resolution. Most WSN systems contain low-energy CPUs that cannot achieve such a resolution [20]. In close distances that we are interested in indoors, such as 1 - 3 meters between the wireless terminals, the PRR between a sender and multiple receivers is often indistinguishable [21]. Therefore, rather than ranging or static tag-based approaches, utilizing RSS for indoor position discovery is investigated in this work.

It is common knowledge that RSS cannot be converted to distance indoors reliably [7, 22–26]. Many RSS based systems require extensive training phases, which cost time, power and money [27]. Such training phases are also very prone to errors and need to be re-done if the building structure or layout changes [28]. Even though RSS is thought to be an unreliable tool to discover device positions in indoor settings, it will be shown in this thesis that it can be leveraged to detect closeness information among peers of multichannel capable WSN nodes. Therefore, use of extra hardware for ranging or preconfiguration is avoided.

It will be assumed that the nodes are deployed in a virtual grid, each cell of which contains only one node. Detection of the positions of nodes relative to each other requires initial position information of one node in single-dimensional settings (in LSN systems) and initial position information of two nodes in two-dimensional WSN deployments. The nodes with known initial positions are called *reference node*s and they are positioned in one end/corner cell of the grid setting. These reference nodes decide which other nodes might be occupying the cells adjacent to theirs. Then these nodes that are newly assigned to their cells decide which other nodes might be occupying the cells adjacent to theirs and this system runs until each node is assigned to one cell/position. We have analysed two types of assignment methods: an easy-to-implemet iterative method and a more precise probabilistic method. The iterative assignment of nodes to positions produces a definitive node-to-position matching for a position and then proceeds to the next position, which ends with a final position classification of nodes. The probabilistic method produces one or more node-to-position matchings with different probabilities and chooses one of the many possible mappings that has the highest probability.

For validating our hypotheses, we have used real-world measurements, most of which were performed on the TWIST testbed [29]. We utilized frequency diversity in the measurements and used RSS values in all available channels of the radio chip. Our proposed methods start with all nodes in the system exchanging packets at all available radio channels and then they compare the measured RSS values to one another. Such a measurement system can be seamlessly integrated to the regular packet exchange of a deployed network. The measurements were then used to compute the most likely position mapping of the nodes of the network.

We have also performed an additional level of validation by using a simulator. We believe that this combination of experiments (simulated and real-life) allowed us a more realistic and representative assessment of our results when compared to many recent studies that are evaluated only on one type of environment.

We also defined a second type of quality metric for our results, which we named "reliability". The reliability assessment classifies the output of our algorithms into one of the three categories: *high*, *medium* or *low*. We show that a node positioning result with a reliability level of $high$ is indeed reliable, whereas a result with low quality cannot be trusted and the measurements should repeated.

Reliability level of the results (discovered node positions) is calculated by initially assuming that the result of the position discovery for that experiment is correct. We then repeat the position discovery for some subset of the nodes in a different direction using the same measurement set and compare these two results. Depending on how well these two results match, we assign the reliability level. This allows us to confirm the results and in case of detecting a "low" reliable result, we can suggest using a different set of measurements, i.e. waiting for some minutes and repeating the measurement campaign.

The approaches that are introduced in this work, have proven that the RSS information, coupled with frequency diversity, can be used for precise relative position discovery of wireless nodes. The performance in recognition of node positions in an LSN has been about 100 % success when the wireless medium had normal level of external interference and noise, and the performance stayed over 60 % when the channel conditions were harsher. Of all these computed results, the success rate was 100 % if the reliability level was *high* and the vast majority of the results with *medium* level of reliability was correct. Most of the unsuccessful results (not achieving perfect mapping of nodes to positions) could be classified to *low* level reliability. These results were obtained both from real world experiments and our simulations.

In the two-dimensional case, the success rate was close to 100 % for the configurations reflecting a common indoor office environment. When the modelled environment represented a challenging wireless medium, the success rate stayed above 58 %. Similarly, the results with *high* reliability rank were over 99 % correct and the majority of the fails could be classified into the *low* reliability rank.

In summary, this work investigates indoor positioning problem as a node-to-position mapping problem, which is free from the challenges of a training phase, an infrastructure or extra hardware. It is shown that, for reliable position discovery it is enough that each node sends a number of beacon messages and these messages are used to compute mapping of wireless nodes to their potential positions with high success in a regular grid-like setting.

1.3 Outline

The rest of this dissertation is structured as follows: first the background information on the technologies that are relevant to this thesis is explained and a summary of previous work on related positioning technologies and systems is provided. The following chapters elaborate on proposed methodologies for determining node positions in single-dimensional and two-dimensional cases. In the final part, main contributions of this research is summarized and possible directions for future research are suggested. A more detailed description of each chapter is given below:

- **Chapter 2 – Background** contains an overview of the technologies, which have been utilized in this work.

- **Chapter 3 – Related Work** provides an introduction to indoor positioning solutions in WSN. It constitutes a brief survey of indoor positioning techniques and it discusses why the existing studies do not solve the problem of mapping nodes to potential positions.
- **Chapter 4 – Determining Node Closeness at a Single Channel** shows the evaluation of RSS information for finding comparative proximity information. As an outcome of experiments, first hints for solving the problem of relative position discovery are retrieved.
- **Chapter 5 – One-Dimensional Node Position Discovery** discusses how small changes in the wavelength of a wireless signal can cause variance in the multipath. Varying wavelengths of the propagated signal result in different propagation paths and therefore a receiver will measure different RSS values from the same stationary transmitter for transmissions at different channels. This observation and the algorithms we built around it are then used to deduce the physically closest receiver node from a set of nodes at varying distances. This chapter continues with explaining how to detect the sequence of nodes, which are positioned along a line, by leveraging RSS measurements. Two main techniques are discussed: an iterative approach and a probabilistic approach. Iterative approach is easy to implement and it is lighter in terms of computational complexity, while probabilistic approach provides more accurate results. Computation of the reliability metric is also introduced in this chapter.
- **Chapter 6 – Two-Dimensional Node Position Discovery** elaborates on how the probabilistic node sequence discovery system can be extended to the two-dimensional space. It analyses the case with the minimum number of needed reference nodes (two nodes) and later how to increase accuracy by adding one more reference node. The system in this chapter also contains an adapted version of the reliability assessment algorithm.
- **Chapter 7 – Conclusions** provides a short review of the proposed solutions and a summary of the major contributions of this work. In addition, directions for future research are discussed.

Chapter 2

Background

2.1 Wireless Sensor Networks

In early 2000s, the shape of information generation has changed. The desktop computers, mainframes and laptops used to be almost the only data generators, which obtained their input from humans or other *traditional* computation units, like themselves. The information that they processed was at best indirectly related to their physical environment [30]. With the advancements in microprocessor and radio technologies, smaller sized low-energy devices could be manufactured and they became ready to operate independent of the fixed power and data networks.

This new type of devices quickly became able to collect information from the physical environment that they were deployed in and transmit their data to fixed workstations. Initially they were embedded into bigger workstations and had one task. These semi-independent systems that interact with the environment and provide data to the system they are physically attached to are called *embedded systems*. The impact of the embedded systems is continuously growing and they are becoming more present in everyday life, such as in common home appliances, vehicles like cars, trucks, trains and planes, cameras and even mobile phones.

With the integration of sensors and actuators to embedded systems, they enabled human-to-machine and machine-to-machine interactions. These type of interactions often required them to be physically detached from their hosting systems. Some application scenarios could easily be implemented using wires for communication of sensor and actuator systems to other systems. However, such cabling of rather small devices have significant drawbacks, such as breaking of the wires and high total cost of these wires. Therefore wireless communication between devices quickly became an essential requirement for the implementation of sensing and actuating applications. This new type of networks are called Wireless Sensor Networks (WSNs).

These networks can interact with their environments by sensing and/or controlling physical parameters. The nodes of a WSN can perform individually, as well as collaboratively to accomplish their tasks. These tasks can support many real-world applications, and these applications are only limited by our imagination. Hence WSNs are also a continuous research and engineering subjects. The WSNs cannot be restricted to a single set of requirements, they are flexible. They can contain and connect different types of hardwares and abstractions. They are often battery-powered, so they can be mobile as well as static. Their hardware is often made

of low-energy components, therefore these networks support (and are often designed for) long-time applications; meaning that they are designed to operate for months or even years without physical maintenance. Their tasks are usually simple enough to not require a high-end processor, which makes them cost efficient, as well. As of today (year 2017) a single WSN node can be purchased for less than 20 Euros with processor, wireless radio chip and a few sensors integrated.

Due to their characteristics of supporting a wide range of applications WSN systems are required to be flexible and adaptable. These requirements bring a range of challenges and features, which have been studied for a long time and still are open for advancements. In the books [30] and [31] these features are classified as type of service, quality of service, fault tolerance, lifetime, scalability, wide range of densities, programmability, maintainability, data traffic flow, network topology, indoor or outdoor, automation, context awareness, mobility, node heterogeneity, housing, power awareness, financial feasibility, security, delay tolerance, reliability, time synchronization and localization/positioning. In this thesis, *indoor* and *positioning* features will be on the focus.

Wireless Sensor Networks can be used in almost all aspects of life that one can imagine. Their flexible and infrastructureless nature, combined with benefits of producing information in different environments enable us to deploy them in many places and allow us collect data and perform actuations in anywhere that we want. Having low computational power on the nodes is not a limitation, since their primary task is to generate and relay information that can be used in powerful backend mechanisms.

Considering current research trends and areas, the application scenarios of WSNs can be categorized as, but are not limited to, control and automation, healthcare, environmental monitoring, security and surveillance, logistics, home and office, transportation, tourism, education and training. Position information of the WSN nodes are either essential or beneficial to all these areas of WSN applications.

2.2 Wireless Sensor Network Hardware Platforms

Wireless Sensor Networks have more limitations on the hardware side than most common types of mobile computers, such as celular phones, laptop computers or tablets. These limitations are natural causes of the application requirements that these systems are targeted at. The nodes of such networks need to be cheap, small and equipped with right sensors or actuators. They need to be energy efficient, yet they need to have enough computational power and memory and they must contain low-power communication modules [32].

Most sensor nodes are composed of a controller, memory, power supply, sensors and actuators and a communication module.

Controller or a CPU is required to process data and information, execute operations and connect other components.

Memory is needed to store programs and external data. Data are often stored in a different memory unit than the programs.

Power supply is often batteries. With increased accessibility to energy harvesting mechanisms various technologies, such as solar cells, have been supporting the rechargeable batteries or they have even been replacing traditional batteries. [33]

Sensors and actuators are nodes' interfaces to the physical world. They can read or control the parameters of the real world.

Communication module is necessary for communicating with other nodes and form a network.

These components need to function with taking energy consumption into consideration. Typically, the controller and the communication module switch to sleep mode whenever they are not active. A preprogrammed soft timer is used to coordinate the sleep-wakeup cycle. Recent advances in low-power listening solutions allow nodes to sleep for more than 99 % of the time, while seamlessly allowing the network to perform its regular operations like routing [34]. In case of non-cyclical, event-based sleep-wakeup mechanisms, sensors trigger that behavior depending on the measurements they take from the physical environment, such as when the carbon dioxide values exceed tolerable limits.

2.2.1 Examples of WSN nodes

There are many hardware platforms that are low-power, wireless capable and can host sensors and actuators. While many of them have been developed by research institutes for research purposes, there is an upward trend in targeting industrial applications. Table 2.1 provides a representative list of WSN hardware platforms.

Any of these hardware platforms could have been used in the real-world experimentation of this thesis as a tool for taking RSS measurements. Their radio modules are programmable and they can change their radio frequency channels during the runtime. All of these platforms are programmable with the commonly used WSN operating systems TinyOS 2.1 [50] or Contiki [51]. Because of popularity and ease-of-use factors we have selected the *TelosB/TmoteSky* platform running with TinyOS 2.1 in our measurements and experimentation.

These platforms are only a small subset of Wireless Sensor Network platforms. There are many other products, designs and even attempts, such as Ember nodes [52], FireFly nodes [53], G-nodes [54], Fleck nodes [55], Particles [56], SquidBee [57] or WeBee [58]. Even the popular IoT system Raspberry PI can be used as a WSN platform [59]. After all, a Wireless Sensor Network node is a component of a wireless network that is capable of some processing and collecting sensor information and it can communicate with other nodes within its geographical proximity wirelessly. Due to the ever increasing popularity of IoT systems and WSNs, new platforms continue to emerge for both industrial applications and academic research purposes.

Table 2.1: A Representative List of WSN Hardware Platforms

Name	**Controller**	**Memory**	**Radio Chip**	**Connections**
EYES [35]	MSP 430	48 KB Flash, 10 KB RAM	TDA5250 (868.35 MHz)	USB, ADC
BT Node [36]	Atmel ATmega 128L	128 KB Flash, 64+180 Kb RAM	CC1000	ISP, UART, SPI, I2C, GPIO and ADC
Scatterweb [37]	ARM7-TDMI	512 KB FLash, 98 KB RAM	CC1100	USB, SPI, I2C, RS232, CAN, SD-Card
Imote2 [38]	Intel PXA271 XScale	256 Kb SRAM, 32 MB Flash, 32 MB SDRAM	CC2420	UART, SPI, I2C, SDIO, GPIO
MicaZ [39]	Atmel ATmega128L	4 Kb RAM , 128 KB Flash	CC2420	USB, SPI, I2C, ADC
IRIS [40]	Atmel ATmega1281	128 Kb Flash, 8 KB RAM	IEEE 802.15.4	I2C, SPI, UART
Mulle [41]	ARM Cortex-M4	16 MB Flash, 4 KB RAM	AT86RF212B	60-pin I/O connector
TmoteSky [42] TelosB [43]	MSP 430	10 KB SRAM, 48k Flash and 1 MB RAM	CC2420	USB, SPI, I2C, ADC
Kmote [44]	MSP 430	10 KB SRAM, 48 KB Flash and 1 MB RAM	CC2420	USB, SPI, I2C, ADC
Shimmer [45]	MSP 430	10 KB SRAM, 48 KB Flash and 1 MB RAM	CC2420	USB, SPI, I2C, ADC, SD-Card
RE-Mote [46]	ARM Cortex-M3	512 KB Flash, 32 KB RAM	CC2538 and CC1200	ADC, SPI, I2C, SD-Card, USB
TI Sensortag [47]	ARM Cortex M3	128 KB Flash, 8 KB + 20 KB SRAM,	CC2650	UART, I2C, GPIO
Seed-Eye [48]	PIC32MX795F512L	512 + 12 KB Flash, 128 KB RAM	MRF24J40MB	ADC, SPI, I2C,CAN,UART, USB
nRF52 DK [49]	ARM Cortex-M4	512 KB ROM, 64 KB RAM	nRF52832	ADC, I2C, SPI, UART

2.3 IEEE 802.15.4 Standard and CC2420

The Institute of Electrical and Electronics Engineers (IEEE) is a professional association and the leading standards development organization, which supports working groups to develop and maintain wireless and wired communication standards, as well as other technical standards like electric power, biomedical technology et cetera. Most known examples of the communication standards that IEEE maintains is IEEE 802.3 (Ethernet) and IEEE 802.11 (Wi-Fi). In Wireless Sensor Networks, one of the varieties of Wireless Personal Area Networks (WPAN) can be used, such as IEEE 802.15.1 (Bluetooth) [60], IEEE 802.15.3 (Ultra-Wideband (UWB)) [61] or IEEE 802.15.6 (Wireless Body Area Network (WBAN)) [62]. In this thesis, the IEEE 802.15.4 is used, which defines the Low-Rate Wireless Personal Area Networks [1] .

The IEEE 802.15.4 standard is developed for low-data-rate control and monitoring applications, for which energy efficiency and low power consumption is essential. It defines the physical layer and the media access control (MAC) layer of the Open Systems Interconnection (OSI) model. Primary purposes of an Low-Rate Wireless Personal Area Network (LR-WPAN) are ease of deployment, reliable data transfer, being low cost, long battery life and flexibility. Two types of devices can participate in an IEEE 802.15.4 network: a full-function device and a reduced-function device. A full-function device can serve as a coordinator to the network, as well as just a regular participant. On contrary, a reduced-function device is not capable of serving as a coordinator and is intended to accomplish very simple, light-weight tasks, such as a light switch or a temperature sensor. Depending on the application requirements, an IEEE 802.15.4 LR-WPAN operates in either a star topology, or a peer-to-peer topology as illustrated in Figure 2.1.

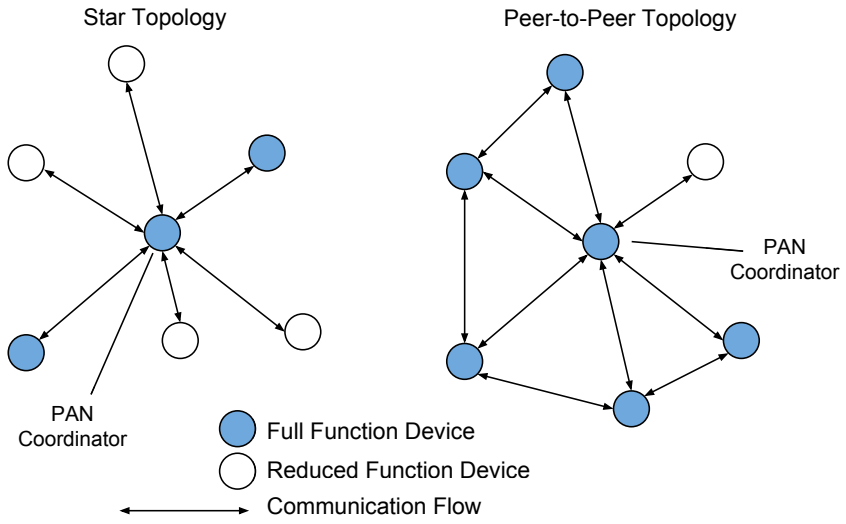


Figure 2.1: Star and peer-to-peer topology examples. (Reproduced from the standard definition [1])

There are three frequency bands that IEEE 802.15.4 operates on: 868 MHz (Europe only),

902 MHz (Americas only) and 2.4 GHz (worldwide). The 868 MHz band has one channel with a bandwidth of 600 kHz and a data rate of 20 kbits/s. 902 MHz band defines 10 channels, each of which is 2 MHz apart and the data rate is 40 kbits/s. At the 2.4 GHz band, IEEE 802.15.4 standard defines 16 channels with a channel bandwidth of 5 MHz and a data rate of 250 kbits/s. In this work, 2.4 GHz band was used and the higher number of channels was essential for our hypotheses.

The hardware that was used in the experiments performed for this thesis featured a radio chip, which is called *CC2420* from Texas Instruments. CC2420 is a 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Tranceiver [63]. It is a single-chip transceiver and designed for low power, low voltage wireless applications. It is suitable for both full-function devices and reduced-function devices with a programmable output power. It has a sensitivity level of -95 dBm and separate transmit and receive buffers. Its output power can be programmable between -25 dBm and 0 dBm with a current consumption of 8.5 mA to 17.4 mA.

2.4 TWIST Testbed

Doing research on WSNs can be challenging. When the deployed number of nodes needs to be more than a few, the material and time costs can rise quickly. The limited resources on the nodes make it impractical for over-the-air reprogramming, which also takes too much time for small and repetitive modifications of the original program.

When we do research and experimentation, we would like to gather as much relevant data as we can and we often want to reach to the nodes individually. TWIST testbed is created at Telecommunication Networks group at Technical University Berlin (TKN) to help with such challenges that the scientists experience [29]. The TWIST testbed has also been our choice for the long-term measurements (lasting several days) of our experiments. Its goal is to support the design, implementation, test and evaluation of WSN applications and protocols. It is based on off-the-shelf hardware and uses open-source software. TWIST is well documented and can be reproduced by any interested organization.

TWIST is a scalable and flexible testbed for indoor deployment of WSNs. It provides basic services like node configuration, network-wide programming, out-of-band extraction of debug data and gathering of application data. It supports co-operation of heterogeneous hardware platforms and active power supply control of the nodes.

TWIST meets following testbed requirements:

Building different system architectures: For example an architecture can be *flat*, where all the nodes are homogeneous, or *segmented*, where some nodes act as gateways.

Easy programming and debugging: Frequent reprogramming is typical for WSN applications, therefore it is essential to be able to update the software on the nodes in parallel. A testbed should also provide support for delivering ”out-of-band” debug messages, such as using a wired back-channel.

On-the-fly re-configuration: The testbed should be able to emulate node-failures or controlled interference.

Testbed management: The testbed must be able to address each node separately through their unique identifiers. It should detect defect nodes and other components individually.

2.4.1 TWIST Architecture

TWIST testbed has six components in its architecture. As illustrated in Figure 2.2, the lowest layer hosts the sensor nodes, which are connected to USB hubs through passive and active USB cables. These hubs are controlled by super nodes, which are connected to the server and the control station at the highest layer.

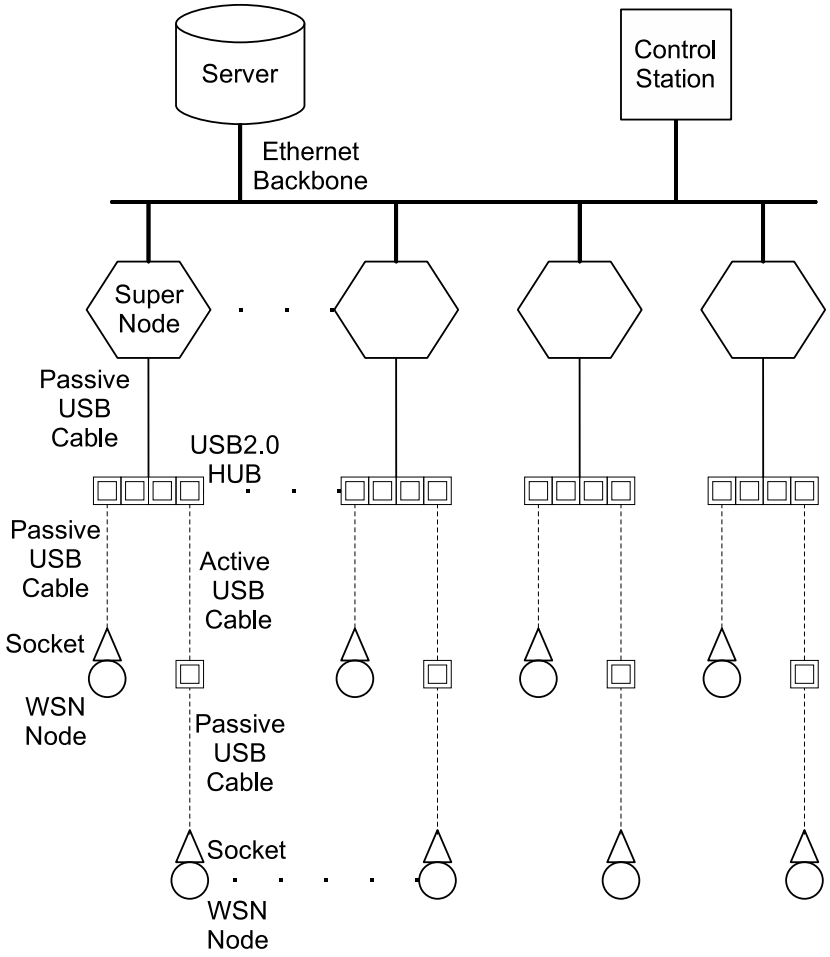


Figure 2.2: Hardware architecture of the TWIST testbed

Sensor nodes

Current deployment of TWIST (as of 2017) contains TelosB [43] and eyesIFX [64] sensor nodes at the lower layer, both of which use their USB interfaces for power, re-programming and serial messaging.

Testbed sockets and USB cabling

The sockets of the testbed are the points where the USB cables are attached to. Each socket has a unique identifier and their exact location is known. The sensors are connected to these sockets throught passive USB cables. The sockets are connected to the rest of the testbed through a combination of active and passive USB cables, depending on the distance due to the 5 meter limitation of the passive cables.

USB hubs

USB hubs enable TWIST binary power-control over the sensor nodes. The self-powered USB hubs support port power switching by sending a control message. This means, TWIST can control the power supply of any sensor node individually.

Super nodes

To relax the limitations of USB infrastructure, super nodes are introduced to the system for controlling several USB hubs in parallel. These are 32-bit embedded devices connected to networked storage systems and run a customised distribution of Linux operating system on them.

Server

The core of the server is a PostgreSQL database that stores the information and configuration data like node IDs, sockets, locations of the components and the binding of the components. This database is also used for recording debug and application data of the experiments.

Control station

The control station is a workstation that runs Linux on it. The control station invokes the scripts that run on super nodes through **ssh** remote command execution. These scripts provide functionalities like sensor node programming, executing, power control and collecting debug and application data.

2.4.2 TWIST Deployment

As of writing this document, the deployment of the TWIST testbed spans over three floors for over $1500\ m^2$ of the TKN building in Berlin, Germany. In the current configuration, there are 102 Tmote Sky (and TelosB) nodes and 102 eyesIFXv2 nodes. The number of super nodes is 46, which are connected to 60 USB hubs. The complete deployment contains over 1300 meters of USB cables.

2.5 Multi-Dimensional Scaling

Multi-Dimensional Scaling (MDS) is a multivariate statistical method that helps estimating the values along one or more continuous dimensions, which are defined as distances between pairs of objects [65]. It takes inter-object relationships (or distances) and represents those objects on a multidimensional map, where the distances between pairs of objects are related to the original relationships. It is used for analysing relationships and patterns given a set of points and dissimilarities among them. For example, it is used for providing a visual representations of these distances on a map. It is originally introduced in an article [66], in the journal of the Psychometric Society, called Psychometrika, which aims to develop psychological studies as a quantitative relational science. Their articles examine statistical methods, discuss mathematical techniques and advance theory for evaluating behavioral data in psychology [67]. Psychometrika has been active since 1936 and MDS is still very popular in psychological research.

In the later parts of this work, MDS has been used to compute the relative positions of sensor nodes. These positions, which are computed with the help of MDS, are then compared to the node positions that are computed by the systems proposed by this thesis.

2.5.1 Algorithm

MDS takes the distances between objects and rearranges them in such a configuration that is representative of the original distances. This rearrangement can be implemented in any number of directions, including one. For example, if the objects are cities and the relationships are the actual distances in meters, the two-dimensional output of MDS produces a geographical map of these cities. The actual orientation of the axes in the output of MDS is arbitrary, but the distances between the objects in the output is monotonically relational to the distances in the input.

What MDS technically does is; given a relationship matrix as input, finding a set of vectors in p-dimensional space, such that the euclidian distances between the pairs of nodes in that p-dimensional space correlate to the euclidian distances between the same pairs of nodes in the input matrix. The system is explained in a simplified way in Algorithm 1.

Algorithm 1 Multi-Dimensional Scaling; given input matrix D

1: Assign points to arbitrary coordinates in p-dimensional space
2: Compute euclidean distances among all pairs of points, to form the D' matrix.
3: Compare the D' matrix with the input D matrix by evaluating the stress function. The smaller the value, the greater the correspondence between the two.
4: Adjust coordinates of each point in the direction that best minimizes stress.
5: Repeat steps 2 through 4 until stress won't get any lower.

2.5.2 Input of MDS

MDS takes a set of relationships between the data points of the system as a symmetric square matrix. This matrix is called either a similarity matrix or a dissimilarity matrix. In a similarity

matrix, the node pairs get closer to each other with increasing value of relationship. A dissimilarity matrix, on the other hand, possesses an opposite measure of closeness, in which the nodes are closer to each other with decreasing value of relationship.

For example, a matrix of cities, with geographical distances as their relationships, is a dissimilarity matrix because the greater the distance is, farther apart the cities are. A *metric* MDS is the one, in which the disimmilarities are quantitative and a *non-metric* MDS is the one, the dissimilarities of which is qualitative (such as ordinal).

2.5.3 Dimensionality

The decision about the dimensionality for analysing data is a critical part of MDS. In most visualization purposes, the dimensions are selected among the set of $\{1, 2, 3\}$, because these are the dimensions that humans can perceive. The dimensions can also represent the number of significant or relevant relationships between the nodes. This is usually the case for dimension reduction problems. The number of dimensions also directly affect the stability and the computation time of the system.

Dimensionality is a complex part of MDS and it can be theoretically analysed much deeper. However, the scope of this thesis is finding geographical positions of objects, and therefore we will be interested in only one and two dimensions of MDS, as well as other algorithms introduced and mentioned in this work.

2.5.4 Stress

The correspondence between the distances among points that are calculated by MDS, which is provided by the input matrix is called $stress$ and it is measured by a stress function. In classical metric MDS the stress is calculated as in Equation 2.1:

$$stress = \sqrt{\frac{\sum \left(d_{ij} - \hat{d}_{ij}\right)^2}{\sum {d_{ij}}^2}} \tag{2.1}$$

, where d_{ij} is the Euclidian distance between points i and j, and the $\hat{d}_{ij}$ is the predicted distance of the MDS model. The lower $stress$ value is, the better the MDS output gets. In the perfect case, $stress$ approaches to zero.

Different implementations of MDS uses different stress functions. The quality of the output is directly affected by the stress function and the number of maximum iterations that an implementation uses to minimize $stress$.

Like most iterative algorithms, MDS algorithms are also prone to failing to reach an optimal solution. This can be caused by reasons like local minima, solution degeneracy, or an inadequate number of iterations. These issues are important to take into consideration when implementing or selecting an MDS system.

In this work, MDS was used to compute relative positions of nodes in a WSN. Several implementations have been tried out on the same measured data and SMACOF (”Scaling by MAjorizing a COmplicated Function”) algorithm [68] was chosen, where it performed better on our data.

Chapter 3

Related Work

Most commonly used positioning systems utilize GNSS technologies (such as GPS and Galileo) with over 5.8 billion devices that contain one of the GNSS chips in year 2017 [69]. However, these systems do not perform well indoors due to their requirement of direct line of sight to the satellites. They also increase the energy consumption of the low-power devices that they are attached to. Therefore GNSS-based systems are not favorable in low power WSN, especially for indoor applications.

Plethora of algorithms are developed for WSNs for calculating the node positions without deploying expensive gadgets (such as GPS modules) on each of them. Yet, positioning in WSN is still very challenging, since the nodes have very limited computational capabilities and often are required to run on small batteries [70]. WSN nodes also have to collaborate with each other to position themselves, with a cost of increased deviation in precision.

3.1 Positioning Techniques in WSN

Most of the positioning techniques in WSNs are based on *Angle of Arrival (AoA)*, *Time of Arrival (ToA)*, *RSS measurements* and *connectivity*. Each system has its own advantages and drawbacks. For example AoA requires special antennas and ToA performs poorly if there is no LoS. RSS-based techniques often require exhausting training phases and they are too sensitive to environmental changes and noise . Some systems use hybrid mechanisms to compensate for drawbacks of the positioning techniques they use [71]. These techniques are discussed in more detail in the following part.

The positioning techniques require few nodes with known geographical position information, which are called *anchors or reference nodes* and their locations are provided to the rest of the network. The nodes with unknown locations are called *non-anchor or blind nodes* and their positions need to be computed using measurements originating from or aligned to the reference nodes.

3.1.1 AoA-based Positioning

In AoA-based techniques, the nodes create virtual bearing lines passing through the relevant anchor node and themselves [72]. When two or more of those bearing lines intersect at one point, the nodes can determine their locations by using the angle between the bearing lines and the distance in between the anchor nodes. The computation is a simple geometrical calculation. In existence of noise the bearing lines from more than two anchors may not overlap. Then, a triangle will be formed from those lines and the location will be estimated by using triangulation calculations.

The angle of the incoming signal can tell a sensor node the orientation of the incoming signals, which is called angulation. Angulation techniques require special equipment, such as antenna arrays, and require direct line of sight between the receiver and the transmitter.

One way of detecting the AoA is *Beamforming*, [2]. This technique uses the anisotropy in the reception pattern of an antenna. The beam pattern of the receiver antenna is rotated mechanically or electronically. The direction of the antenna where the maximum signal strength is received, is assumed to be the direction of the transmitter, see Figure 3.1. However, this approach fails to detect the correct direction of the transmitter if the transmitted signal amplitude changes. To handle this issue, a second omnidirectional antenna is used to normalize the strength measure of the received signal.

Figure 3.1: The illustration of the horizontal antenna pattern of an anisotropic antenna [2]

Another similar approach is using more than one stationary anisotropic antennas with known patterns, [2, 73]. This approach makes the antenna patterns overlap to decide from which is the signal coming. The antenna that receives the highest signal strength is considered to determine the direction of the signal and other antennas are used for fine tuning the angle. In this approach, more number of antennas lead to the better estimation of the direction. According to [73], eight antennas can provide an accuracy of 2 degrees.

A different type of AoA technique is detecting the phase differences in the arriving signal, [74]. This approach requires an array of antennas, which are separated by a uniform distance. The transmitter's signals arrive at the elements of the antenna array with different phases. By using the distances between antennas, one can calculate the angle of the incoming signal. However, this approach requires relatively high signal strength and it is very sensitive to inter-

ference and multipath.

The directivity of the antenna, shadowing and multipath limit AoA accuracy. The AoA measurements highly rely on the LoS path between the transmitter and the receiver. Multipath reflections can appear as separate signals coming from an entirely different direction, which make AoA measurements difficult for positioning. To counteract, some deterministic and stochastic *maximum likelihood* algorithms have been proposed [74], which can increase the accuracy of AoA measurements with a cost of greater complexity.

AoA-based techniques have limited use, due to the LoS requirement and high sensitivity to noise. One study by Lee et al. [75] combines the angles between the pairs of nodes with the distances between the same pairs and computes the coordinates using basic trigonometry. These calculated coordinates are disseminated through the rest of the network and the procedure is repeated for each node. Then the calculated coordinates of each node by other nodes are averaged to find the final position. The simulations of this study give over 1.5 m of location error, which increases as the node density increases.

A study, named Localization and Routing in Sensor Networks by Local Angle Information, shows a use of local angle information and connectivity in geographic localization of WSNs [76]. The authors state that the angle information is sufficient for a topology control problem that assumes location information, even though it is not sufficient to derive the global geometry. The topology is taken as a graph and the intersection of the edges in the graph is decided depending on the angle in between those edges and the lengths of the edges do not matter. Their work focuses on generating the routing paths rather then providing the locational coordinates to the nodes. To do this, embedding of unit disk graphs with angle information is used. The authors discussed that this problem is in fact NP-hard but their linear programming based solution provided good results. The study remains on a theoretical level, showing how the angle information can be used for location based routing without having actual location information.

3.1.2 ToA-based Positioning

In WSN, distance based positioning techniques are common. Computing the positions by using the distances to multiple reference points is called *multilateration*, [70]. In two-dimensional space, at least three anchor nodes are required to estimate the location of a sensor node. When a node computes its distance from an anchor node, then it should be somewhere within the circle that has the anchor node on its center and whose radius is the distance from the non-anchor node. By measuring the distance from three or more anchor points, the non-anchor node can tell that it lies inside the area where all three circles intersect. Ideally, a non-anchor node tries to minimize the difference function, given in Equation 3.1, for all combinations of three anchor nodes that it can communicate to.

$$\begin{aligned}\sigma_{x,y} =& |\sqrt{(x-x_A)^2+(y-y_A)^2} - r_A| + |\sqrt{(x-x_B)^2+(y-y_B)^2} - r_B| \\ &+ |\sqrt{(x-x_C)^2+(y-y_C)^2} - r_C|\end{aligned} \tag{3.1}$$

In the formula above, the non-anchor node is at point (x, y) on the plane, where the anchor nodes A, B and C are at $(x_A, y_A), (x_B, y_B)$ and (x_C, y_C) respectively. The measured distances

to the anchor nodes are given by r_A, r_B and r_C and the location accuracy estimation quality $\sigma_{x,y}$ is minimized.

One way of computing distance between wireless tranceivers is by propagation time measurements. Propagation time is either one-way or roundtrip propagation time, Time of Arrival (ToA), or Time Difference of Arrival (TDoA) measurements [2].

GPS-based positioning is the most typical example in this category. The GPS receivers on earth, receive transmissions from GPS satellites and calculate their distances to those satellites. Although only three satellites are sufficient to locate a receiver, a fourth satellite is required to correct possible synchronization error of the receivers.

In ToA measurements, the distance between the transmitter and the receiver is estimated by measuring the time a signal takes to reach its receiver. One-way propagation time measurements timestamp the signal at the sender and subtract that time from the receiver's local time. This method requires highly accurate synchronization of the nodes in the network, which increases the complexity of a WSN application. Such precise local time stamping mechanisms make ToA-based approaches often inapplicable for energy constrained systems. Similarly, a roundtrip propagation time measurement timestamps the transmitted packet twice at the sender. It measures the total time of the transmission, which is sent and is bounced back to the original transmitter immediately by the receiver. Hence, the transmitter can use its local time to compute how long it takes for the packet to travel that distance twice. However, the problem with this approach is the unpredictable delays at the receiver side, before sending the packet back to the transmitter. To overcome this problem, the receiver puts its local receive and send times inside the packet, allowing the initial sender to subtract the delay in the receiver side from the total travel time of the packet [70]. However, most power constrained WSNs cannot benefit from roundtrip propagation time of the communication packets as it is since the RF signals travel through the air at the speed of light, which is much greater than the resolution of the processors used in most WSN hardware.

When there are more than one receiver, which are all able to receive transmissions from a single sender, TDoA becomes another applicable method [2]. In this technique, several receivers receive the same signal from one transmitter and the reception lags in compare to other other receivers are measured as a function of distance. This technique, too, requires highly accurate time synchronization and it suffers from multipath. TDoA-based positioning can be used when the non-anchor nodes know the time difference in their receptions of signal from an anchor node. This way they can create a virtual hyperbola, which passes through the exact location of the anchor node. The higher is the number of non-anchor nodes receiving the same signal, the higher is the number of hyperbolas created. This allows better estimation of their locations. By definition, hyperbolas are non-linear and so are their equations. Therefore, location estimation by using TDoA methods requires lots of complex non-linear matrix calculations and a good Maximum Likelihood estimator. These issues make the use of TDoA based localization techniques less applicable for the sensor networks.

The authors of RF Time of Flight Ranging for Wireless Sensor Network Localization [77] studied the use of RF signals for wireless sensor network localization. This work requires a very specific type of hardware, as they used low power external boards with an RF interfaces, very accurate clock and a microcontroller. They transmitted RF signals at 2.4 GHz frequency and

measured the Time of Flight (ToF) between two sensor nodes. After a short synchronization process, the nodes exchange ranging signals and measure the time of flight in nanosecond precision. The system is tested in indoor and outdoor experiments. Outdoors, the accuracy was as good as 1 meter, while it was 3 meters indoors.

NanoLOC [78] is another technology that is used to measure distance. It contains a mixed signal chip which uses a ToF method that employs a ranging signal sent by a reader and an acknowledgement sent back from the tag to cancel out the requirements for clock synchronization. The method provides a protection against multipath propagation and noise by its Chirp Spread Spectrum modulation technique and uses 80 MHz of bandwidth. The system operates at 2.4 GHz. For ranging, chirp pulses are used, which are linear frequency modulated signals with a constant amplitude. So each chirp pulse that's transmitted through the air, has an linearly increasing frequency. The system is commercially available and is tested many times. The measured precision of ranging is 2 meters indoors and 1 meter outdoors.

The Cricket location support system [79] proposed the use of ultrasonic signals in addition to the RF signals for supporting localization systems, since the speed of sound is much lower than that of RF, which allows the WSN hardware measure the time difference in between the transmissions. In this approach, the transmitter sends an RF signal, letting the receiver know that the ultrasonic signal is also transmitted. On reception of the RF signal, the receiver enables its ultrasonic receptors and waits for the sound signal to arrive. The time difference between the RF reception and sound reception is then used to calculate the exact distance between the transmitter and the receiver nodes. Cricket is one of the studies that has been heuristically tested. The experiments show that the position estimation error rate in a stationary deployment (non-mobile) approaches to 0 % with the use of enough number of samples. Even though the accuracy of this method is rather high, it suffers from multipath effects, which increases the complexity of the system. Another limitation is that common low power ultrasonic modules have approximately five meters of effective transmission range [80]. Another problem with this approach is, if the RF signals collide, the receiver node will have difficulties in distinguishing the ultrasonic signals one from another. Also, the sound signals suffers more from severe multipath effects that are caused by the reflections from the walls.

The n-Hop Multilateration Primitive for Node Localization Problems [80] explains a way to position sensor nodes in a multihop setting, which the authors call *collaborative multilateration*. The positions of the nodes are estimated by setting up a global non-linear optimization problem and solving it by using iterative least squares. Collaborative multilateration can be done either in a centralized or in a distributed manner. This algorithm is a range-based algorithm and the range is assumed to be measured by the use of ultrasonic sensors. The system has four phases. In the first phase, the nodes create groups among each other to form subtrees to determine which nodes have unknown locations. In the second phase, the nodes estimate their locations initially by using their distances to other nodes and the locations of the anchor nodes. In the third phase, the initial location estimates are refined by using iterative least squares method. Finally the new location information is used to tune the locations of the nodes that might have poorly estimated locations. The simulations show that the localization accuracy has an average error of 3 cm.

UWB signals can be used to increase the accuracy in distance measurements [81]. The bandwidth of UWB signals are typically very large and hence a pulse in the signal takes a very

short time. Sending such short pulses in the signals allow the receivers decouple the original signal from the reflections, which results in better prevention of multipath effects. However, the UWB systems get harshly affected by NLoS situations.

In the study [81], UWB radios are used for generating distance information. The UWB radios are selected because their high time resolution allows UWB hardware to provide better accuracy with time based distance measurement schemes. The experiments show that this approach provides more accurate results as the bandwidth of the transmitted signal increases. With a bandwidth value that is above 2.4 GHz, we can get less than 0.2 meters accuracy. When the bandwidth goes above 5 GHz, the accuracy becomes less than 0.1 meters and approaches to 0 as the bandwidth increases.

A different approach to measure the distance between two points is by using light beams, which is called *lighthouse approach*, [82]. This approach uses a light source which is constantly producing light strong enough to reach to the nodes that want to measure their distance. The source is rotating around its own axis, like a lighthouse, and the receiver derives the distance to the light source by measuring the duration it receives the light beams from the source. The nodes can have a very small optical receiver, which is an advantage in compare to using ultrasonic receivers, but the transmitter will be large. This approach, too, requires direct line of sight between the light transmitter and the receiver.

3.1.3 RSS-based Positioning

RSS measuremets is another commonly used method of position discovery, through either distance estimation or fingerprinting. A Received Signal Strength Indication (RSSI) is provided in almost all wireless communication devices. RSSI feature does not cost extra power consumption and does not require any additional hardware. This technique is based on the fact that RSS's being inversely proportional to the distance from transmitter by the Friis equation [83], which is shown in Equation 3.2.

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2} \tag{3.2}$$

where P_t and P_r respectively are the transmit and receive power values, G_t and G_r are transmitter and receiver antennas' gains, λ is the transmitted signal's wavelength and d is the distance in between the transmitter and the receiver. This formula represents the free space model and it is does not consider any environmental attenuation. This over-idealization is decreased by using a path loss exponent parameter α, which makes the equation more realistic in Equation 3.3 [2] [83].

$$P_r(d)[dBm] = P_0[dBm] - 10\alpha log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \tag{3.3}$$

, where $P_0[dBm]$ is the reference power value in dBm at a reference distance d_0 and α is the path loss exponent, which corresponds to a rate at which the received power strength decreases with distance. X_σ is the effect of shadowing, which is the Gaussian variation around the path loss value.

RSS-based fingerprinting is preferred for indoor scenarios, due to its implicit feature of taking characteristics of the environment into consideration. This technique works in two phases: training and testing [84]. In the training phase the fingerprints (usually RSSI and sensor values) are collected and stored in a database. In the testing phase the fingerprints in the database are matched with the current measurements to estimate positions. The matching of the fingerprints are usually done through a machine-learning-based system, such as *nearest-neighbor* or *k-nearest-neighbors*, or even *neural-networks*.

In the study Indoor Fingerprint Localization in WSN Environment Based on Neural Network [85], RSSI fingerprints are collected and fed into a neural-network (Levenberg-Marquardt) algorithm to detect node positions. The authors defined 240 measurement points in a 86 m^2 indoor space at roughly equal distances to each other. The fingerprints of 100 RSSI values were taken from 5 anchor nodes. These measurements were then fed to the neural-network model with 5 inputs (for each anchor) and 2 outputs (x, y coordinates). The system resulted in 0.3 m positioning error in most cases, and the error went up to over 10 m in some other cases.

Similar results were reported by the authors of [86]. They have used both range-based RSS positioning (namely, multilateration and Min-Max) and RSS fingerprinting mechanisms (nearest-neighbor and k-nearest-neighbors) to compute node positions. Their results show between 0.12 m and 0.60 m position error for range-based techniques, and between 0.312 m and 0.55 m position error for fingerprinting-based techniques.

In the study Probabilistic Localization for Outdoor Wireless Sensor Networks [87] an outdoor localization methodology based on RSS measurements was proposed. The inaccuracy in the RSS measurements are handled by a distributed, probabilistic approach and the complexity of the computations is reduced by the use of *fast Fourier transforms*. The proposed algorithm starts with a calibration process, in which they create a map of RSS to a probability density function (PDF) of the corresponding distance. The calibration is done by placing a transmitter and a receiver at several known distances from each other and the RSS is measured (and logged) at the receiver for many packets. After this, a non-anchor node assumes its location by looking at the map and measures the RSS from an anchor node. Then it intersects this new measurement with the old PDF and broadcasts the new PDF to its neighbors. The estimations of the non-anchor nodes are then improved by using negative constraints. In addition, the authors reduce the computational complexity of the algorithm by the use of fast Fourier transforms. The simulations show 0.2 error rate in node positions.

The authors of Wireless Sensors Self-Location in an Indoor Wireless Local Area Network (WLAN) Environment [88] analysed positioning by information from indoor WLAN systems for positioning of indoor WSN. Two systems are studied; a triangulation method and a heuristic method using neural networks both of which use RSS to localize the nodes with the help of WLAN APs. In neural networks based approach, the sensor nodes start with a training phase, in which they save the RSS values for each AP and the values are stored in the database. In the latter phase, they compute the positions by comparing the received signal strengths to the data stored in the database. In the second case, the sensor nodes locate themselves with triangulation by using the prior knowledge of the AP locations and their antenna gains. The distance of sensor nodes to the APs are measured by the momentary RSS value and triangulated for location estimation. As performance of the algorithms, the authors conclude that, triangulation-based localization works

better then the neuronal-network based system in general. But if the system has various types of walls among the sensors and APs, the neural network based positioning works better than the triangulation approach. The positioning error they achieved with the triangulation based system is 1.3 meters, while with the neuronal network based system it is 2.5 meters. However, these results are taken in different sites.

3.1.4 Connectivity-based Positioning

Position discovery is also possible using the connectivity information between the nodes within the network. By evaluating the information of which nodes can communicate to which other nodes, the node locations can be estimated [89].

In [90], the packet delivery ratio between the nodes is taken as the connectivity metric. Using this method, a non-anchor node evaluates its connection quality to its neighbors with known (or computed) locations and it estimates its own position by calculating how much distance does the measured connection quality correspond to.

Another method is introduced in [91] is called *distance vector - hop (DV-hop)*. This work assumes that the nodes of the network are spread into the area uniformly. The system starts with the anchor nodes disseminating their location information within the network. While this information is flooded by the non-anchor nodes, the hop count, which starts at zero from the anchors, is incremented. Whenever an anchor node receives a location packet from another anchor, it calculates the average inter-node distance by dividing its distance to that anchor node by the hop count. This distance information is then spread back inside the network and non-anchor nodes calculate their distances to those anchor nodes. When a node has enough distance information from different anchors, it estimates its location by trilateration.

In [89] an MDS-based system is proposed, which takes connectivity information as the input. The distances are represented as shortest path distances. These inter-node distances are converted to coordinates by using the classical MDS algorithm. Then, the results are normalized to the map by fitting the computed positions of the reference nodes to their actual positions. This system resulted in 2.4 m positioning error in their simulations (assuming that the unit lenght is selected as 1 m). With refinements to their system, by clustering the nodes to position and joining these clusters, they achieved 0.29 m of positioning error in the simulations. In later chapters of this thesis, an adaptation of this system will be used for performance comparison with our proposals.

Another connectivity-based positioning system is introduced in the paper “Range-free Localization Algorithm Using Expected Hop Progress in WSN” [92]. This study proposes a range-free localization algorithm that uses expected hop progress for sensor location prediction, which is based on accurate analysis of hop progress with randomly deployed sensors. The authors show that the expected hop progress is a function of node density, node connectivity and transmission range. The proposed system fuses the trilateration techniques with the expected hop progress. Each sensor node is assumed to store a database in its memory which maps the node connectivity to the corresponding expected hop progress. The simulations show that this algorithm leads to a distance estimation error of 0 with 0.425 probability. As for the performance, over 90 % of sensors estimate their positions within a deviation of one transmission range from the actual positions.

Summary

Computing the locations of very low capacity sensor nodes is challenging. Many algorithms have been developed using various positioning techniques.

For many sensor network positioning systems, the size of the network is a big concern. The cost and scalability limitations make *angle of arrival* techniques less applicable. Mounting extra directional antennas or using anisotropic antennas add up to the cost and require more complex architectures. A better alternative to AoA-based approaches could be the *lighthouse* technique. However, in many indoor sensor network applications, expecting a direct line of sight to the reference light sources is hard and the preconfiguration of the system is cumbersome.

ToA/ToF-based techniques are relatively more applicable to the sensor networks. Those techniques can be very efficient outdoors and can locate nodes in relatively big distances. However, RF ranging techniques are highly error prone indoors. Although UWB seems like a good alternative to RF based ranging systems, their precision is also within the ranges of several meters and often require LoS. Ultrasonic based positioning provides centimeter-level precision quality, however it requires installment of special infrastructure and sound signals do not travel more than a couple meters away when utilized on low-energy sensor networks.

RSS-based positioning receives more interest from the scientific community because RSS information is very often readily available on the communication radio chips. Two biggest type of this kind of positioning is Fingerprinting and RSS-ranging. Fingerprinting requires an exhausting training phase and once the anchors are replaced, the training phase must be repeated. Ranging methods are too prone to the environmental noises and conversion of RSS to physical distance is not reliable enough.

Connectivity-based positioning is the easiest to implement in comparison to other types of positioning techniques, however they do not work if the network is too dense. If all of the nodes are within each other's proximity, it is not possible to make a distinction between the nodes. Some hybrid algorithms combine connectivity information with other types of information, such as RSS, but this brings extra cost to the application and it is still not precise enough.

3.2 Discussion

In the previous section some positioning techniques, which we think are most representative to WSN, are discussed. Indoor positioning systems are not limited to the field of WSN, in fact it is a widely studied research topic. Plethora of papers are published each year in this field, and many of them provide very promising results.

Studying indoor positioning or localization is not easy, especially when it comes to the evaluation of the systems and experiments. Researchers often analyse their systems on a simulated environment, or they have access only to limited spaces, which causes over-customizing their algorithms to their environments. These issues make it very challenging for the scientists to compare their systems to other systems. Because of these problems, there are now performance evaluation platforms, such as PerfLoc [93].

Microsoft Research has been organizing the *Microsoft Indoor Localization Competition* [94] since 2014. The purpose of this series of competitions is to give different academic and industry

groups an opportunity to test their indoor localization technologies in a realistic, unfamiliar environment [27]. They categorize the competing systems as *infrastructure-free* and *infrastructure-based*. The infrastructure-free systems utilize only existing Wi-Fi infrastructure that is assumed to be already there and they fuse the Wi-Fi signals with sensors, such as accelerometer and gyroscope. The infrastructure-based systems deploy extra hardware to the environment, such as bluetooth beacons, magnetic resonators, ultrasound transmitters and customized RF transmitters.

In the 2014 competition, the winning team achieved 0.72 m accuracy and only three teams achieved an accuracy under 2 m. The average position error changed between 0.72 m and 10.22 m. The position error that was seen from these systems were 2.5 m and above according to the CDFs of the location errors of these systems [27].

In later years lidar-based systems also appeared in the competition and achieved higher accuracy (between 0.2 m and 0.03 m) than the other systems. However, despite their high accuracy, their cost, size and high energy requirements prevent these systems from becoming a mainstream indoor positioning system, especially for WSNs [28].

In the 2014, 2015 and 2016 competitions a few infrastructure-free positioning solutions achieved an accuracy of 1-2 m but most of them around 3-4 m. Even the fingerprinting-based infrastructure-free systems showed great variation in performance across years and across teams, and they were not able to provide below 2 m accuracy, although fingerprinting has been a well-studied methodology. Similar inconsistency also appeared in infrastructure-based systems, with their performances varying between 0.23 m and 3.22 m in accuracy.

In the 2017 competition, the teams were given 7 hours for their preparation and training phases. Each team needed on average 5 hours to finish preparation, which is a significant amount of time if an environment changes day to day.

Considering the published literature work in the field of indoor positioning systems, one can claim that relative position discovery on a grid-like setting can be achieved by using one of those published studies that provides an accuracy level that is smaller than the sizes of the cells of the grid. However, the evaluations and competitions for indoor localization solutions in foreign and uncontrolled environments show that consistency in performance and adequate precision (in scales of few decimeters to one meter) has not been achieved yet. Despite the claimed research results, practical applications of indoor position discovery are far from being applicable. Absence of a mainstream indoor position discovery solution, which has been commercialized as of 2017 is also an anecdotal evidence that, an adequately precise and consistent indoor positioning solution does not yet exist, which is low on cost and energy requirements. Therefore, in this work we aimed to develop the cheapest (in terms of hardware, cost, energy and time requirements) system for discovering the positions of nodes in a WSN under the assumption that precise cartesian coordinates are not necessary.

3.3 Towards Preconfiguration-free Position Discovery

Developing a cheap indoor positioning system encouraged us to use only the available information from the common WSN hardware, therefore we investigated utilizing RSS as the main source of information for the rest of this thesis. RSS information during the regular communication of the wireless nodes is readily available in most common WSN hardware platforms, such

as the ones listed in Section 2.2.1.

Multiple studies, such as [95], show that RSS data by itself does not provide practical use for range estimation. The studies that rely on RSS information often require an intensive calibration process or a database for mapping the RSS values to actual distances or positions. Those computations are environment specific and computed mappings are not directly reusable in different locations without recalibrating. In [96], a calibration method that promises to reduce the error in localization has been described. Their method places two wireless nodes at known locations (hence, known distance to each other) and then samples RSS between each other. They repeat this procedure at varying distances and generate a dB scale RSS attenuation model versus distance. They update this model through out the operation of the network amongst the anchor nodes.

A good comparison of some of the best known RSS based positioning algorithms is given in [97], with conclusions that the relative positioning error is between 50 % and 100 % in most cases.

Another comparison is given in [98], focusing on indoor applications. It is shown that even when the nodes are located between 1 to 2 meters apart in a grid, the error is more than 2 meters with a probability of 0.5 or higher.

Despite several years of prolific research activity in the area of radio-based localization algorithms [99], the most popular and precise methods for indoor localization still leverage the tried method of signal *fingerprinting*, following the pioneering work of Microsoft on the RADAR system [100]. Several studies have demonstrated that the fading-related distortions, which make *range-based* RSS approaches impractical for precise indoor localization [3, 95, 101].

This realization has stimulated much interest in *range-free* approaches like the centroid or APIT methods [102]. This dissertation on position and sequence discovery shares the view with the range-free methods that the RSS information should not be directly used to invert the path-loss curve in order to estimate the range between two communicating nodes. Instead of complex approaches to estimate and compensate the distortion effects of the fading, we put higher weight on leveraging the diversity in the frequency domain which allows us to achieve slightly different fading behaviors even for static nodes [103]. This has been proven very useful also for non-RSS range-based methods like the ToF approach presented in [104].

Averaging RSS samples across different channels, as means of improving the stability of the mean RSS estimate, has also been explored in the works presented in [105] [24]. This thesis uses the simple averaging as only one of several different metrics for position detection, along with better ones. For example, the comparative benefit of concentrating on statistics involving the higher-magnitude tail of the RSS sample population is also analysed, similarly to the focus on the shortest ToF samples in [104]. Also, the inter-dependence of the different proximity metrics with our algorithms are considered and the combination in more challenging environments with strong NLoS components are evaluated.

The field of indoor position discovery clearly lacks an affordable system that solves the problem for all types of applications. Even if we want to discover the positions of the objects relative to each other, there is no one-size-fits-all type of solution. Therefore we want to solve this problem for the specific case, where the nodes are placed on a regular grid setting and the positions are represented relative to each other. This definition of the problem fits to many indoor, office-

like settings, where the space is partitioned into regular sections and relative positions of the objects can be mapped to real locations in the space prospectively.

This work proposes several methodologies for discovering the positions of a definite number of wireless nodes in a roughly regular grid setting. Both single-dimensional ($1 \times n$) (LSN) and two-dimensional ($m \times n$) settings are discussed and solutions are proposed. LSN approaches and applications are relatively recent, yet there has been a good number of studies done in this field. The applications of LSNs include, but are not limited to, liquid pipeline monitoring, bridge and tunnel monitoring and actuations, railroad and subway monitoring, border monitoring, sea coast monitoring, as well as powerline monitoring [16] [106] [107].

Even though a grid setting is a common configuration indoors, it has not been exclusively studied within the scientific community throughly enough. Hence, this chapter continues with a brief discussion of most relevant work that can be utilized towards node position discovery in a grid setting.

In [108] the authors describe a ranking algorithm, which discovers the ordering of the nodes using connectivity information and the knowledge of first node in the network. The system first discovers one and two-hop neighbors by sending a $hello$ message. Two-hop discovery requires exchange of one-hop neighbor lists within the network. After that they run a centroid algorithm and compute the order (ranks) of the nodes. We see two big limitations in this type of connectivity-based approaches; first, they require the node deployment to be sparse enough, such that their systems can implicitly label the nodes that are not within each others communication range as far nodes, secondly they cannot support a fully connected wireless network, in which any node can communicate immediately to any other node. The success rate of the algorithm reaches to 95 % if the connectivity degree is 1 within the network, and their results show that when the connectivity of the reference node increases to 6 or more, the success rate drops under 40 % for a network of 13 nodes. These rates are achieved with 5 m inter-node distances, while our system works with inter-node distances of under 1 m. Their approach was also tested only in the simulation environment, while we verified our approach both on real-world measurements and on simulated data.

In [109] the authors created a spatial ordering algorithm for one-dimensional WSNs. They advocate that RSS information is useful for localization algorithms and a comparative ranking of RSS values correlate with the spatial ordering of the nodes in a linear setting. Their claim is, smoothing (averaging) the signal strength values with as little as 2 samples provide 99 % chance of correctly identifying the closest node information from a single node's point of view. Therefore the authors conclude that ranking of RSS values on a single transmission channel can be mapped to node ordering for about 99 % of the time. Unfortunately we have not been able to reproduce these results with our real-world experiments (see Chapter 4). Also other studies in the literature (such as [110]) appear to be in contradiction to the statements that are presented in that work.

The authors of [111] published their results for a system to detect relative node positions in a LSN topology. They rank the immediate neighbors of each node in a neighborhood list and iteratively select the most likely next node for building the order. It seems that the authors did not notice our work [112], which has been published two years earlier. But their results, using a similar approach to what we have presented in our paper, confirm the correctness of our work!

In [113] the authors introduce the concept of sequence-based localization, a 2D range-free approach where the space is divided into regions that can be uniquely identified using sequences representing the rank of distances from a set of reference nodes. Their procedure for determining the unknown node location sequence (the distance rank table from the unknown node to all reference nodes) faces the same challenges due to the distortion of the RSS signal through shadowing and multipath as in this work. They do not attempt to mitigate these effects using statistical methods or frequency diversity, and only rely on comparing the corrupted distance rank sequence to the set of all possible rank sequences. Their method implicitly requires a high number of reference nodes, in comparison to the number that is required in the proposals of this thesis.

Another 2D range-free positioning and tracking scheme, based on relative distances between neighbors, has been presented in [114, 115]. The approach aims at improving the performance of pure connection-based, range-free methods through the concept of a "Relative Signature Distance (RSD)", new metric expressing the expected proximity between 1-hop neighbors. Even though that approach has also been evaluated for a long one-dimensional node configuration outdoors, the focus is on estimation of the spatial coordinates of the nodes instead of just the node sequence. The approach also lacks any mechanisms to deal with the challenging indoor conditions that are in the focus of this work. A similar attempt to improve the connection-based range estimates is presented in [116]. They derive the node locations locally from expected hop progress, in which the nodes broadcast their own transmission capabilities, and use the anchor locations that they are aware of. However, the accuracy of the positioning is only within half of a node's transmission range. In comparison, our work not only deals with much denser settings, but also implicitly supports heterogeneous networks and the nodes do not need to announce their transmission capabilities, since the closest node selection mechanism uses one sender and multiple receivers.

A method for relative position discovery of wireless sensor nodes in linear typologies, presented in [110] is the most similar to our work, from the point of view of the common focus on determination of the correct node sequence, instead of estimation of spatial coordinates. The proposed method shares resemblance with our single-channel, greedy, average-RSS approach, introduced and it is more extensively evaluated in one of our papers [112]. Similarly to the other related work, however, their method has not been evaluated in indoor environments. This simple approach, too, is not able to deal with the challenging fading disturbances that are typical for indoor environments. In that study, accuracy was commented on, however any indication of accuracy detection does not exist like it is provided in this thesis.

An anchor-free localization method is proposed in [117] that is comparable to MDS-MAP. The authors use estimated distances between the neighboring nodes and generate the node positions as a result of a central non-linear optimization process. They claim that they can increase the localization accuracy, if MDS output is given to their algorithm as the initial state. However, they rely on measured ranges from one node to another, which is in practice sensitive to the challenging propagation effects elaborated which results in erroneous location estimates.

The surveys [118], [119], [120] commonly concluded that range-free position discovery systems fall short in accuracy. In contrast, this thesis proposes techniques for a highly accurate indoor position discovery when the nodes are relatively close to each other, and ways of assessing

whether the perfect accuracy has been reached. This qualifies our work as being more applicable for a range- and preconfiguration-free position discovery system, even for challenging and dense indoor environments.

Chapter 4

Determining Node Closeness at a Single Channel

In previous chapters, the ineffectiveness of converting RSS values into distances, as well as difficulty of utilizing raw RSS values for various types of indoor position discovery methods were discussed. Despite the fact that computing geographical positions by relying only on signal attenuation level is very difficult, we attempt to find ways for identifying the closest node to a particular transmitting node in this chapter.

Our intuition is, if each node can detect the closest node (among many nodes) to itself, the spatial sequence mapping of nodes can be created. For example, the sequence of nodes, which are positioned in a linear setting, can be iteratively discovered starting from the node at the beginning of the sequence (the reference node). Therefore, detecting the closest node is essential to further construction of the complete map of relative node positions.

We will start investigating the possibility of using RSS for closeness detection, as it is an already available information that comes at no extra cost from the radio chip. In this chapter, different ways of processing such data will be analyzed through real-world experiments. These experiments will be repeated for different transmission power levels and different distances between the pairs of nodes.

4.1 Approach

The goal of this section is to analyze different techniques for detecting the closer one of two nodes. For the analysis, the RSS values from a number of transmissions between one sender (reference) and two receiver nodes are sampled. By examining the distributions of all the values taken in the measurements, we will try to choose one of the receivers as the closer node to the reference node. An experiment is composed of a set of transmissions at the same transmit power level from one reference node and the RSS values are collected from two receiver nodes.

To conjecture whether an experiment is conclusive or not, we defined the metric $Trust\ level$, which takes a value between 0 and 1. Greater values of the trust level indicates higher confidence in our decision. After having a reasonable number of experiments, we will observe the corre-

lation between the *trust level* and the verdict by looking at how many times and under which circumstances the ground truth is reached (the closest node is correctly detected). We plan to set the *trust thresholds* for classifying the future verdicts.

The symbols that we use here are as the follows:

Assuming that Node A and Node B are measuring from the same reference Node R:
$i \rightarrow$ Index of the transmissions of R in an experiment
$*RSS_A \rightarrow$ Set of RSS measurements by Node A throughout an experiment
$RSS_A^i \rightarrow$ Value of an RSS measurement by Node A from the transmission i
$\mu_A \rightarrow$ Mean value of $*RSS_A$
$\sigma_A \rightarrow$ Standard deviation of $*RSS_A$
$E_a \rightarrow (RSS_A^i > RSS_B^i)$ The event that RSS_A^i is higher than RSS_B^i for transmission i
$*E_a \rightarrow$ The set of all events E_a
$P(E_a) \rightarrow$ Empirical probability of event E_a
$Closernode = A \rightarrow$ The verdict is node A is closer to Node R node than Node B
$\lambda_A \rightarrow$ **Trust Level** if $Closernode = A$

We divide this approach into several sub-hypotheses and examine them under the same condition (measuring on a single channel), which are given in the following sections. We anticipate that at least one of these sub-hypotheses will help us to find a correlation between a verdict for *closer node* and the *trust level* of that decision.

4.1.1 Sub-Hypotheses

We introduce 5 hypotheses for identifying node closeness, which take the RSS values as input.

Sub-Hypothesis 1: The bigger the mean of RSS, the closer the node

We will first try comparing the mean of the RSS values from both nodes as a way of indicating the closer one of two adjacent nodes. These RSS values will be sampled from the same set of transmissions from a particular reference node. Therefore, if the mean of RSS values measured by Node A is greater than that of Node B, then we conjecture that Node A is closer to the Reference Node R than Node B is. We evaluate the *Trust Level (λ)* of this verdict by finding the proportion of the difference between the mean RSS values of both nodes to the mean RSS value of the closer node.

$$\begin{aligned} &if\ \mu_A > \mu_B\ then;\ Closernode_R = A \\ &\lambda_A = (\mu_A - \mu_B)/\mu_A \end{aligned} \tag{4.1}$$

Sub-Hypothesis 2: Comparison of empirical probabilities

Each of the two nodes measures the RSS value from the same transmission i, which is indicated with RSS^i. For a particular instance of transmission, where $RSS_A^i \neq RSS_B^i$, either of these two events occurs: $E_a = RSS_A^i > RSS_B^i$ or $E_b = RSS_B^i > RSS_A^i$. The empirical probabilities

of these two events are found by calculating the ratio of the cardinality of each event set ($|*E|$) to the total number of events, which are indicated by $P(E_a)$ and $P(E_b)$. We will use these probabilities as an indicator of the closer node, which is the node that is associated to the greater one of these two probabilities. The Trust Level (λ) is conjectured by the difference of these probabilities.

$$
\begin{aligned}
&E_a = RSS_A^i > RSS_B^i \\
&E_b = RSS_B^i > RSS_A^i \\
&P(E_a) = \frac{|*E_a|}{|*E_a| + |*E_b|}\ ,\ P(E_b) = \frac{|*E_b|}{|*E_a| + |*E_b|} \\
&if\ \forall_i,\ P(E_a) > P(E_b)\ then;\ Closernode_R = A \\
&\lambda_A = P(E_a) - P(E_b)
\end{aligned} \tag{4.2}
$$

Sub-Hypothesis 3: RSS interval comparison

In this sub-hypothesis, we define the "RSS Intervals (RI)" as a symmetrical interval around the mean values (μ) of RSS readings of each node. The RIs are bounded by the standard deviation (σ) of the measurements.

$$RI = [RI^{min}, RI^{max}] = [\mu - \sigma, \mu + \sigma] \tag{4.3}$$

By using σ, we will be processing on a more representative portion of the data. The decision on the closer node is given by comparing the boundaries of the RIs that are defined on the probability density functions (PDFs) of the measured RSS values by both of the nodes.

The use of *RSS Intervals* for data evaluation leads us to following three cases:

Case 1: If the RSS intervals of the two sets of samples are not overlapping, then the node which has the lower bound of its RI (RI_A^{min}) higher than the upper bound of the other node (RI_B^{max}) is taken to be closer to the reference point (transmitter node) with 1.0 level of trust.

$$
\begin{aligned}
&if\ RI_A^{min} > RI_B^{max}\ then;\ Closernode_R = A \\
&\lambda_A = 1.0
\end{aligned} \tag{4.4}
$$

Case 2: If the RSS intervals are overlapping, then we compare the mean values (μ_A, μ_B) and the upper bounds of the RIs (RI_A^{max}, RI_B^{max}). If both the mean value (μ_A) and the upper bound of the RSS interval (RI_A^{max}) of node A are greater than those of the other node's, then we conjecture that node A is closer. The trust level (λ) of the verdict is calculated by dividing the *non-overlapping* part of the two RIs to the RI of the $Closernode$.

$$
\begin{aligned}
&if\ RI_A^{max} > RI_B^{max}\ AND\ \mu_A > \mu_B\ then;\ Closernode_R = A \\
&\lambda_A = 1 - \frac{min(RI_B^{max}, RI_A^{max}) - max(RI_A^{min}, RI_B^{min})}{RI_A^{max} - RI_A^{min}}
\end{aligned} \tag{4.5}
$$

Case 3: If the comparison in Case 2 show that only one of those values (μ_A *or* RI_A^{max}) is greater than that of node B's, then we compare the differences between the mean values ($\Delta\mu$) to the differences between the upper bounds of the RIs (ΔRI^{max}).

The greater one of those differences (**significant difference**) will be used as an indicator of the closer node, with the set of equations below:

$$\Delta\mu = |\mu_A - \mu_B| \tag{4.6}$$
$$and$$
$$\Delta RI^{max} = |RI_A^{max} - RI_B^{max}| \tag{4.7}$$
$$and$$
$$\phi = \begin{cases} \mu_A, & \Delta\mu > \Delta RI^{max} \; and \; \mu_A > \mu_B \\ \mu_B, & \Delta\mu > \Delta RI^{max} \; and \; \mu_B > \mu_A \\ RI_A^{max}, & \Delta RI^{max} > \Delta\mu > \; and \; RI_A^{max} > RI_B^{max} \\ RI_B^{max}, & \Delta RI^{max} > \Delta\mu > \; and \; RI_B^{max} > RI_A^{max} \end{cases} \tag{4.8}$$

Above; ϕ is the value, which has a significance in the PDFs of one of the nodes. Therefore; the assumption is, whichever value $\phi \in \{\mu_A, RI_A^{max}, \mu_B, RI_B^{max}\}$ takes, is associated with the closer node.

$$Closernode = \begin{cases} A, & \phi \in \{\mu_A, RI_A^{max}\} \\ B, & \phi \in \{\mu_B, RI_B^{max}\} \end{cases} \tag{4.9}$$

The trust level (λ) of the verdict is again calculated by proportioning the *non-overlapping* portion of the two RIs to the RI of the $Closernode$. If the $Closernode = A$, then:

$$\lambda_A = 1 - \frac{min(RI_B^{max}, RI_A^{max}) - max(RI_A^{min}, RI_B^{min})}{RI_A^{max} - RI_A^{min}} \tag{4.10}$$

Sub-Hypothesis 4: Comparison of every n^{th} transmission

This sub-hypothesis performs the same evaluation as the sub-hypothesis 2, but uses less transmission samples that are taken from every n^{th} transmission. This is done to create a more sparse set of values, hence reduce condensed noise if there were any.

$$E_a = RSS_A^i > RSS_B^i$$
$$E_b = RSS_B^i > RSS_A^i$$
$$if \; \forall_{i=n\times j} \; P(E_a) > P(E_b), \; for \; j = \{0, 1, 2, ..., \lfloor N/n \rfloor\} \; then; \; Closernode = A$$
$$\lambda_A = P(E_a) - P(E_b) \tag{4.11}$$

Sub-Hypothesis 5: Smaller portions of the large measurement

In this sub-hypothesis, we divide the set of measured RSS values into subsets of 100 transmissions, having n subsets. Then we choose the k^{th} subset, which contains the smallest $\sigma_A^k + \sigma_B^k$ value and we apply the Hypothesis 2 on that subset only. This k^{th} subset is supposedly the most stable part of the collected data, due to having less variation. Here total number of transmissions should be more than a few multiples of 100.

4.1.2 Evaluation

We performed a set of experiments to test the proposed sub-hypotheses in a building at the Technische Universität Berlin. Three TelosB type WSN nodes were placed roughly along a line and one of them was used as the reference node while others were used as receivers.

The reference node was assigned the ID 0, closer receiver node was assigned ID 1 and farther receiver node was given ID 2. After applying our hypotheses on the measurements, if the algorithms indicated Node 1 as the closer node, the decision was considered correct and the decision was classified as incorrect if the Node 2 was chosen as the closer node.

The reference node (Node 0) transmitted 1000 packets at 10 ms intervals and two receiving nodes captured the RSS values of those transmissions from the RSS Indicator (RSSI) of the CC2420 chips on the nodes. Each set of experiments was repeated with a different transmission power level from the set $\{0, -5, -10, -15, -25\}$ dBm. 0 dBm is the highest and -25 dBm is the lowest power level that CC2420 transmits at. The reason for running the algorithms at different transmit power levels was to understand if these power levels would influence the results.

During the first set of experiments, the nodes were positioned at equal distances in a large hall without any furniture. There were two active Wi-Fi access points on the walls and the building was hosting institutes that performed research on various wireless networking technologies, therefore some uncontrolled electromagnetic noise was expected.

The nodes were placed 100 cm apart from each other (meaning; the first receiver is 100 cm and the second receiver is 200 cm away from the reference node) and the RSS values were observed from 1000 transmissions. The nodes were on a single plane, put on tables without any obstacles in between, roughly along a straight line. The results of the evaluations of the sub-hypotheses are given in Table 4.1 and the observations are plotted in Figures 8.1,8.2, 8.3, 8.4, 8.5 (in the Appendix A) at each transmission power level, which include the probability density distributions. In these figures, the part that is marked with a gray mask shows the subset of data which was selected by the sub-hypotheses 5. These results show that all of the sub-hypotheses could detect the closer node correctly at all of the transmit power levels and the channel was fairly stable even for lower levels of transmit power, an example of which can be seen in the Figure 4.1. However when we changed the distance between the nodes, we have observed that the performance of the algorithms were not consistent.

Table 4.1: Dist: 100 cm, Line of Sight

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	0.98	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.96	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	0.98	Correct
Sub-Hypothesis 3	1	0.92	Correct
Sub-Hypothesis 4	1	0.99	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	0.94	Correct
Sub-Hypothesis 3	1	0.82	Correct
Sub-Hypothesis 4	1	0.93	Correct
Sub-Hypothesis 5	1	0.95	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

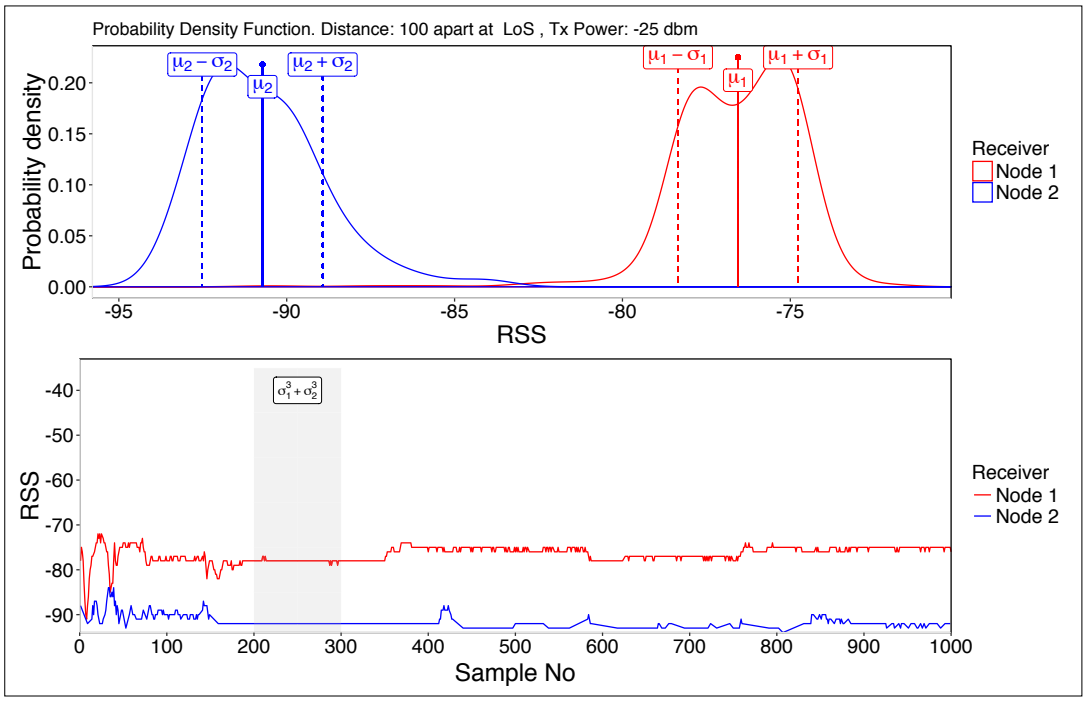


Figure 4.1: 100 cm apart nodes; reference node Tx Power: -25 dBm

The second set of experiments were performed with the same settings as the previous one, except the nodes were placed 200 cm apart from each other. Observations are plotted in Figures 8.6,8.7, 8.8, 8.9, 8.10 (in the Appendix A) for each transmission power level.

The results of the sub-hypotheses from this set of measurements are given in Table 4.2. The Figure 4.2 shows that even when all the controlled parameters (distance, transmit power level, node positions, packet length) stayed same, the channel conditions changed. In the first 200 transmissions the Node 1 registered higher RSS readings than the Node 2. Between about 220^{th} transmission and 680^{th} transmission Node 2 was measuring higher values, and after that Node 1 measured higher RSS values again.

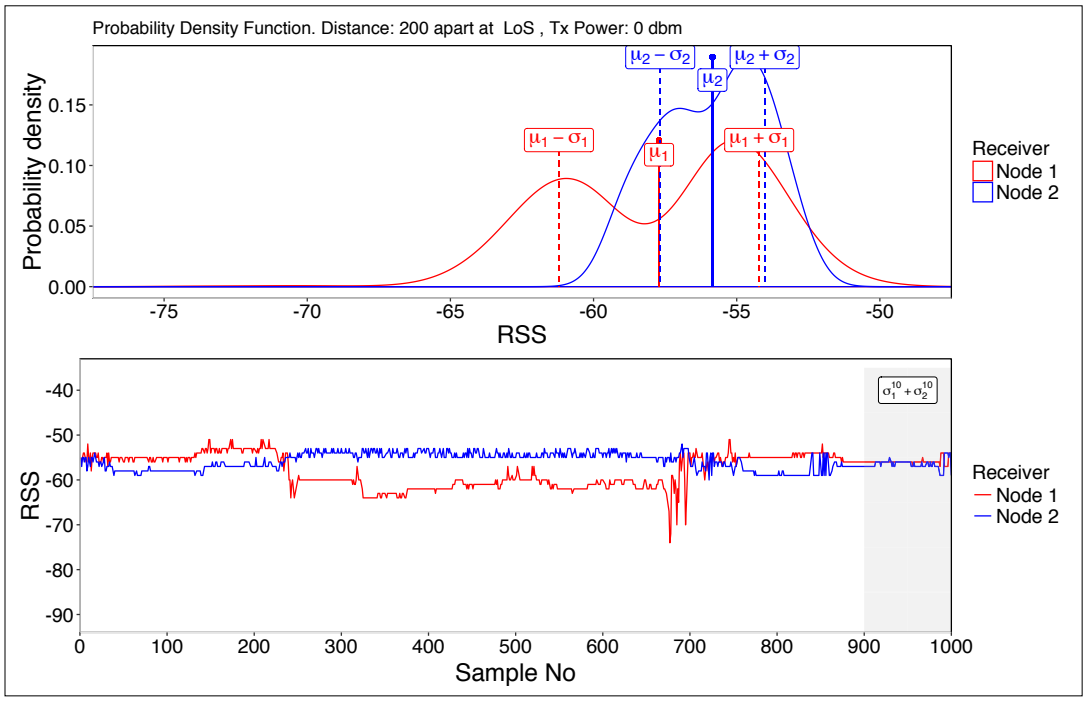


Figure 4.2: 200 cm apart nodes; reference node Tx Power: 0 dBm

These observations showed that the channel did not stay coherent for the complete duration of the measurements and the some indication of fading channel characteristics was noticeable.

The third experiment set was done with NLoS conditions in a lecture room with multiple rows of seats in it. The nodes were put right behind the backsides of the seats, which were 80 cm apart from each other. This means Node 1 and the reference node had one seat in between and Node 2 was two seats away from the reference node. Observations are plotted in Figures 8.11,8.12, 8.13, 8.14, 8.15 (in the Appendix A) at each transmission power. The results of the sub-hypotheses are given in Table 4.3.

Table 4.2: Dist: 200 cm, Line of Sight

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.02**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.05**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.02**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -5 dBm

Hypothesis:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.43**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.34**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.40**	**Incorrect**
Sub-Hypothesis 5	**2**	**0.74**	**Incorrect**

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 2	1	0.02	Correct
Sub-Hypothesis 3	1	0.50	Correct
Sub-Hypothesis 4	**2**	**0.04**	**Incorrect**
Sub-Hypothesis 5	**2**	**0.12**	**Incorrect**

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	0.76	Correct
Sub-Hypothesis 3	1	0.34	Correct
Sub-Hypothesis 4	1	0.76	Correct
Sub-Hypothesis 5	1	0.85	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.07**	**Incorrect**
Sub-Hypothesis 3	1	0.39	Correct
Sub-Hypothesis 4	**2**	**0.12**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Table 4.3: Dist: 80 cm, Non - Line of Sight

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	0.99	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.99	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

In the fourth set of experiments that were performed in the same room, the nodes are placed two seats apart from each other, which was equal to $160\ cm$ in between the nodes (hence the receiver 2 was $320\ cm$ and 4 seats away from the reference node). Observations are plotted in Figures 8.16, 8.17, 8.18, 8.19 and 8.20 for each transmission power level. The results of the sub-hypotheses are shown in Table 4.4.

Table 4.4: Dist: 160 cm, Non - Line of Sight

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	1.00	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

The third and fourth set of experiments at NLoS provided all successful results. Inter-node distances were comparable to LoS experiments that are given above. These experiments are so far not conclusive and it was not possible to recreate the conditions that caused an experiment to be unsuccessful in a repeatable way. However, these results hint that when we had a Trust Level of 0.5 or higher with the sub-hypotheses 2, 3 or 4, their verdict was correct for most of the time. But this is not a conclusive result.

For verifying the observation about the Trust Level threshold, the unidistance setting was changed. We placed the reference node at a different distance (dS) to the first receiver than the distance between the receiving nodes (dN). In other words, the first receiver was dS cm away and the second receiver was $dS + dN$ cm away from the sender (reference node). We performed new measurements with such parameters at LoS and NLoS settings. In one experiment, when the distance from the sender to the closer node was $dS = 100\ cm$ and the distance between the receiving nodes was $dN = 50\ cm$, we observed both successful and unsuccessful results at different transmit power levels, as given in the Table 4.5.

More experiment results at varying dS and dN paremeters are provided in the Appendix A, in Tables 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 8.10 and 8.11.

Table 4.5: Line Of Sight, dS=100, dN=50

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.05	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.09	Correct
Sub-Hypothesis 2	1	0.98	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.96	Correct
Sub-Hypothesis 5	1	0.99	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.02**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.69**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.87**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.71**	**Incorrect**
Sub-Hypothesis 5	**2**	**0.90**	**Incorrect**

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.18	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.20	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

These results show that when the distance between the sender node and the receiver nodes is increased, the verdicts become unreliable. They were giving us wrong results with high trust levels.

It is intuitive to assume that multipath is causing instability when we rely on RSS to determine closeness. Some research, such as [121] [122], indicate that displacing the reference node's antenna for a few centimeters (less than a wavelength distance) should have an impact on the measured values. So, we performed experiments by displacing the nodes for 6 cm and summarized the results in Tables 4.6 and 4.7. These two experiments show that small repositioning of the antenna can have a big impact on the RSS measurements.

Moving the antennas in space changes the propagation path of the transmitted signal and therefore the multipath effects change as well. Using two antennas, or displacing the antennas during the runtime of a network is not practical. Therefore we decided to investigate ways of changing the propagation paths of the signals while keeping the nodes stationary, which is elaborated in the next chapter.

Table 4.6: Line Of Sight, dS=500, dN=50

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.18**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.99**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.97**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.19**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.97**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.15**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.14**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.06**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Table 4.7: Line Of Sight, dS=506, dN=50

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.05	Correct
Sub-Hypothesis 2	1	0.81	Correct
Sub-Hypothesis 3	1	0.55	Correct
Sub-Hypothesis 4	1	0.78	Correct
Sub-Hypothesis 5	1	0.98	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.07	Correct
Sub-Hypothesis 2	1	0.94	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.95	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.05	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.04	Correct
Sub-Hypothesis 2	1	0.85	Correct
Sub-Hypothesis 3	1	0.93	Correct
Sub-Hypothesis 4	1	0.84	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.03	Correct
Sub-Hypothesis 2	1	0.96	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.80	Correct
Sub-Hypothesis 5	1	0.96	Correct

4.2 Summary and Next Steps

In this chapter we have proposed different hypotheses in an attempt to identify the closest node to a particular transmitter (reference) node. We worked with the distribution of the RSS values at a single channel and at different transmit power levels. The parameters that we have considered were the mean of the RSS values, the empirical probabilities of measuring a higher RSS value, and the non-overlapping areas in the probability densities of RSS values. The experimental results of the introduced sub-hypotheses were most of the time indifferent from each other. A few of the cases, in which the results differed from each other (such as all incorrect), could not be reproduced in other settings. Therefore, we refrain from pointing at any of them for being better than the other ones.

The experiments that were performed in various configurations showed that, the closer node was detected correctly for about fifty per cent more frequently than it was detected incorrectly, which was more often the case when the receiving nodes were placed closer to each other than to the reference node (including the results shown in the Appendix A). But we do not consider this difference significant enough to produce a conclusion.

The results have also demonstrated that diversity in the power levels of the transmission did not have an impact on the results. Therefore, we can eliminate the suspicion that diversity in transmission power could help separating closer and farther nodes in space if RSS is the source of information.

We have observed indications that controlled displacement of transmission antenna could help with identifying the closer receiver to the transmitter node. In such small scales, the obvious impact of such a displacement is the changed multipath. Since physically manipulating the nodes in a WSN during its runtime is not practical, we looked for other ways of altering the multipath phenomenon. This reasoning encouraged us to study the impact of the frequency diversity in transmissions as the next step.

Chapter 5

One-Dimensional Node Position Discovery

In the previous chapter, we attempted to utilize RSS information to identify the spatial closeness of the nodes with regard to each other. The hypotheses that were introduced succeeded more often than failed, but identifying the cases when they would succeed or fail was not possible. Analysing the distribution of the RSS values or repeating experiments with varying transmit power levels did not increase the chances in relating RSS to closeness of the nodes. These results are in line with the findings of many other studies in the scientific community, such as [23] or [26]. It was also observed that moving the antenna of the transmitting node only by a fraction of the signal's wavelength impacted the results, as shown in Tables 4.6 and 4.7. Therefore, we started to look deeper into the basic characteristics of wireless signal propagation in indoor settings. RSS is a monotonic function of distance between the transmitter and receiver pairs. But in reality signal strength does not always correlate with the actual geographic distance, due to many reasons like wireless fading and multipath propagation. As it is discussed in Chapter 3, even under stationary conditions RSS can vary in time; which makes it unsuitable for indoor ranging [95]. Regardless, we hold on to our claim that it is possible to recognize the spatial closeness of the nodes relative to each other by using available RSS information, based on the findings that are discussed previously in Section 4.2.

The power of transmitted radio signal attenuates with the distance in space. This means that under ideal conditions, a farther node will measure a lower RSS than a closer node to the source of a signal in LoS settings, because the signal attenuation is a function of distance. The derivative of this function is negative and the absolute value of this derivative decreases as the distance to the transmitter grows. Therefore, in theory, a node that measures the highest RSS value from a particular instance of transmission, must be the closest node to the transmitter among other nodes.

We assume that all transmissions are done at a constant power level Tx. Let us assume that Rx_{AB} is the expected RSS value at node B for the transmission of node A in the ideal case of having no external effects, but only distance related attenuation. Similarly, Rx_{AC} is the ideal RSS value of the reception at node C. If node B is closer to node A than node C is, then the inequality of $Rx_{AB} > Rx_{AC}$ should be true under ideal and isolated conditions. This

phenomenon is, however, severely affected by multipath distortion as it was shown in Chapter 4.

Multipath distortion can be defined as a combination of multipath propagation (multiple paths caused by reflection or refraction of the transmitted signal) and RF interference along this path (constructive or destructive). In this context, we refer this shortly as *multipath*. Multipath can corrupt or destroy the signal, as well as increase or decrease its amplitude at the receiver antenna. Multipath does not affect just a few single transmissions, but it has a more severe and longer lasting impact, such that not even taking the mean of many RSS measurements could neutralize its impact in our previous experiments.

Inevitably, some of these multipath effects will add to the magnitude of Rx value. We denote this effect with Ψ, which is a random variable. So the measured signal strength at node B will be $RSS_{AB} = Rx_{AB} + \Psi_{AB}$, and it will be $RSS_{AC} = Rx_{AC} + \Psi_{AC}$ at node C.

Let us consider a simple three-node sequence discovery scenario (shown by Figure 5.1), in which node A wants to determine whether node B or node C is closer, where all of them are placed d distance apart from each other.

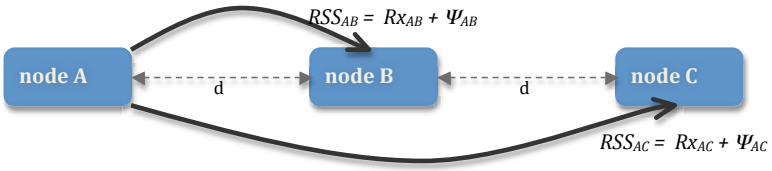


Figure 5.1: Three nodes (A,B,C) and effective multipath (Ψ)

We would be able to recognize the sequence correctly, if $RSS_{AB} >> RSS_{AC}$ was always correct. Therefore we want the inequality given below to be true :

$$\begin{aligned} RSS_{AB} &\gg^{?} RSS_{AC} \\ \Rightarrow Rx_{AB} + \Psi_{AB} &\gg^{?} Rx_{AC} + \Psi_{AC} \\ \Rightarrow Rx_{AB} - Rx_{AC} &\gg^{?} \Psi_{AC} - \Psi_{AB} \end{aligned} \tag{5.1}$$

To make this inequality hold, we can either increase the left side, or decrease the right side of the inequality, and we will both. We increase the difference on the left side, by taking the nearby nodes into account for comparison. Hence, the derivative of the ideal signal attenuation curve remains bigger.

Concurrently, we want to reduce the difference on the right side of the inequality by finding similar values of Ψ at both of the receivers. There is always a chance, that the farther away node can be affected by a bigger constructive multipath than the closer node, or the closer node can be affected by a greater destructive multipath. Yet, *the peak* of the multipath for a transmission by node A is expected to be similar when received by neighboring nodes B and C if they are physically close to each other: $\Psi_{AB}^{max} \approx \Psi_{AC}^{max}$. Let this peak value of multipath be Ψ^{max}. Therefore, to use in our comparisons (Equation 5.2), we are trying to find a value of received signal strength that is close to such a peak magnitude of the RSS at the receiver (RSS^{max}), which is transmitted signal power with attenuation plus multipath effect.

$$
\begin{aligned}
&\text{if } \Psi_{AC}^{max} \approx \Psi_{AB}^{max} \text{ , then } \Psi_{AC}^{max} - \Psi_{AB}^{max} \approx 0 \\
&\quad Rx_{AB} - Rx_{AC} \gg^{?} \Psi_{AC}^{max} - \Psi_{AB}^{max} \\
&\quad Rx_{AB} - Rx_{AC} \gg^{?} 0 \\
&\qquad \Rightarrow RSS_{AB}^{max} \gg^{?} RSS_{AC}^{max}
\end{aligned} \tag{5.2}
$$

To increase the chances of finding an RSS measurement that is affected by a level of multipath, which is close to the Ψ_{AB}^{max} and Ψ_{AC}^{max}, we should seek for enough diversity in RSS that are presumably affected by varying multipath effects. Changing the wavelength of the wireless signal changes the reflection and refraction points inside a room and therefore it changes the paths it travels. Different propagation paths results in different multipath effects, which can be less or more impactful on the original signal. For triggering such a change in the measurements, frequency diversity is introduced to the measurements, which is explained in the following section.

5.1 Frequency Diversity

In Chapter 4 it was suggested that some small spatial displacement of the transmitter antenna could have an impact in correlating RSS to physical proximity, by causing a change in the propagation paths of the signal. Because spatial movement is not always possible during the runtime of a network, we are looking for the ways to change the propagation paths of wireless signals.

When the distance between a node (A) and another node (B) changes, the measured signal strength (RSS_{AB}) should also change for the transmitted signals between them. However, as we discussed previously, such a change is not always correlated to the distance in small distances due to multipath. In [3], the signal strength has been shown as a function of distance and it is shown that the replacement of receiver antenna in factors of the wavelength, causes a big change in RSS gain in both positive and negative directions in mobile channels.

Figure 5.2 is produced by a simulation, explained in [3]. That simulation models a wireless fading channel for a signal at 1 GHz frequency and has a 30 cm wavelength. We therefore speculate that Ψ^{max} can be found by some positional displacement of the transmitter's radio antenna within a wavelength distance. However, it is not possible to reposition the nodes in a network to find the best positions separately. At this point, we want to benefit from the multichannel radio for displacing the propagation path of the transmitted signals. The displacement of a disseminated signal in fractions of a wavelength in high frequencies (such as 2.4 GHz) can have a big effect on aggregate gain, which we examine below.

It is visible in the Figure 5.2 that there is a rough periodicity in the amplitude of the received signal strength that is roughly equal to half of a wavelength. But even such a big movement would not assure achieving Ψ^{max}; sometimes a displacement for several wavelengths would be needed. When using the *CC2420* radio chip, which operates on $2.4\ GHz$ frequency, we need roughly $1.2\,GHz$ bandwidth for observing the whole change in RSS. But we have only $80\ MHz$ bandwidth spread across 16 channels (11 to 26) that are $5\ MHz$ apart [63]. Our hypothesis is

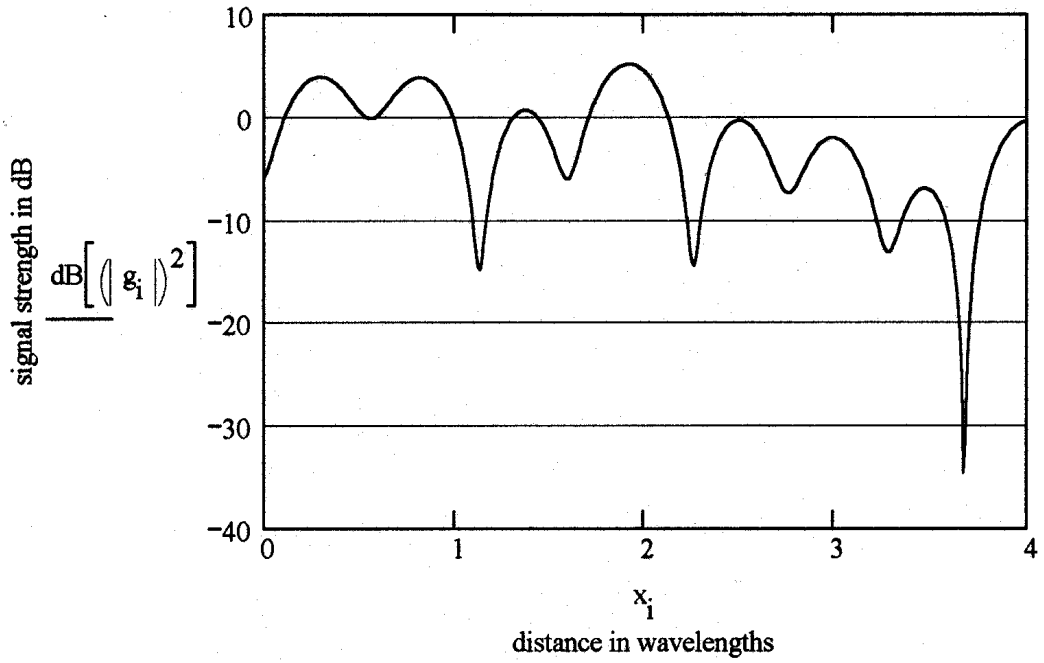


Figure 5.2: Signal Strength as Function of Position [3]
(figure obtained from author's lecture notes)

to combine this limited *frequency diversity* with *time diversity* in order to achieve an acceptably good approximation of RSS^{max}, therefore Ψ^{max}.

In search for the RSS^{max}, we can make many transmissions spread along a window of time in all available channels to have adequate number of samples. Among all those samples, we will select the highest measured RSS as being probably the $Rx + \Psi^{max}$. Since multipath (Ψ) is a random variable, we want to find the maximum that it takes within a certain time for all nodes that are in comparison. Therefore we claim that, if we collect enough samples of RSS data over time, we will have a better chance in reaching (or getting sufficiently close) to one of the peaks that are illustrated in Figure 5.2.

We verified this hypothesis by experiments.

In one experiment, node A is positioned $100\ cm$ away from node B at its line-of-sight and node A transmitted a bulk of modulated IEEE 802.15.4 encoded packets starting from the lowest available frequency level to the highest one in 1 MHz intervals [1] at a constant transmit power. The plotted values in Figure 5.3 are the averages (over 100 measurements) of RSS_{AB} values at each frequency. Here we observe that the difference in the strength of the received signal on different channels can be as big as 25 dBm. This change achieves pretty much the whole scope of the changes in Ψ as theoretically predicted in [3]. In our later experiments, we observed that

[1] The channels, at which the CC2420 radio chip works, are 5 MHz apart, but the radio chip is programmable in 1 MHz steps.

sampling 40 times with 20 ms intervals is just as good as sampling 100 times at the same speed and having 200 samples did not improve the values, either. This observation is consistent with the existing literature within the community working on wireless channel propagation models. In [123] the authors state that for reliably modeling a wireless channel, a sample measurement set of size between 20 and 40 is enough. Their conclusion is also supported by real-world measurements. We also analysed the number of required measurements later in Section 5.4.5 for finding a good sample size for reliable results.

Insights into frequency diversity was also extensively studied in [124]. It was confirmed that the wireless medium does not stay constant over time and the best channel for increasing packet reception between pairs of nodes was analyzed. The authors also confirmed that the wireless link quality shows large variation not only over frequency, but also over time. This conclusion is consistent with the findings in this thesis. Hence combining the frequency diversity with time diversity is promising for measuring close Ψ magnitudes at nearby nodes, therefore we can reduce the right side of Equation 5.1.

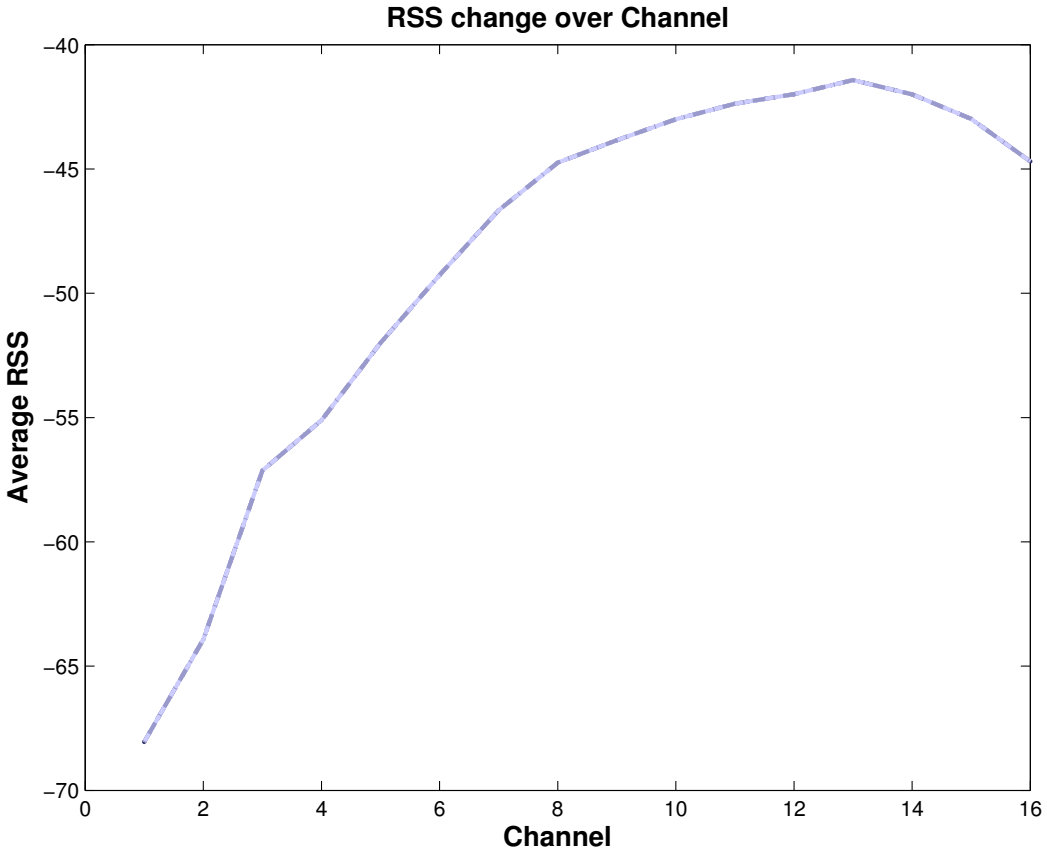


Figure 5.3: Change in RSS between two nodes over 16 channels at 2.4 GHz using CC2420

The same experiment was also repeated for different orientations of the receiver antenna,

which is illustrated in the Figure 5.4. We can see that the antenna orientation does not affect the measurements much, as the peak RSS values for different orientations are close to each other. Hence, the approach is suitable for two-dimensional position discovery purposes, as well.

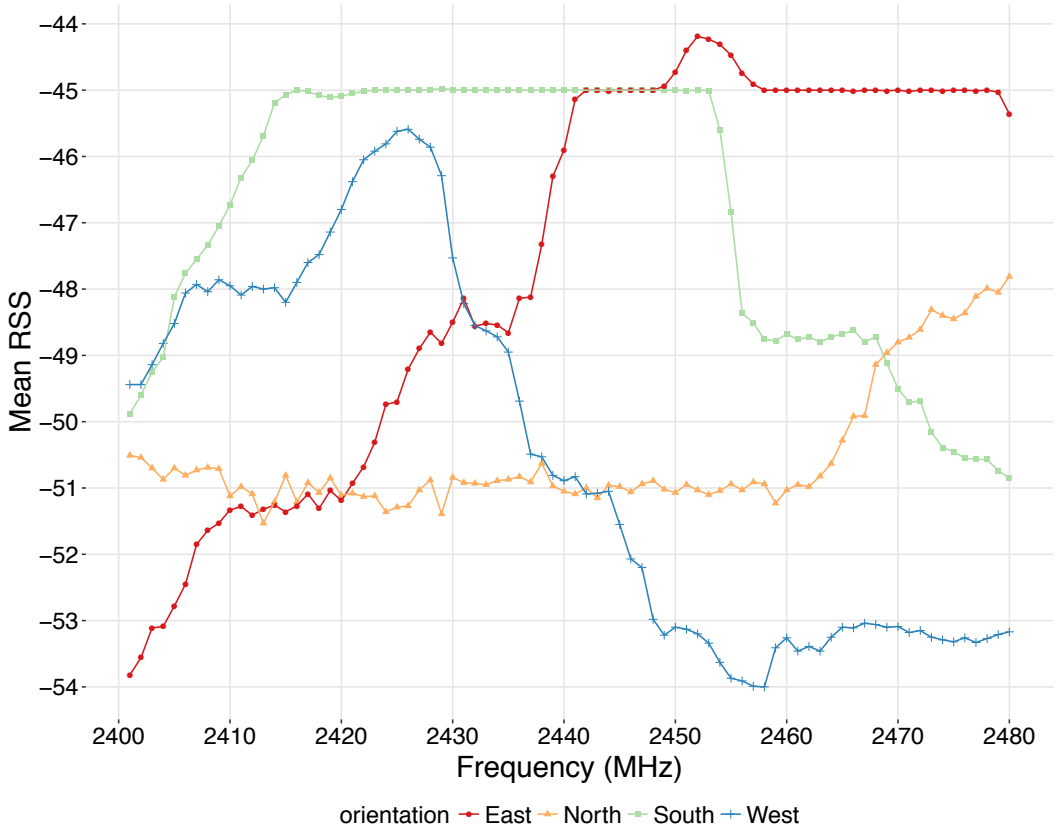


Figure 5.4: Change in RSS between two nodes over all available frequencies at 2.4 GHz by CC2420 radio chip of a TelosB node. Each dataset (by color) represents a different orientation: north, south, west, east.

With these experiments we understand that, by finding the maximum RSS value that we measure from a set of transmissions over time and frequency from node A to node B and node C, we can reduce the randomness in the impact of multipath and have a stronger indication about the closer node by comparing the RSS^{max} values of receiving nodes.

5.2 Detecting Closeness with Frequency Diversity and Position Computation

In the previous section we discussed how the change in measured RSS values can be created by involving frequency diversity in the measurements. This change can be used to reduce the

comparative impact of multipath at the receiver nodes that are positioned close to each other and can help detecting the closer node to a sender. If the closer one of two or more receiver nodes can be detected, then a linear sequence of nodes in space can be constructed by starting from one head of the node deployment.

We start with the assumption of having the information on the identity of the node that occupies the first position in the sequence, which we call the **reference node**. The reference node is the first sender node and performs z transmissions at all available radio channels according to the Algorithm 2. The receiver nodes measure the RSS and each one reports the highest RSS measurement associated with that particular transmitter by using the Algorithm 3. This information is then collected from the receiver nodes and the closest receiver node to that particular sender is centrally selected with the Algorithm 4.

Algorithm 2 for Sender node

for $firstChannel \rightarrow lastChannel$ **do**
  for $i = 1 \rightarrow z$ **do**
    $Radio.Send(Packet_i)$
  end for
end for

Algorithm 3 for Receiver nodes

$Y \leftarrow THIS\_NODE\_ID$
for $firstChannel \rightarrow lastChannel$ **do**
  $i \leftarrow 0$
  while $i < z$ **do**
    $Packet\ pkt \leftarrow Radio.Receive()$
    $i \leftarrow getOrder(pkt)$
    $X \leftarrow getSenderId(pkt)$
    $rss \leftarrow getRss(pkt)$
    if $rss > RSS_{XY}^{max}$ **then** ▷ $RSS_{XY}^{max} = (Rx_{XY} + \Psi_{max})$
      $RSS_{XY}^{max} \leftarrow rss$
    end if
  end while
end for
Report RSS_{XY}^{max}

Algorithm 4 Decision algorithm

$CloserNode \leftarrow arg\ max(RSS_{AB}^{max}, RSS_{AC}^{max}, ...)$

If there are more than three nodes in the deployment, this sequence detection system is reiterated for constructing a sequence from closeness information. The roles are then changed and every node in the sequence temporarily becomes the sender (in turns) while others, as well as the previous senders, remain as receivers.

If there are only three nodes in the network, comparing the RSS^{max} for two adjacent nodes can indicate the closer node of the two. But if there are more nodes, whose positions are unknown, a simple comparison or sorting of these RSS^{max} values would not be reliable. The signal attenuation is an inverse exponential function of distance. When the distance from the transmitter is increased, the difference between the theoretical RSS (Rx) values decrease and the effects of multipath become more prominent. Therefore the transmissions only from the reference node is not enough for having a strong decision about the sequence. More measurements must be taken from the farther nodes in the sequence.

In a network of m nodes, above procedure can be applied to compute the spacial sequence of nodes as following:

In turns, each node becomes a sender once. After each sender node N_i $(1 \leq i \leq m)$, collected the maximum RSS measurements from each of its receivers, a sequential vector of neighborhood is created by using the sorted RSS measurements. This vector is called *Proximity Vector (PV)*.

$PV_i = [N_j, ..., N_k, ..., N_m]$, where

$$RSS^{max}_{N_iN_j} > RSS^{max}_{N_iN_k} > RSS^{max}_{N_iN_m}$$

In this sequence, the nodes that are closer to the head position are more likely to be geographically closer as well, since they measured a higher RSS^{max} than the nodes that are farther from the head node in the PV_i.

For the final decision, we take the reference node (N_R) and assign it to the first place in the sequence and flag it. Then starting from PV_R, we take the first unflagged node (N_t) and continue iterating with the vector of that node (PV_t). This procedure is given in Algorithm 5.

Algorithm 5 Sequence discovery algorithm

```
*PV ← Set of Proximity Vectors
*N ← Set of All Nodes
S ← {} // Empty Sequence
N_R ← N.getReferenceNode()
N_t ← N_R
while UnflaggedExists(*N) do
    S.add(N_t)
    N_t.setFlag(true)
    N_t ← firstUnflaggedIn(PV_i)
end while
return S
```

5.2.1 Evaluation

We have verified above hypothesis with real world experiments. In the experiments, we chose the closest node for each node by using the algorithms that are described above. The first node in the node sequence has been taken as the reference node. Then, iteratively, we sorted the

remaining nodes into a single-dimensional sequence. All experiments were done indoors, in the same building that was used in the previous measurements, where the impact of multipath was noticeable.

In one experiment, we placed three TelosB nodes in a single-dimensional setting 100 cm apart from each other. We collected RSS^{max} values in all available 16 channels that the CC2420 radio operates at. Each transmission is repeated 100 times. The nodes were assigned IDs $5,\ 6,\ 7$ and they were positioned in the order $5-7-6$. The reference node was Node 5 and it was at the first place in the node sequence. Hence Node 5 was the transmitter while the nodes 6 and 7 were the receivers at the first iteration. After having collected the RSS data from the measurements, we sorted them and found the correct ordering as being $5-7-6$. The measurements of the experiment are shown in the Figure 5.5.

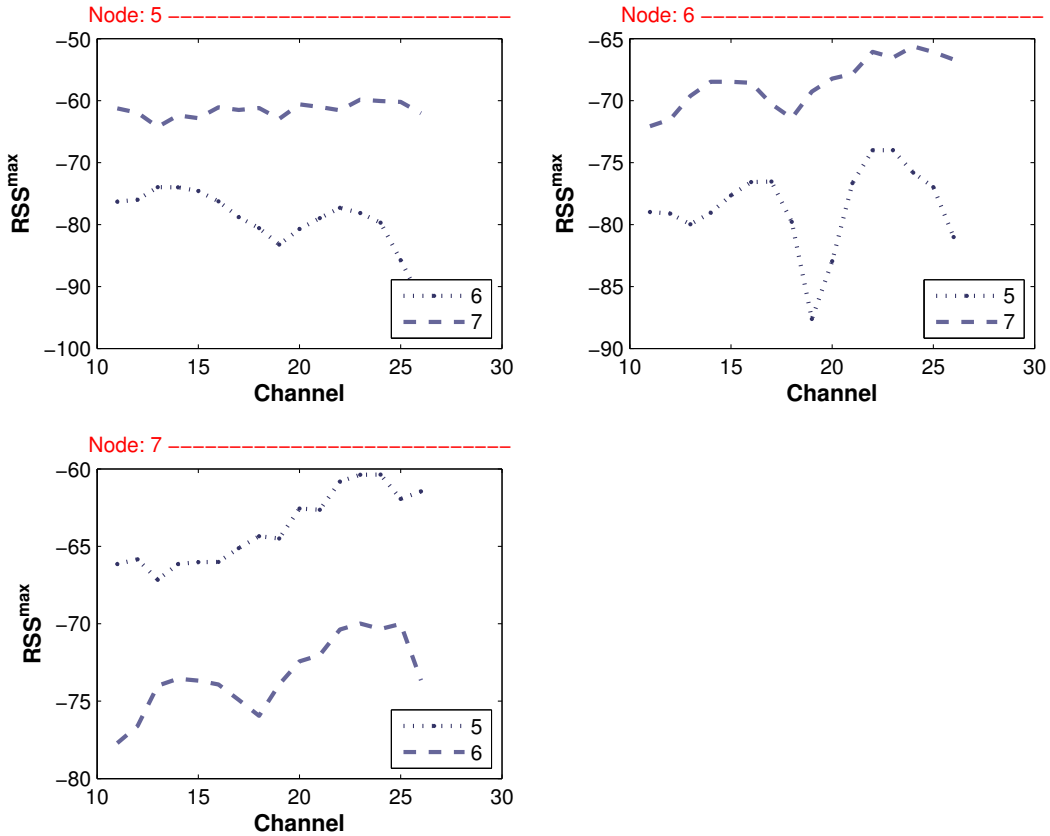


Figure 5.5: Experiment output gives the correct ordering: 5-7-6

The correct ordering is found, but the plotted measurements also showed that at each channel the closer receiver to the reference measured higher signal strength than the farther receiver. This was a case that any of the hypotheses from Chapter 4.1.1 would also detect the closer node. However when we examine the measurements while the middle node (7) was the sender, we

observe a greater difference in the measured signal strength values from the same distances. This can have two reasons; either the impact of multipath was very severe at least at one of those two different points in space and in that time, or the RSSI modules of the radio chips were not calibrated. We eliminated the second option because the variation of RSSI readings across different devices with the same chipset does not cause a difference as big as 12 dBm. We have also discarded a possible calibration problem with our further experiments. Other research, such as in [125], on the topic of mismatching RSSI values and node calibration also confirms that calibration error does not cause such big differences in RSS measurements.

We extended the same experiment to four nodes, having a sequence of nodes with IDs $5 - 7 - 6 - 8$ relatively and the Node 5 was the reference node. In these experiments, the node placement was done irrespective of node IDs intentionally to prevent any false positive verdicts due to software error. After applying the Algorithms 2 to 5, we were again able to find the correct ordering with the measurement data shown in Figure 5.6.

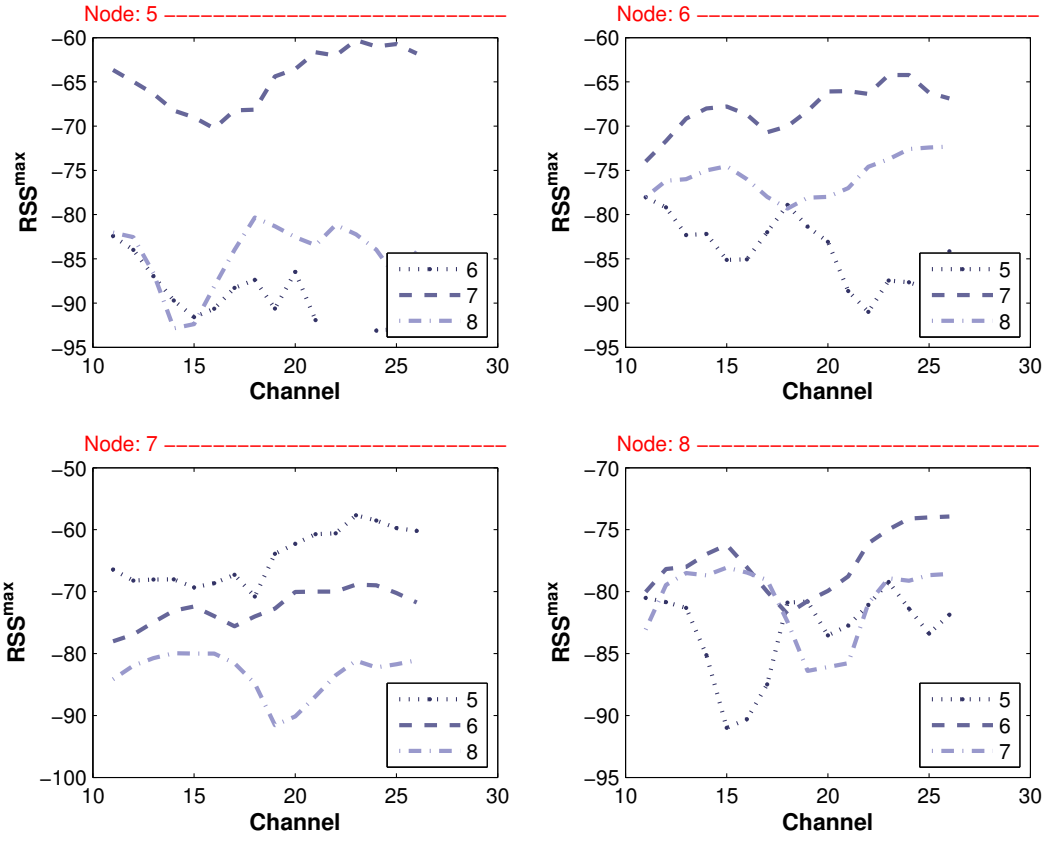


Figure 5.6: Experiment output gives the correct ordering: 5-7-6-8

In another experiment we used 5 nodes with IDs $5,\ 6,\ 7,\ 8,\ 9$ and placed them in the sequence $5 - 6 - 7 - 8 - 9$ while decreasing the distance in between nodes to 50 cm, and we

again assigned Node 5 as the reference node. After following the suggested methodology, the outcome of this experiment was again the correct sequence. The measurement data from that experiment are shown in Figure 5.7.

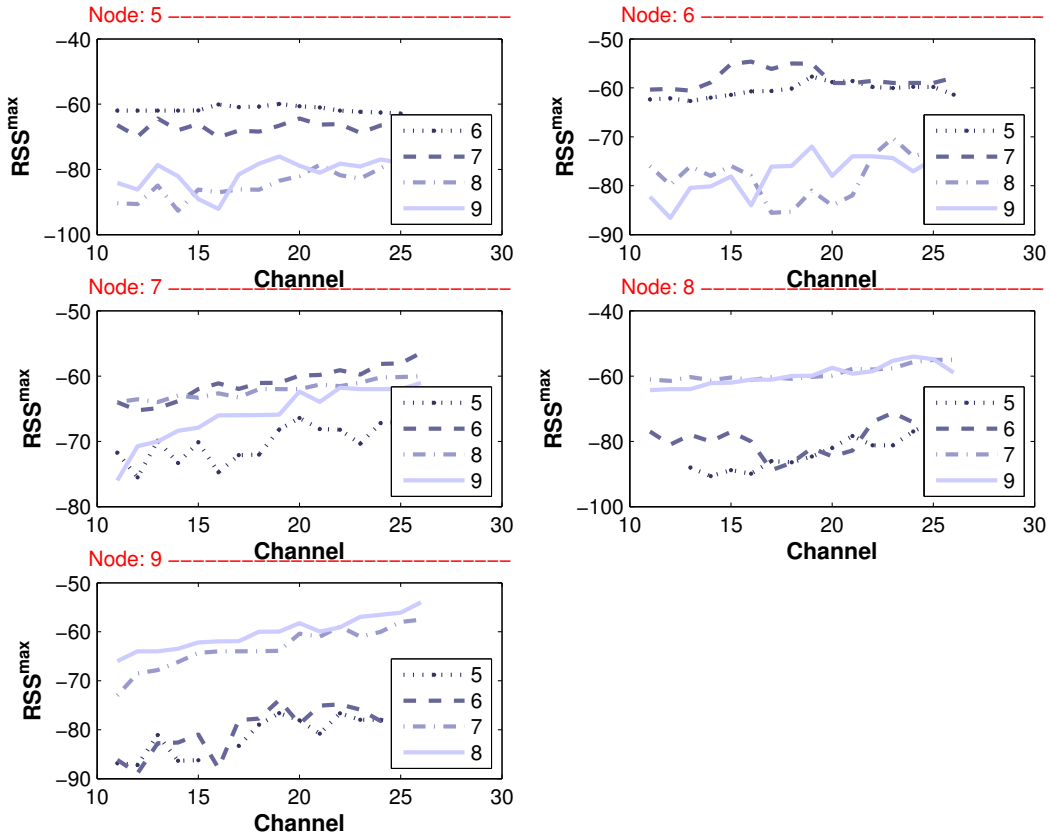


Figure 5.7: Experiment output gives the correct ordering: 5-6-7-8-9

Even though in most cases the suggested method performed very well, letting us to discover the correct sequence of nodes, there has been few cases in which the sequence of two of the neighboring nodes were swapped. Such as, the data illustrated in Figure 5.8 has given us the ordering as $5-7-6-8-9$ with Node 5 being the reference node, while the correct ordering was $5-6-7-8-9$. This experiment was done with the same settings and at the location as the other experiments and the internode distance was $100cm$. However, when we took the reference node as Node 9, we could find the correct ordering as $5-6-7-8-9$.

To further verify the effectiveness of frequency diversity on closeness information, we performed two other sets of experiments. We placed 5 nodes 50 cm apart from each other in a small, multipath-prone room. Each node transmitted 40 packets with 20 ms intervals, only on Channel $11(2405\ MHz)$ and we sorted the nodes according to the methodology in subsection 5.2. We repeated this experiment 10 times. Later, we repeated the same set of experiments

with the complete channel diversity, which is 16 channels, without physically manipulating the nodes.

In these experiments, we positioned Node 0 as the first node (reference node) in the sequence and positioned nodes $1, 2, 3, 4$ next to it respectively, each node was placed 50 cm farther than the previous one. Hence, if the output of our algorithm was the sequence $\{0\ 1\ 2\ 3\ 4\}$, then the verdict was True (T), otherwise False (F).

In Table 5.1, each row is the result of a separate experiment. We see that in single channel case none of the 10 experiments resulted in correct sequence, while in multiple channel experiments 9 out of 10 sequences were correct. It is also noticeable that in the incorrect verdict of multiple channel experiments, only one node was not found in its respective position.

5.2.2 Summary

In this section, we suggested an approach for determining the closest node within multiple nodes to a particular reference node and discovering the node sequence in a single-dimensional space.

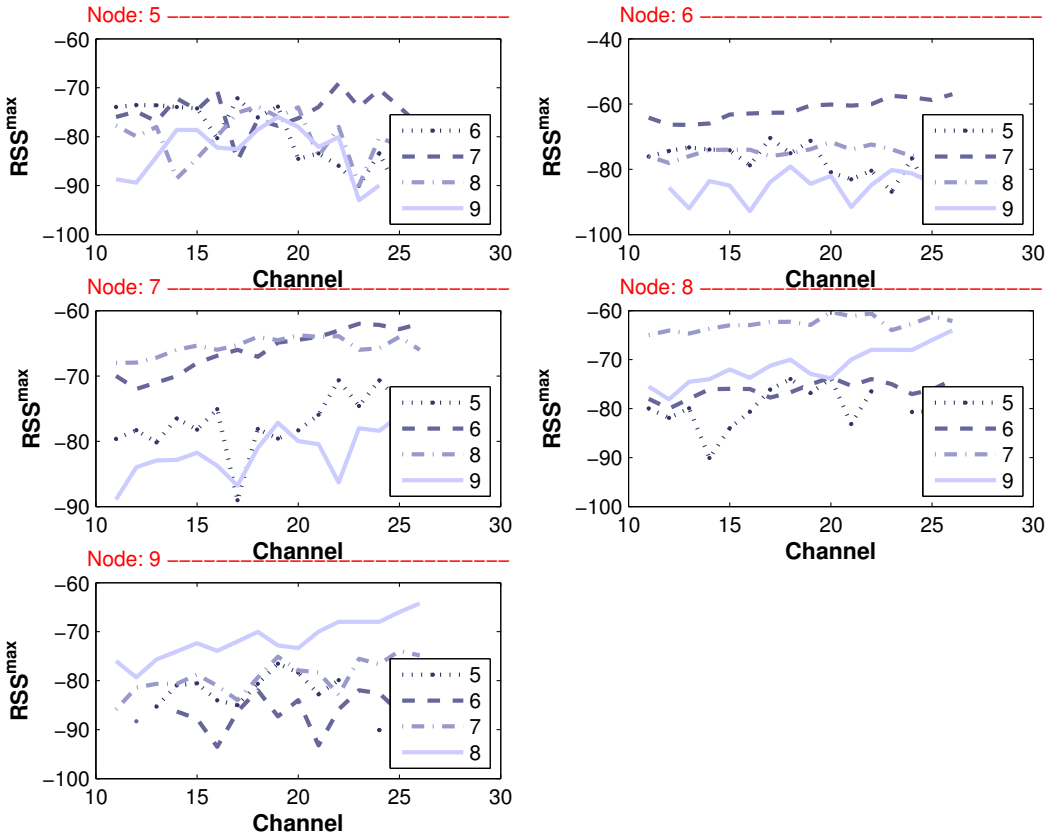


Figure 5.8: Experiment output gives the ordering slightly wrong as: 5-7-6-8-9

	Single Channel		Multiple Channels	
Exp. No:	Detected Sequence	Result	Detected Sequence	Result
1	0 2 1 4 3	F	0 2 3 4 1	F
2	0 2 1 3 4	F	0 1 2 3 4	T
3	0 2 1 3 4	F	0 1 2 3 4	T
4	0 2 1 3 4	F	0 1 2 3 4	T
5	0 2 1 3 4	F	0 1 2 3 4	T
6	0 2 3 4 1	F	0 1 2 3 4	T
7	0 2 1 3 4	F	0 1 2 3 4	T
8	0 2 1 3 4	F	0 1 2 3 4	T
9	0 2 1 4 3	F	0 1 2 3 4	T
10	0 2 1 3 4	F	0 1 2 3 4	T
Total:	**0/10 Success**		**9/10 Success**	

Table 5.1: Experiment results: Single-Channel vs Multiple-Channel

We use RSS samples from transmissions at all available channels that the radio chip allows us to operate at. For decision criterion, we take the highest measured RSS across all channels between a node pair and by comparing these values with the values from all other pairs we give a binary decision for choosing the closest node. Starting from one known reference node, we iterate sorting the nodes in the sequence until positions of all nodes are computed. In our experiments we had high success in sequence discovery with occasional error of swapped adjacent nodes, while the overall sequence was still being correct.

This part of the study was demonstrated to technical and non-technical audiences in two occasions. In first occasion, we performed our experiments at the research center of Airbus, who sponsored the research during that time, and the system was demonstrated for detecting positions of the passenger seats in a life-like mock-up of an airplane fuselage. This plane mock-up was constructed with the same materials of a real airplane with the identical architecture. The fuselage had five rows of seats with 80 cm spacing in between. Five nodes were placed on the armrests of the seats in consecutive rows and these rows were discovered. The repetitive results of the sequence discovery were correct and the audience has confirmed. The demonstration was repeated with the nodes inside the life vest drawers of the seats and the results reflected the correct rows of the seats again.

The second demonstration was performed at an event called *Lange nacht der Wissenschften (Long night of science)* in Berlin, at a hall of the Technische Universität Berlin in Germany. Five nodes were placed in cardboard boxes and they were shuffled. After running the introduced sequence discovery system in this section, the audience could confirm that the results mapped to the actual spatial sequence of the nodes. These boxes were 50 cm in width.

The proposed system should be seen as an evidence for the possibility of successful discovery of relative node positions in a linear setting. Above results have shown that the suggested system works both for LoS and NLoS settings. The proposed method and the results were also published in the peer-reviewed paper [126]. For a better analysis for robustness, we used the

TWIST testbed for the following experiments. In TWIST testbed, the nodes have walls in between and it is deployed in a building where various types of research on wireless networking systems are performed, therefore the wireless medium is very unpredictable. The suggested sequence discovery system did not perform well in the testbed, which we speculate due to the reasons that are just mentioned. Therefore we decided to investigate further and develop better algorithms that would accommodate challenging situations like the TWIST testbed was exposed to.

In the following sections, we will extend the experiments to an increased number of nodes and we will consider NLoS conditions that come with the testbed, which will bring us more challenge in discovering the node sequence. The use of a testbed also allows us to perform more number of experiments faster and create a base for comparing and quantifying the quality of proposed systems.

5.3 Iterative Sequence Building

In the previous section it was shown that frequency diversity enables discovery of the closest node to a particular sender and how this information can be used to deduce the node sequence. We claim that for a reliable sequence discovery —even in heavily faded indoor environments where the network is dense and localization is tough— it is enough that each node sends a limited number of beacon messages and that these messages are received by as many other nodes as possible (but not necessarily all), while recording the RSS of the incoming transmissions at multiple radio channels.

In this section, we suggest methods for statistical processing of the collected RSS measurements to construct the node sequence in a more robust way and we investigate the impact of different parameters and decision criteria on the performance of sequence discovery methods. Finally, we demonstrate the high precision of our selected approach in developing the node sequence, leveraging experimental data collected in a WSN testbed. The techniques and the results in this section are published in our paper [112].

We will focus on developing a more robust way of node sequence discovery mechanism, which will handle more challenging NLoS conditions better than the system that was introduced in the previous section. The nodes, whose relative positions need to be discovered, are considered to be placed roughly along a straight line, in a single-dimensional grid setting. The cells of the grid are assumed to be occupied with only one node of the network and no cell is assumed to be vacant. The cells (or the node positions) might or might not be separated by obstacles, such as walls or furniture. The prerequisite of knowing which node occupies the first position in the grid is still there and the positions of the rest of the nodes need to be discovered. The nodes are assumed to contain radio interfaces that can operate on multiple frequencies and they can report RSS values upon packet reception.

Further, it is also assumed that the nodes are not mobile and they transmit at a relatively high power setting, increasing the chances of correct packet reception. That is also because we would like to allow the use of normal data packets for our approaches and hence produce less overhead for sequence discovery.

In this part of the work, we approach the problem of sequence discovery in three discrete

steps: *1. Measurement of RSS values*, *2. Proximity Vector generation* and *3. Sequence Building*. In the first step the RSS measurements are taken in all channels available to the radio hardware. In the second step, we use the RSS values that are measured by the receiver nodes upon packet reception, to derive a **rough estimate** of the closeness (in space) between the nodes. The vector of these estimates, the *proximity vector*, contains a single node's perspective, which might or more likely might not, be correct. Then in the *sequence building* step, we fuse these local views in a global view, giving the whole sequence.

We will take the MDS algorithm, which was explained in Section 2.5, as an alternative way of creating the node sequence from the RSS information and it will be used in the comparative analysis.

5.3.1 Measurement Procedure

This is the first step to sequence building as illustrated in Figure 5.9. This procedure is based on the measurement method that is explained in Section 5.2.

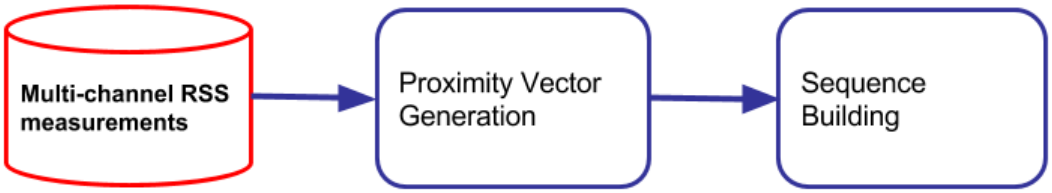


Figure 5.9: Initial step of relative position discovery

All of the presented methods in the rest of this thesis share the same data collection stage as follows:

1. n nodes are assumed to be placed in a grid with n cells
2. Each node in set $*N = \{N_1, N_2, ..., N_n\}$ is set to the first channel C_1 from the set of channels $*C = \{C_1, C_2, ..., C_{max}\}$
3. Each node $N_i \in *N$, transmits z packets per each channel C_1 *to* C_{max}
4. Each node that is in the transmission range of N_i records the measured RSS from the received packets at each channel C_1 *to* C_{max}. All participating nodes change the radio channel synchronously
5. After node N_i completes transmission z at channel C_{max}, all nodes move back to the first channel (C_1) and next node in $*N$ begins transmitting
6. The procedures 3 to 5 are repeated for each one of the n nodes in the set $*N$

This procedure does not need to be separated from the regular data transmission of a WSN, as long as nodes use as many frequencies as possible during their communication. For the sake of simplicity in explaining the algorithms, we will assume that all the information, collected by the above procedure, is made available centrally for processing. We will elaborate on this at the end of Section 5.3.3.

We also assume that the nodes are roughly synchronized to change channel at the same time. This level of synchronization does not need extra transmission overhead. The sequence number and the total number of packets are provided in each transmission. The nodes can utilize a software alarm, which can be reset to the next channel switch time upon reception of each packet. Thereby, nodes will roughly know when it is time to switch to the next channel. And finally, an extra guard window of delay after each channel change and before start of new transmission allows the nodes to settle to the same channel even when some timer drift occurs. This is not a strict timer synchronization, but a technique to trigger channel switch inside a network at the same time. This has been the actual procedure that we used in our demonstrations.

5.3.2 Proximity Vectors

This is the second step of iterative sequence building system, as illustrated in Figure 5.10.

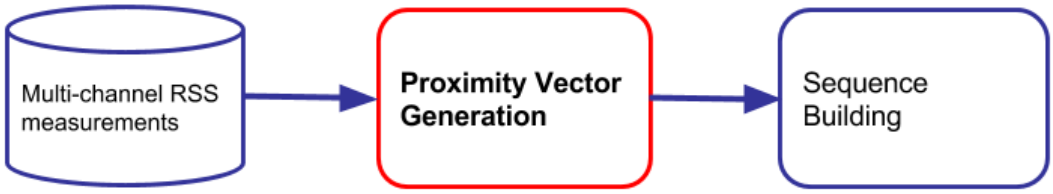


Figure 5.10: Proximity vector generation step

In our algorithms, we will iteratively build the sequence by utilizing Proximity Vectors (PV), which contains the set of nodes from a single node's perspective in the order of estimated relative proximity. The Node i will have the proximity vector PV_i, which roughly indicates that the closest neighbor is the N_{i^1}, the second closest neighbor is the N_{i^2} and so on.

$$PV_i = \{N_{i^1}, N_{i^2}, ..., N_{i^{n-1}}\}$$

We suggest three methods to create proximity vector PV_i for any given node N_i in an n node setting. First method takes the highest RSS measurements as suggested in the previous section, the second method takes the average of all RSS measurements, which is applying the frequency diversity to the first hypothesis in the Section 4.1.1, and the third method is a fusion of both, which considers the average of a set of maximum RSS values that are measured from a particular sender.

Maximum of all RSS readings (Max RSS)

For each transmitter N_i, the geographically closest node N_r is chosen by comparing the maximum RSS values that are measured by each of the receiver nodes within its transmission range.

For relating RSS measurements to closeness, we compare the RSS readings of the nodes from the same sender, but not necessarily from the same transmission. By transmitting packets at different frequencies, the propagation paths of the transmissions are altered. The receivers can measure very different values at different frequencies, while variance is very small within the same frequency. Here the highest measured RSS value from the same transmitter is selected from each node to compare. Each sender assumes that one of the many receivers, which reports the highest RSS, is the geographically closest node to itself and places that node into the first place in it's *PV*.

In other words, if the maximum RSS value, across all channels, that node N_j measures is greater than the maximum RSS that node N_k measures from the same transmitter node N_i, then it is more probable that node N_j is closer to node N_i than node N_k is. Consequently, N_j is put at a prior position in the proximity vector PV_i for node N_i.

This procedure is iterated for all the receiver nodes of the same sender by finding the next biggest RSS value that is received by each of them until all receivers are placed in the proximity vector. For each node N_i in the network, a separate PV_i is created using the same procedure.

We can formulate the above procedure to a network with n nodes as below:

Each node N_i $(1 \leq i \leq n)$ collects the maximum RSS measurements from its receivers, while N_i was the transmitter. Its proximity vector (PV_i) is then created such that:

$$PV_i = [N_j, ..., N_k, ..., N_m] \text{ , where}$$
$$RSS^{max}_{N_iN_j} > RSS^{max}_{N_iN_k} > RSS^{max}_{N_iN_m}$$

In PV_i, the nodes with smaller indices are considered being geographically closer to node N_i too, since they measured a higher RSS^{max} than the nodes that are farther from the transmitter node in the setting.

In case there are receivers with equal values of RSS^{max} from a particular sender i that are competing for the position p in the PV_i, such that

$RSS^{max}_{N_iN_j} = RSS^{max}_{N_iN_k} = K$

we count the number of times the value K is measured at any channel by each of these receivers and put the one that has measured this value more often in the prior position in PV_i.

$$N_{i^p} = \begin{cases} N_j & if\ count(RSS^{max}_{N_iN_j}) >= count(RSS^{max}_{N_iN_k}) \\ N_k & if\ count(RSS^{max}_{N_iN_j}) < count(RSS^{max}_{N_iN_k}) \end{cases}$$

Average of all RSS values (Avg RSS)

As an alternative to the method above, we also compute the average of all measured RSS values for the same transmitter to construct the proximity vectors. In this case, the proximity vector

PV_i of node i is formed as:

$$PV_i = [N_j, ..., N_k, ..., N_m] \text{ , where}$$

$$\overline{*RSS_{N_iN_j}} > \overline{*RSS_{N_iN_k}} > \overline{*RSS_{N_iN_m}} \text{ , where}$$

$$\overline{*RSS_{N_iN_j}} = \frac{\sum_{t=1}^{r^{ij}} RSS^t_{N_iN_j}}{r^{ij}}$$

and r^{ij} is the total number of packets that N_j received from N_i.

Average of maximum $\theta\%$ of RSS Values (Avg Max RSS θ)

In this approach, average of a subset of the sorted RSS values are used to construct proximity vectors. This subset contains the highest $\theta\%$ values of all RSS values across all channels for each sender-receiver pair.

The set of these values are denoted by $*RSS_{ij}^{max(\theta\%)}$ for the sender N_i and receiver N_j, and the proximity vectors are constructed as:

$$PV_i = [N_j, ..., N_k, ..., N_m], \text{where}$$
$$\overline{*RSS_{ij}^{max(\theta\%)}} > \overline{*RSS_{ik}^{max(\theta\%)}} > \overline{*RSS_{im}^{max(\theta\%)}}$$

This subset is illustrated in Figure 5.11.

Figure 5.11: Schematic representation of processing RSS measurements

5.3.3 Sequence Building

The next step (Figure 5.12) in the position discovery process involves identification of the node sequence, given rough estimates for the relative distances between node pairs, in the form of the proximity vectors. Starting with the reference node, The final sequence is constructed using either one of the two algorithms described below.

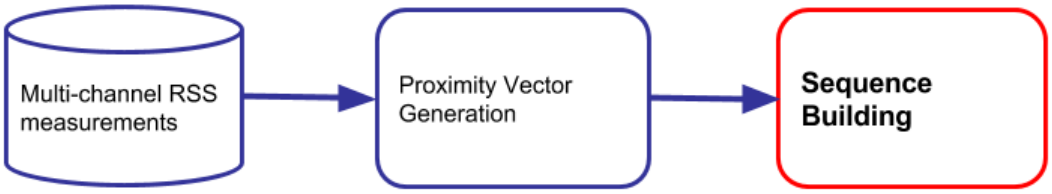


Figure 5.12: Sequence building step

Greedy Sequence Building

This algorithm starts with placing the reference node N_R into the first position of sequence vector, S. Then, the first node in PV_R is added to S. Each time a new node is added in S, the first unplaced node in its proximity vector is inserted in the next position and the placement is iterated until all nodes are added to S. The procedure is outlined in Algorithm 6.

Algorithm 6 Greedy Sequence Building

$*PV \leftarrow Set\ of\ Proximity\ Vectors$
$*N \leftarrow Set\ of\ all\ Nodes$
$S \leftarrow \{\}$ // Empty Sequence
$N_i \leftarrow N_R$ // Starting from the Reference Node
$S.add(N_i)$
for 1 to $(|*N| - 1)$ **do**
  $N_i \leftarrow firstUnplacedNodeIn(PV_i)$
  $S.add(N_i)$
end for
$Return\ S$

For a network with n nodes, first node will choose the next node among $n-1$ nodes, second node will choose among $n-2$ nodes and so on.. So total number of iterations is:

$$(n-1) + (n-2) + (n-3) + ... + (n-n-1) = \frac{(n-1)n}{2}$$

The greatest term in the sum of iterations is n^2, hence the complexity of this algorithm is $O(n^2)$.

Greedy Lookback Sequence Building

This is an enhancement over previous basic greedy sequence building algorithm. Our observations have shown us that most of the wrong results are caused by very small differences in RSS values. Therefore we set a threshold to tell a farther node from a closer one. This threshold depends on the RSS Indicator resolution of the radio chip. If the difference between the values (we call them *score*), that are used to produce the proximity vectors (such as MaxRSS, or AvgRSS),

between the two best candidates (unplaced nodes with smallest indices) is less than the threshold, then we ask the node that is placed in the growing sequence in the previous iteration. If that node doesn't have a difference greater than the threshold either, then we iterate this querying until we find a node that has this difference bigger than the threshold or we reach to the first node. If we reach to the very first node in the sequence, then it chooses a node regardless of the threshold. The algorithm is as below:

Algorithm 7 Greedy Lookback Sequence Building

```
*PV ← Set of Proximity Vectors
*N ← Set of All Nodes
S ← {} // Empty Sequence
Threshold ← (RSSI Resolution of Radio Chip) × 2
N_i ← N_R // Starting from the Reference Node
S.add(N_i)
for 1 to (|*N| − 1) do
    N_i ← S.getLastNode()
    score1 ← firstUnplacedNodeIn(PV_i).score()
    score2 ← secondUnplacedNodeIn(PV_i).score()
    while (score1 − score2 < Threshold) AND N_i ≠ N_R do
        N_i ← S.get(S.indexOf(N_i) − 1)
        score1 ← firstUnplacedNodeIn(PV_i).score()
        score2 ← secondUnplacedNodeIn(PV_i).score()
    end while
    N_i ← firstUnplacedNodeIn(PV_i)
    S.add(N_i)
end for
Return S
```

In a network with n nodes, if no node needs to look back, in the best case, then the complexity of this algorithm will be same as the previous one: $O(n^2)$. In the worst case, all nodes will need to look back until the first node in the growing sequence. Therefore, for the last decision, the last node in the sequence will do $n-2$ queries. Second last node will do $(n-3)+(n-3)$ queries, third node will do $(n-4)+(n-4)+(n-4)$ and so on. So total number of queries is $(n-2)+2(n-3)+3(n-4)+...+(n-2)(n-(n-1))$. The non-constant parts of this series is:

$$n+2n+3n+...+(n-2)n = n\frac{(n-2)(n-1)}{2}$$

The greatest term in this equation is n^3. Hence the complexity of this algorithm is $O(n^3)$.

These algorithms assume a centralized computation scheme. In a wireless setting it requires delivering the RSS measurements of all nodes to a sink node. This overhead can be costly if we are dealing with a big network. It is also possible to apply these algorithms in a distributed manner, using few extra control packets. In this case, each node N_i will create its own proximity vector PV_i. At each iteration of sequence building, the growing sequence vector must be

sent to the last added node in the sequence vector, so it has the information of placed and unplaced nodes. Finally, the last node in the iterations will initiate the propagation of the finalized sequence vector into the network.

5.3.4 Evaluation

We have experimentally evaluated the suggested methodologies in the TWIST testbed environment. The testing area is exposed to various types of wireless traffic, mostly IEEE 802.11b/g/n, IEEE 802.15.4 and Bluetooth, and has relatively high density of metal structures. The testbed is also spread over the offices, so occasional change in the environment was expected but the nodes stayed stationary.

We selected two sets of nodes in the testbed: nine TmoteSky nodes in the third floor of the TWIST building and ten TmoteSky nodes in the fourth floor of the building. All nodes in each of these sets are uniformly positioned along a line. The nodes are 3 m apart and there is one wall in between each adjacent node pair. The testbed setting with nine nodes is illustrated in Figure 5.13.

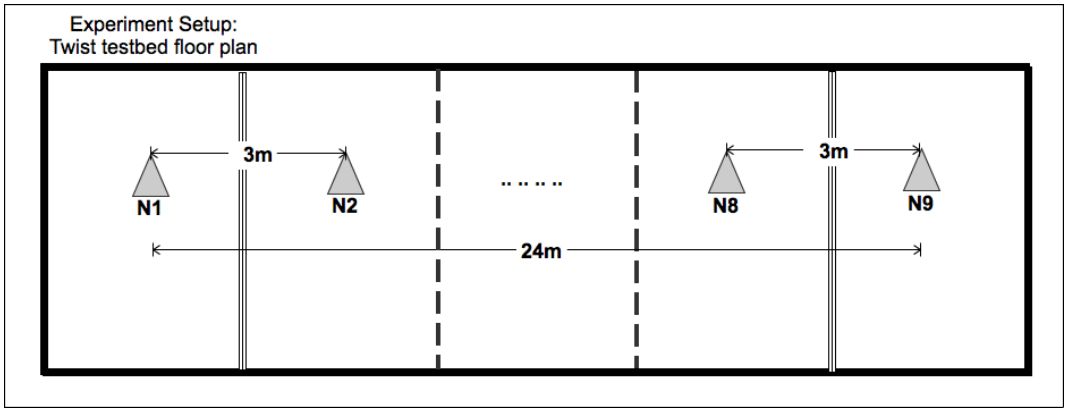


Figure 5.13: Schematic representation of the experimental setup on the third floor

In the experiments, one measurement cycle was so composed, such that all nodes, in turns, transmitted 40 beacon packets with 40 ms intervals on each of the channels $C \in \{11\ to\ 26\}$, while the remaining nodes logged the RSS information from the received packets. All nodes have synchronously changed channels after each period of 40 transmissions, which was approximately every 2 seconds at each channel while receiving from the same sender. A full measurement cycle, involving sending 40 packets from each of 9 nodes in one set, across all 16 channels, takes approximately 4 minutes. In this analysis, 1000 repetitions of a measurement cycle were performed on the set with nine nodes and 2000 repetitions of a measurement cycle were performed on the set with ten nodes.

TWIST testbed provides a backchannel over the USB ports of the nodes for direct communication with the WSN hardware, which we used to synchronize the nodes' channels and collect the measurements. In the untethered experiments outside of the testbed, a control message was

disseminated among the nodes when it was time to change channel. Total time to receive this packet, change channel and relay this control packet is about 10 ms. So an additional time window of $n \times 10\ ms$ was used to allow all the nodes in the network to change their channels synchronously.

Using the obtained RSS information from the measurements, for each node N_i, we created the proximity vector PV_i for estimating the closeness between N_i and the remaining nodes in the sequence, as we defined in the previous part. We analyzed the *average of maximum θ%* method with three different parameters: $\theta = 2$, $\theta = 5$ and $\theta = 10$.

Using the created PVs, a node sequence S was computed for each measurement cycle. This was done for each combination of proximity vector and sequence building methods, described in Sections 5.3.2 and 5.3.3. For example, we applied Greedy sequence building on the proximity vectors generated by Max RSS, Avg RSS and Avg Max θ% (for three values of θ: 2, 5 and 10). We repeated the same parametrization to Greedy Lookback sequence building and MDS as well. Finally, each computed sequence was evaluated as "correct", if the resulting node order matched the ground truth (the sequence reflecting the real positions of the nodes), and it was classified as "false" if the computed sequence did not match the ground truth.

We set the threshold value for the Greedy Lookback algorithm as 2 dBm. The RSS indicator resolution of the radio chip in the hardware that was used in the experiments is 1 dBm, hence 2 dBm is the smallest significant threshold for this case.

The presented bar charts show the percentages of correct sequence discovery from all 1000 repetitions of the measurement cycle. Each bar color represents a different proximity vector metric : maximum RSS (MaxRSS), average of all RSS values (AvgRSS), average of maximum $\theta = 2\ \%$ (AvgMax2%), average of maximum $\theta = 5\ \%$ (AvgMax5%) and average of maximum $\theta = 10\ \%$ (AvgMax10%).

Results on Isolated Channels

To better observe the impact of frequency diversity in the following part, we first present the results from Greedy Sequence Building algorithm on the data taken from the third floor of the testbed with nine nodes. The collected RSS values are grouped by the channels and the algorithm was applied.

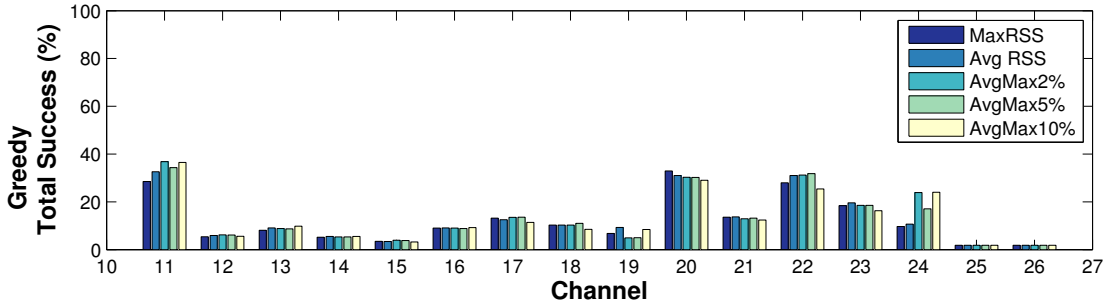


Figure 5.14: Sequence discovery on separate channels, no frequency diversity

The results presented on Figure 5.14, show that using RSS data collected from isolated chan-

nels do not lead to satisfactory sequence discovery performance by using any of the proximity vector forming method. The success rate remained below 40 % for all of the channels. The success rate was also not consistent across different channels.

Success in Sequence Discovery

Next, the sequences were computed using all the measured RSS values at all channels. In a linearly positioned setting of nodes, two different reference nodes can be selected at each end of the sequence separately. After collecting the dataset, we ran our algorithms to build from both ends of the sequence. If one imagines the physical arrangement of the nodes on a horizontal line, we will refer to one direction of sequence construction as the LeftToRight direction and to the other direction as the RightToLeft. Since the measurements are taken in a one-to-many fashion, different directions have different orders of senders and receivers, therefore the dataset that is processed and their results will not be same.

The results of the computations are given as bar plots in Figures 5.15, 5.16, 5.17, 5.18. These results show that the frequency diversity in RSS measurements have a significant impact, with the sequence detection rate immediately raising to over 80 %, while it was below 40 % for isolated channels in the results from the same dataset given above in Section 5.3.4.

When we compare the results of LeftToRight and RightToLeft computations of the sequence, we observe that the links are not symmetrical. Yet, our Greedy Lookback algorithm deals with this asymmetry better than others, while maintaining the highest success ratio.

For comparisons we applied MDS using the same measurements. The RSS values were converted to distances using the Equation 3.3 with the path loss exponent taken as 3, which is frequently used for indoor environments. These distances were then scaled to a dimension of 1. MDS does not benefit from the information of an anchor node, but only takes inter-node distances as a distance metric. For the sake of fairness, we assumed the locations of the reference nodes for MDS and ran the algorithm on the remaining nodes.

The results for MDS show high inconsistency in the success rate. The results vary between almost no successfully detected sequences in the experimental data on the fourth floor (RightToLeft) to almost perfect scores for the third floor (RightToLeft) scenario. This observation hints at high sensitivity of MDS on the imprecise distance estimates from the proximity vectors. In search of a globally optimal configuration, the classical MDS essentially puts equal weight on all range estimates, disregarding the fact that the information is more reliable for closer nodes, than for nodes that are farther away. In contrast, our greedy algorithms rely almost exclusively on this info, only tapping at the information between more distant nodes when the first info is inconclusive (in lookback version).

In terms of the impact of the proximity vector metric, the results show that "maximum RSS" and "average of maximum θ % RSS" performed very closely to each other and better than the "average of all RSS" values. The reasons for the difference can be traced to a larger instability of the average metric. Figure 5.19 summarizes the distribution of the different proximity vector metrics for the sender N_1 on the third floor, across the 1000 repetitions in the evaluation study.

To check if the sequence detection improvements are really due to the frequency diversity

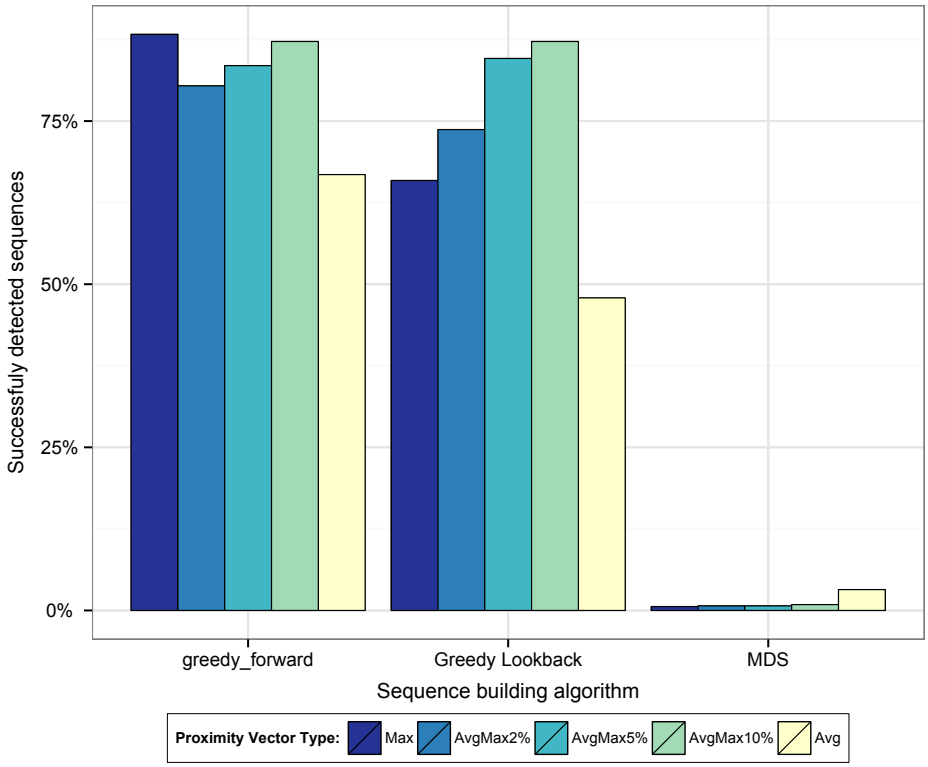


Figure 5.15: Sequence discovery using combined RSS values from all channels; Third floor, 1000 runs, LeftToRight

and not to time diversity or due to difference in the number of RSS samples[2], we have repeated the experiments on the third floor with 16×40 samples at each channel separately. The results are summarized in Figure 5.20 and show that the sequence detection success rate, without frequency diversity, but with higher time diversity, still remains low (this time below 20 %).

When we compare Figures 5.20 and 5.14, we also observe that the channels characteristics are not stable over time. Therefore, we verify that a set of RSS measurements over a single channel is not reliable for node position discovery purposes.

To analyze the stability of the sequence detection success rate, we split the frequency diversity results from the third floor experiments into 10 batches, which are shown in Figure 5.21. The results indicate higher stability of the "average of maximum $\theta\%$" proximity vector metric with respect to the other metrics.

We have further analyzed the batches that had lower success rates. In those cases, we have observed that promoting the most frequently computed sequence as the final decision, even after a few tens of runs would result in detection of the correct sequence with very high probability.

[2]Single-channel proximity vectors operate over 40 RSS samples, while all-channel vectors operate over 16×40 samples, ignoring potential lost packets

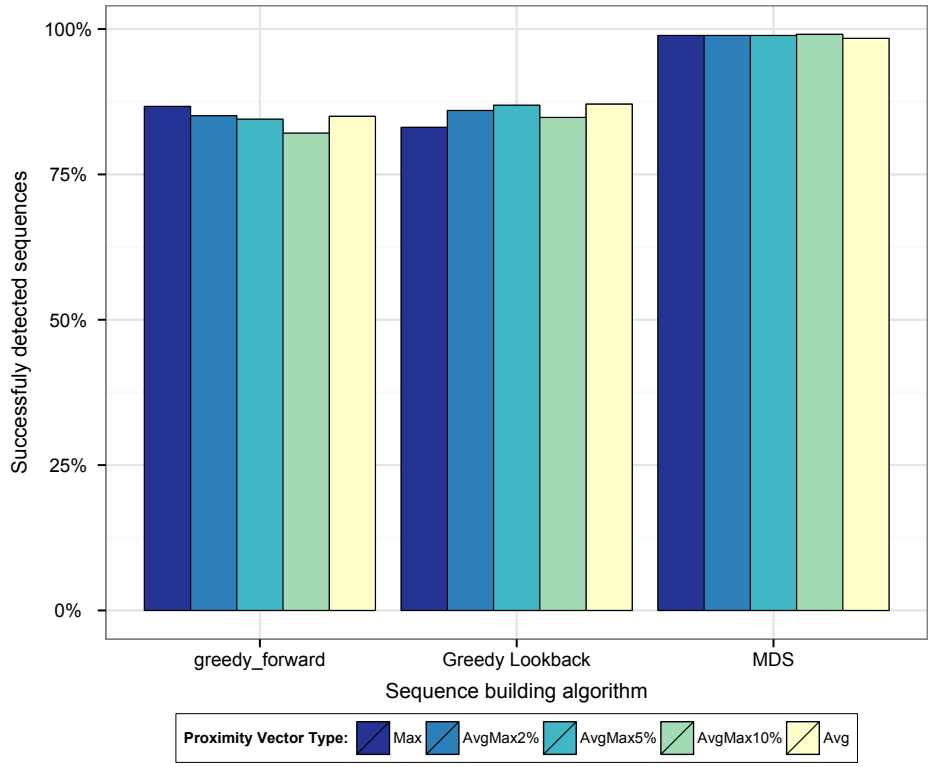


Figure 5.16: Sequence discovery using combined RSS values from all channels; Third floor, 1000 runs, RightToLeft

5.3.5 Summary

In this section we have tackled the problem of node sequence discovery using imprecise, RSS-based, proximity vector estimates. The presented methodologies for proximity vector creation and iterative sequence building algorithms show that even imprecise estimates of the relative closeness between pairs of nodes can result in very precise estimation of the whole sequence of nodes. We analysed two methods for creating sequences, a greedy iterative approach and a greedy-lookback iterative approach. The first algorithm works by selecting the next node in the sequence from the last node's point of view. The greedy-lookback algorithm considers the measurements from priorly appended nodes of the sequence for selecting the next node, if the last node does not have a strong candidate. This greedy-lookback algorithm increases the stability of the results with an added computational complexity.

Using these algorithms, the rate of successful recognition of the node sequence approached to 100 %. Crucial factor for this success lies in the ability to leverage the available propagation distortion diversity over multiple frequencies, allowing more robust relative comparison of the closeness between node pairs despite the effects of large-scale and small-scale fading.

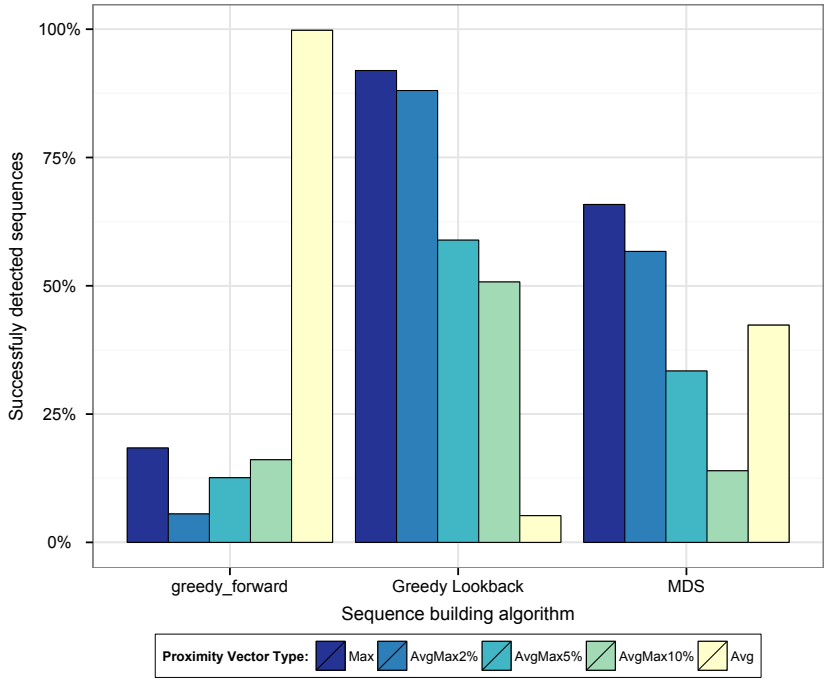


Figure 5.17: Sequence discovery using combined RSS values from all channels; Fourth floor, 2000 runs, LeftToRight

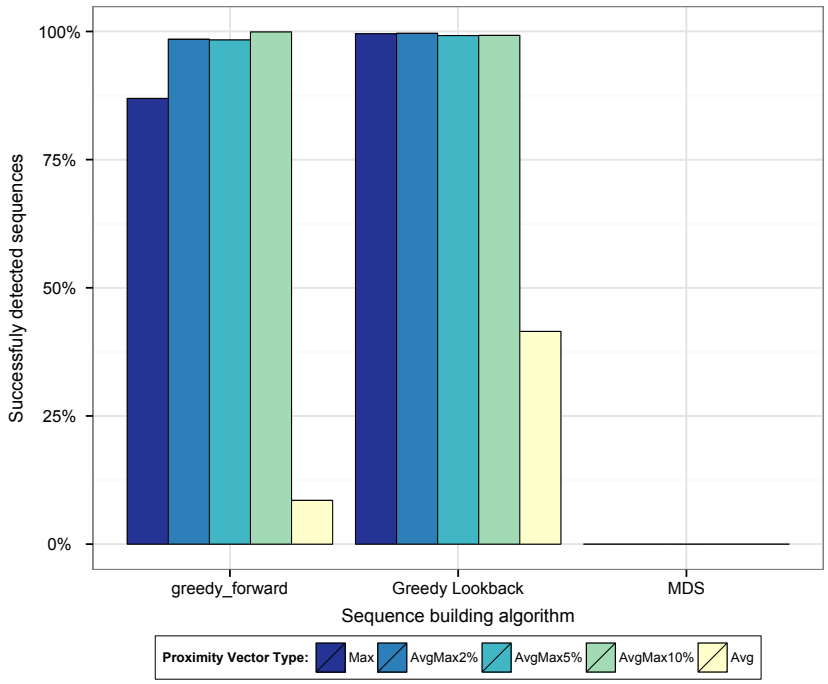


Figure 5.18: Sequence discovery using combined RSS values from all channels; Fourth floor, 2000 runs, RightToLeft

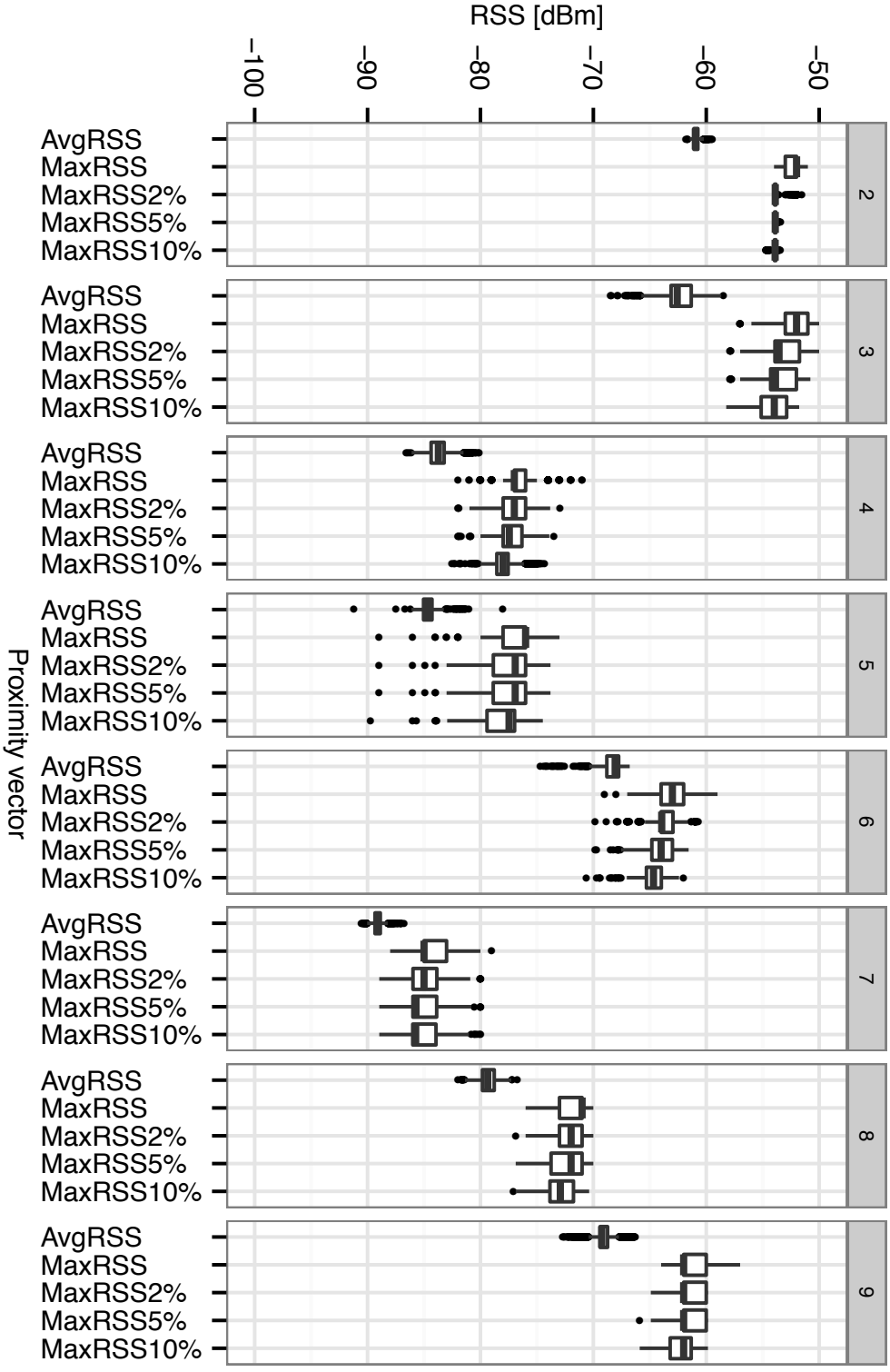


Figure 5.19: Distribution in the proximity vector values for N_1

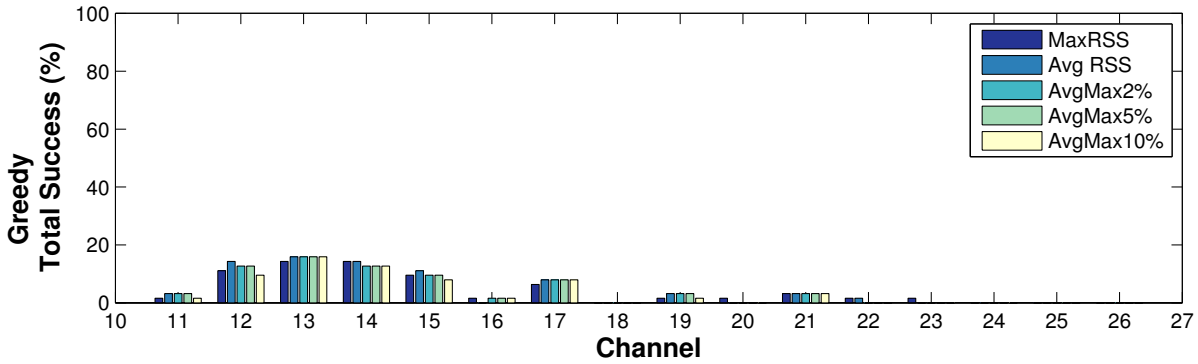


Figure 5.20: Sequence discovery with larger time diversity

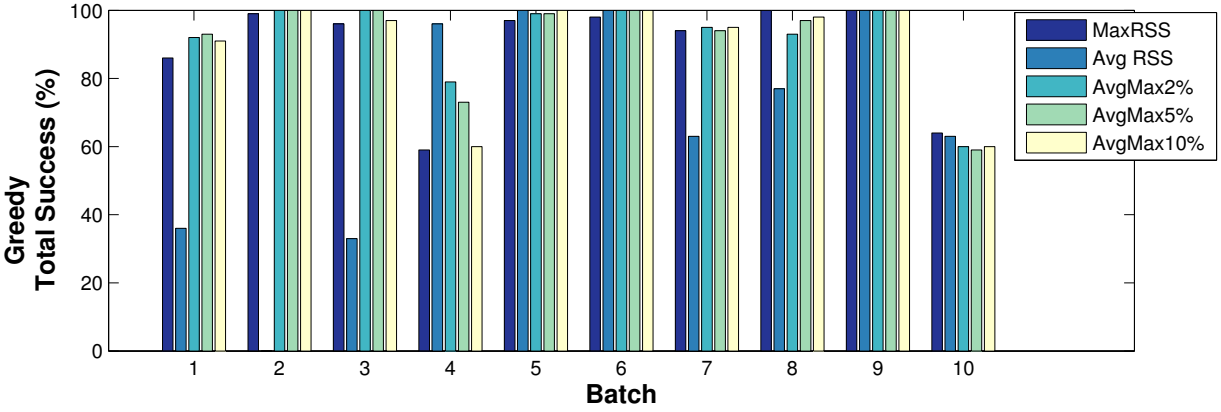


Figure 5.21: Stability of sequence discovery with frequency diversity

5.4 Probabilistic Sequence Building

The previous sections have demonstrated how RSS information, enhanced with frequency diversity, can support the relative node position discovery in a linear deployment of a WSN through sequence building algorithms. Then the sequence was iteratively built by the simple comparison of average of all RSS measurements and the results were not stable despite the utilization of multichannel radios. Better sequence building algorithms developed to leverage only some selected values of the RSS measurements, which are considered to be higher quality. The results were considerably better but there is a potential for improvement, as in some cases the success rate remained below 80 %. The motivation of this part is using more information by processing all (or most) of the measurements for detection of closeness and deducing the best fitting placement of the nodes in the sequence. Simple averaging mixes high-quality and low-quality measurements with each other and we cannot know how to separate them. The binary proximity decisions made with such mixed quality data causes instability, examples of which can be seen by the Figures 5.17 and 5.18. Therefore we will consider a different approach, which assigns a probability to each node for being the closest node to a particular sender node.

This probabilistic approach keeps a memory of the system, so it is best implemented centralized. The measurements are used to create a probability tree of the nodes, root of which contains the reference node and each level corresponds to one distinct position in the sequence.

This proposed system is called Probabilistic Node Sequence Discovery (PNSD), a solution for the indoor positioning problem for the cases in which precise room-level accuracy is desired. By utilizing the diversity in frequency and time, in the same way that was defined in the previous section, it will be shown that the probabilistic approach to this problem enables a more precise classification of the spatial node sequences. This probabilistic methodology, and the results of the analyses are published in our peer-reviewed paper [127].

We again focus on discovering positions of the nodes, relatively to each other, on a single-dimensional setting (i.e. in rooms along a corridor). Same assumptions that are taken in the previous parts of this chapter also pertain in this part. A set of n rooms, R_1 to R_n, is considered being the potential positions for the nodes to be mapped. Any adjacent pair of nodes might be separated by a wall of the room or alike. Each room R_i is considered to contain one of n stationary nodes, whose presence is to be discovered and denoted as N_i. First room (position) in the setting contains the $Reference\ node(N_1)$ and positions of the remaining nodes are not known. We assume that all the positions are occupied or unoccupied positions do not need to be discovered.

The system first collects RSS data in the same way that was described in Section 5.3.1. Here, too, we desire diversity in RSS measurements for benefiting from diversity in signal propagation. At any time during the measurement process the transmitting node is referred as the Sender while all other nodes are referred as Receivers. The sender node transmits z packets in e channels. All other nodes which successfully receive these transmissions keep a record of the sender, transmission channel, transmission number and measured RSS.

After all nodes have assumed the sender roles and the measurements have been completed, the collected data are sent to a processing center. All this data are later used for creating a probability tree of the node positions, root of which contains the reference node.

5.4.1 Building the Probability Tree

The tree is initiated by placing the reference node (N_1) at the root position. The children of this node in the tree will contain the receivers of N_1, with probability weights. The weight is the empirical probability of being the closest node. We define this probability as the ratio of the number of times that the highest RSS value was measured in all transmission to the total number of transmissions across all channels, for all the receivers of a sender. This way of calculating the probability is based on the second sub-hypothesis in the Section 4.1.1. For example, from 100 transmissions by N_1, if node N_2 measures the highest RSS value 60 times and N_3 measures 40 times, then the weight of the edge between node N_1 and N_2 is 0.6 ($P(N_2|N_1) = 0.6$), whereas the weight of the edge between node N_1 and N_3 would be 0.4 ($P(N_3|N_1) = 0.4$). Other receivers will not be placed if they do not measure the highest RSS value for any transmission, therefore $P(N_x|N_1) = 0,\ x \notin \{2, 3\}$. For each added vertex (node) to the tree, this procedure is repeated and the tree is accordingly extended until the depth of the tree is equal to the number of all positions in the system. The procedure is shown in Algorithm 8.

Algorithm 8 Probability Tree Building Algorithm

Step 0: Initialize empty tree T

Step 1: For each node N_i in the sequence

Step 1.1: Measure RSS from multiple frequencies
(triggered on demand or during normal communication)

Step 2: Collect measurement reports from each node

Step 3: Place reference node at the root (T_1) of the tree

Step 4: For each leaf (T_x) of the growing tree:

Step 4.1: Add all receivers of T_x
except if they are already in the path $T_1 \rightarrow T_x$

Step 4.2: Assign probability weights to new added edges

Step 5: Repeat *Steps 4, 4.1, 4.2* until depth of T reaches
the total number of nodes

Step 6: Find the path ($*\vec{S}$) with greatest joint
probability from T_1 to any of the u leaves T_x in T

Here, each vertex of the tree represents a node and each level of the tree represents a potential position in the sequence. If an edge exists between two vertices V_a (containing node N_a) and V_b (containing node N_b), and $level(V_a) < level(V_b)$, then the "weight of $Edge(V_a, V_b)$" is the assumed probability of "node N_b being at the next position after node N_a", $P(N_b|N_a)$, at that level. So the joint probability of a path S_i is $P(S_i)$, which is the product of all the weights of edges form T_1 to the last vertex (node) in S_i. Our verdict is the most probable path (sequence) in S, which is the path S_i that has the maximum joint probability.

$i \in 1 \rightarrow u$ // Total number of paths in the finalized tree

$*\vec{S} = \{S_i, S_j, S_k, ..., S_u\};$

$S = argmax(P(*\vec{S}))$

$$=> S = argmax(\prod_{y=T_0}^{T_z} P(V_y|V_{y+1}))$$

A probability tree is visualized in Figure 5.22. For readability purposes we have considered that maximum two nodes will have a probability greater than 0 to any parent node in the tree.

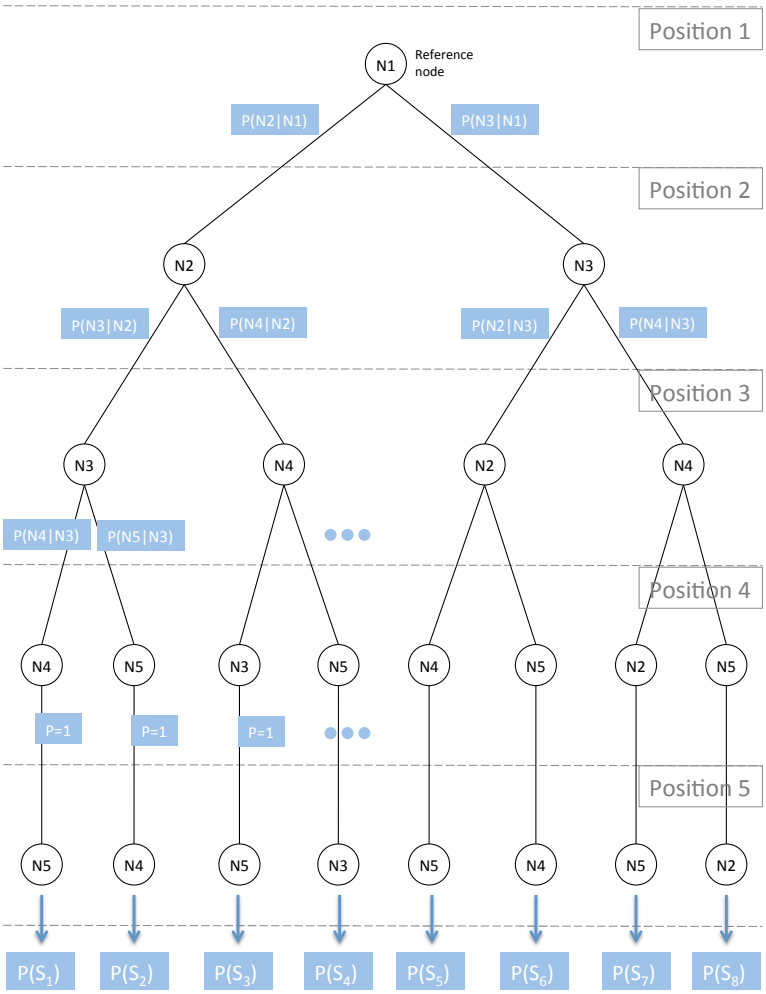


Figure 5.22: An illustration of a finalized probability tree.
Selected sequence is $S = argmax(P(*\vec{S}))$

5.4.2 Evaluation on Testbed

We tested the presented probabilistic sequence determination method on the measurements that were obtained from the TWIST testbed [29]. We used three sets of nodes: Set-A, Set-B and Set-C. In each of these node sets in the testbed, the nodes are uniformly positioned 3 meters apart along a line and there is one wall partition in between each adjacent node pair.

Set-A contained ten nodes from the fourth floor of TWIST building on the side faced to the other buildings of the Technische Universität Berlin. Set-B was the set of six nodes on the fourth floor of the TWIST building, on the side which faced an open area. Set-C contained ten nodes in the same rooms as Set-A, but slightly closer to the doors of the rooms than the windows. The nodes in the Set-A, located in the south side of the building, are the same nodes that were previously used in the experiments in Section 5.3.4. The placement of nodes in Set-A is illustrated in Figure 5.23.

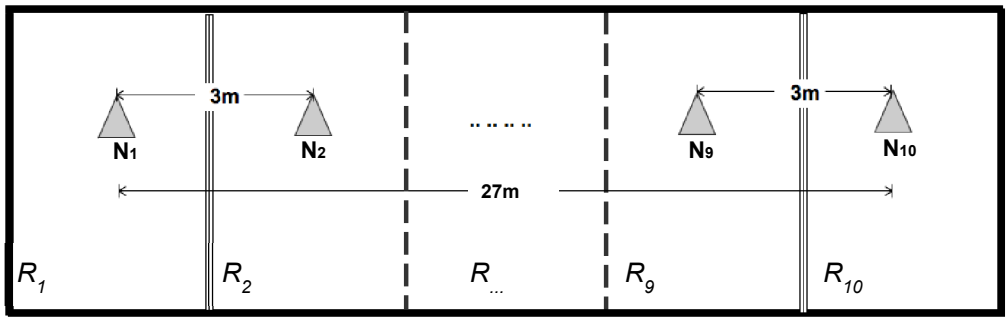


Figure 5.23: Schematic representation of the experimental setup, Set-A

We took the measurements with the procedure we defined in Section 5.3.1. A measurement cycle for one sender took about 2 seconds. The full measurement cycle, involving sending 40 packets from all of 10 nodes in the sequence, across all 16 channels, takes approximately 4.3 minutes. We have initially performed 2000 repetitions of the measurement cycle with the nodes in Set-A over the span of roughly 6.5 days.

All of the measured RSS measurements were collected using the wired backchannel that is provided by TWIST testbed. This backchannel was also used to synchronize the channel switching during the experiments.

When applying our algorithms, we assumed N_1 as the reference node, whose position was known to be the first of the sequence on the (virtual) left end. We also repeated our computations by taking the node at the right end of each node set as the reference node. Note that, when we are looking for the closeness relationship among nodes, we consider the transmissions of one-to-many, in other words from one sender to multiple receivers. Hence, reversing the computation direction means that the order of senders and receivers are different. So each different direction represents a different dataset and the results will not be related. We refer these different directions of the same sequence of nodes as “LeftToRight” and “RightToLeft”, like in the previous section. All these RSS readings are used for generating a weighted probability tree.

For improving the computational performance, we have pruned the weakest paths from the tree during the runtime of the algorithm, which we elaborate in Section 5.4.5.

For evaluating, we have compared our algorithm to RSS-based ranging and Multidimensional Scaling (MDS). We used the Smacof implementation of MDS [68], since it gave better results than the standard implementation of MDS in R-project environment [128]. As input to MDS, we converted the RSS values to distance using path loss attenuation equation with path-loss coefficient $\alpha = 3$.

For RSS ranging, we used the greedy approach: starting from the reference node, we added the next closest node to the sequence and iterated this for each last added node until all nodes are placed in a "greedy" manner. As input, we took the average of the values collected in all channels (Greedy Avg) and on single channels 11 , 18 and 26 (Greedy AvgCh$\{11,18,26\}$) separately. This method was explained previously in Section 5.3.3 in more detail.

We have initially tested our algorithm on the Set-A with 2000 measurements. We repeated the computations for both directions, namely RightToLeft and LeftToRight. In one direction our methodology (PNSD) has achieved 100 % success, while in the other direction it has achieved around 93.6 % success, which was influenced by external parameters to a small extend, in our opinion. In this scope, "success" is again defined as **perfectly recognizing the correct sequence**. In Figure 5.24 the performance of PNSD in comparison to alternative methods is visible.

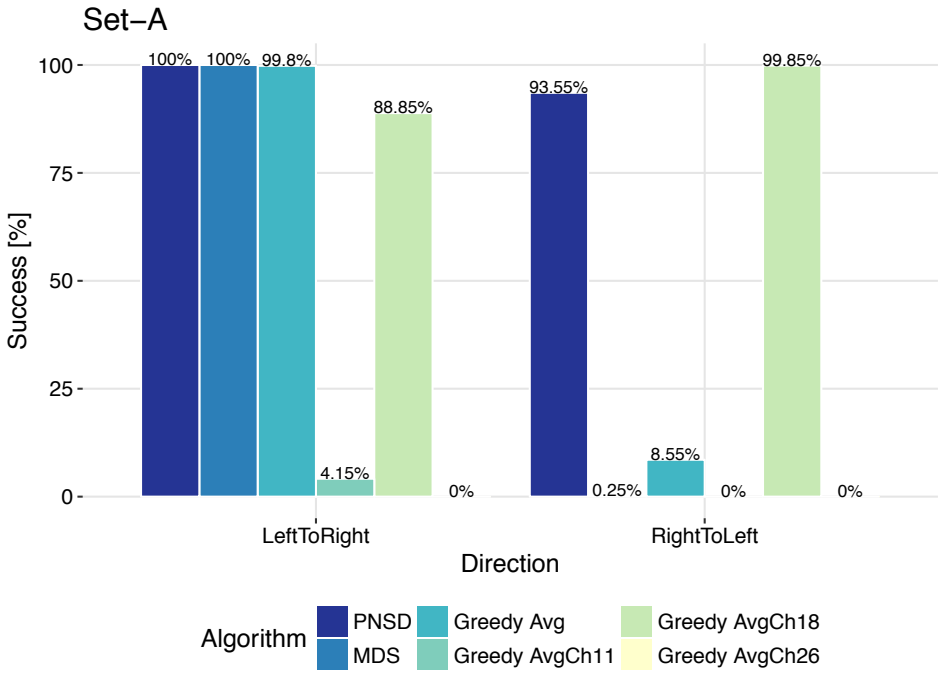


Figure 5.24: Success on Set-A

Although being a very good position estimator for range-based localization, MDS could not perform good enough for discovering sequences with RSS information. In one direction it pro-

vided 100 % success but in the reversed direction it almost never succeeded (0.25 %) to generate the correct node sequence. This shows that MDS is too sensitive to the asymmetry of the wireless channel. Although it has been used widely [95], averaging RSS measurements on a single channel does not give us satisfactory results. On channel 11 (Greedy AvgCh11), 4.15 % and 0 % of the time (for each direction) correct sequence were detected using the Greedy approach and on channel 26 (Greedy AvgCh26) the success has been 0 % in either of directions. On the other hand, the correct sequence was detected on channel 18 (Greedy AvgCh18) at 88.85 % rate in one direction and 99.85 % in the other direction. This shows that the single-channel approaches can sporadically lead to good results, but they have limited practical value because the optimal channel changes with time. The Greedy approach with channel diversity (Greedy Avg) has been quite successful (99.8 %) in the direction LeftToRight while it has shown very little success (8.55 %) in the direction RightToLeft, which shows us that even channel diversity is not enough by itself for highly successful sequence discovery.

The building we used in our experiments has a rectangular structure and it has offices on its north and south sides. The measurements given above were taken from the south side, which faces the rest of the campus area. We also took measurements from the north side, which faces an open space, with a set of 6 nodes, Set-B. Due to the configuration of the environment, we expect reduced interference effects on this side of the building, as opposed to the campus facing side. Since the building has very similar features for both node sets, the measurements from Set-B hint us that in the absence of the severity of interference that we are facing in our facilities, the proposed method works flawlessly. We performed 692 (time-limited) measurement cycles and had 100 % success with PNSD, which is shown in Figure 5.25.

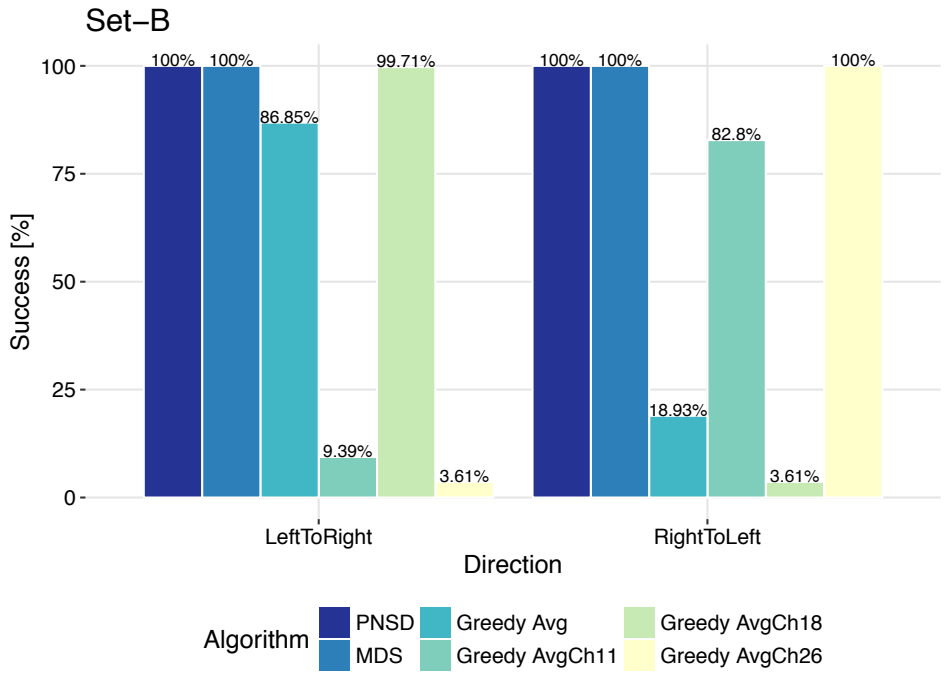


Figure 5.25: Success on north side, Set-B

The side that hosted the nodes of the Set-B has minimal interference in comparison to the other side of the facility. This is also improving the performances of other methodologies, such as MDS had 100 % success in both directions. Greedy Average was about 87 % successful in LeftToRight direction but only 18.93 % successful in the other direction. Greedy sequence building on single channels again vary drastically, for example on Channel 18 correct detection was 99.71 % and on Channel 26 success ratio was as little as 3.61 %.

All these experiments suggest that being in a high-interference medium, in comparison with the other evaluated methods, PNSD shows better stability. We are analyzing this further in Section 5.4.3.

As stated before, the misclassifications were not random, but rather localized into a small part of the measurement environment. That is, in vast majority of the cases with Set-A, just two of ten nodes were imprecisely estimated to be in swapped rooms with each other. We repeated the experiments with the same settings but using a different set of nodes (Set-C) at slightly different positions in the campus-side rooms. The 10 nodes in the experiments again had a linear configuration. The major difference was that this time we performed the experiments during the week, which meant a much bigger possibility of higher density of wireless transceivers around the testbed. The ratio of finding the correct sequence was lower as expected. The results are plotted in Figure 5.26. Although over all success was decreased, our proposed methodology was much better, compared to alternative methods, in terms of both stability and success rate. It is worth noting that success rate of Greedy Average at channel 11 results were better than others in one direction. However, the results from the other direction show that getting good results without frequency diversity is solely coincidental, just like we have seen also in Figure 5.24.

By looking at the results we could notice that there was an anomaly in some place or time of the testbed-based experiments. Despite of a lot of effort, we have not been able to pinpoint the reason for those anomalies. We tested and excluded the option of “faulty hardware”. We intentionally created and monitored Wi-Fi interference, without being able to recreate same type of anomaly. Nevertheless, we have introduced an additional feature to our algorithms for “classification of result reliability”, which is explained in Section 5.4.4, in an effort to detect the likelihood of existence of an anomaly at the time of the measurements.

Analysis of Imperfect Results in Set-A

All the results of PNSD, which did not provide the perfectly correct sequence, had single position error, which means two nodes are swapped in their estimated positions. The analysis of the results shows that these errors were localized at all cases, namely sixth and seventh nodes in the sequence falsely appeared in each other’s places. We then examined the exact timestamps at which the RSS measurements were taken in the non-perfect experiments. The measurements were taken around christmas time, which is an official holiday in where TWIST is located. We can see that the measurements which were taken outside of the official holiday time caused a localized swap, as they were more likely to be exposed to interference. The other part of the measurements, which fall within the christmas break time, could be used to determine the sequence of the nodes with PNSD with perfect accuracy. We have plotted the indices of failed experiments in Figure 5.27.

One sample of the raw measurements where PNSD had a successful discovery is shown in

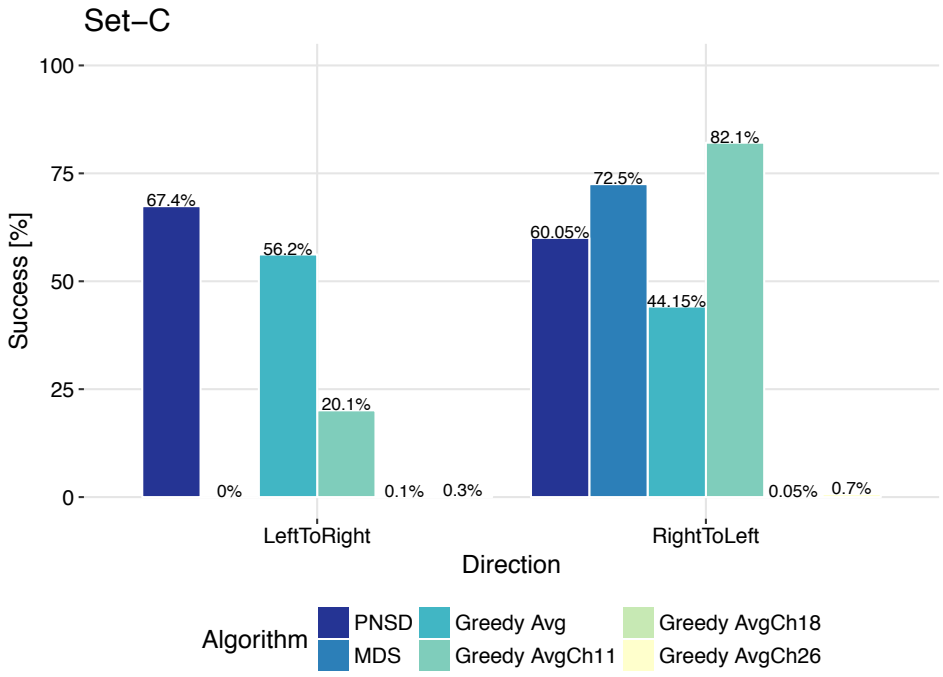


Figure 5.26: Success on second set of 10 nodes, Set-C

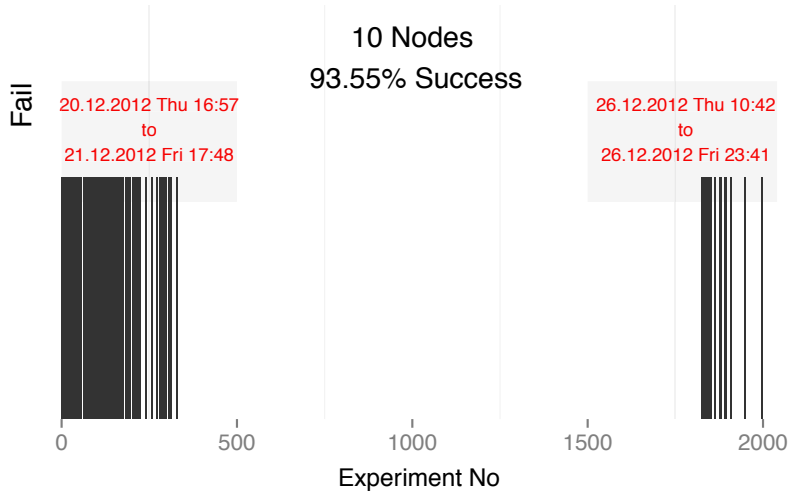


Figure 5.27: Failed Experiments: each bar corresponds to the index of a failed experiment with PNSD

Appendix B in Figures 9.21 to 9.24. It is evident in those measurements that the selection of the closest receiver to a transmitting node is not trivial. Another sample of the raw measurements,

with which PNSD could not compute the correct sequence perfectly, is given in Figures 9.25 to 9.28, which also indicates the non-triviality of closest node detection.

5.4.3 Evaluation with Simulations

The results we present in the previous section were taken only in one specific building and one specific testbed. Although we repeated the tests and measurements in different areas of same testbed, these limited number of measurements will never be enough to guarantee that the current results are not testbed or location specific. It is also not practical and time-wise feasible to repeat the experiments in many different buildings.

To investigate the testbed dependency of our results, and different channel conditions, we used a simulation model that represented an indoor environment. The RSS is assumed to be affected by various propagation effects. Large scale variation of RSS is assumed to be caused by change in distance. We do not consider shadow fading in this model. Small scale variation of RSS is due to the fact that the wireless signals arrive at the receiver from multiple paths and therefore suffer from multipath fading. As each channel bandwidth is smaller than coherence bandwidth of usual office environments with around 100 ns delay spread, we assume a frequency flat fading for each channel but the fading gain is expected to change from one channel to the other channel which provides the frequency diversity of the system. We assume that the environment is static and changes slowly in time which means that the coherence time is very large. RSS is also affected by the white noise around the receiver.

In the design of simulations, the propagation effects were modeled using statistical channel with path loss models [129, 130]. The signal strength values were produced based on the above assumptions. The transmission power was constant and the path loss exponent α was assumed to be static in the environment. We used Rayleigh fading model to model small scale fading where the channel fading gains came from a Rayleigh distributed random variable. The fading gains were assumed to be constant on a channel due to slowly fading assumption however it was assumed that the fading gain was different on each channel. The relation between received power P_R and transmit power P_T, both in $milliwatts$, is presented in the following equation:

$$P_R = |h|^2 \times P_T \times \left(\frac{\lambda}{4 \times \pi \times d}\right)^{\alpha} + P_n \tag{5.3}$$

where h is the channel gain, which is a Rayleigh distributed random variable. λ is the wavelength at 2.4 GHz, α is the path loss exponent and d is the distance between transmitter and receiver antennas in meters. P_n is the power of additive white Gaussian noise in the environment. We have chosen the noise variance as -45 dBm and the fading gain variance h as 10. Both random variables have zero mean. Path loss exponent α was taken as 3, which is typical for indoor office environments.

In Figure 5.28, we show a set of generated RSS values using above formula and parameters. Each produced RSS value corresponds to the reading at different transmitter-receiver distance. This graph shows that the signal strength value is a function of the distance, however it is not always monotonically decreasing with distance due to multipath fading.

Later, we produced 16×40 values at 9 different receiver distances of 3 m apart and compared them to one of the measurement sets from the testbed containing the same amount of RSS data.

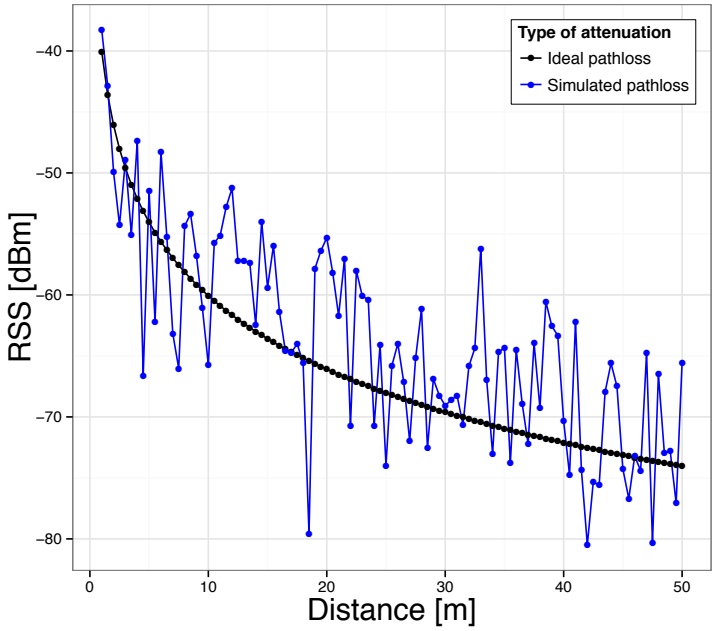
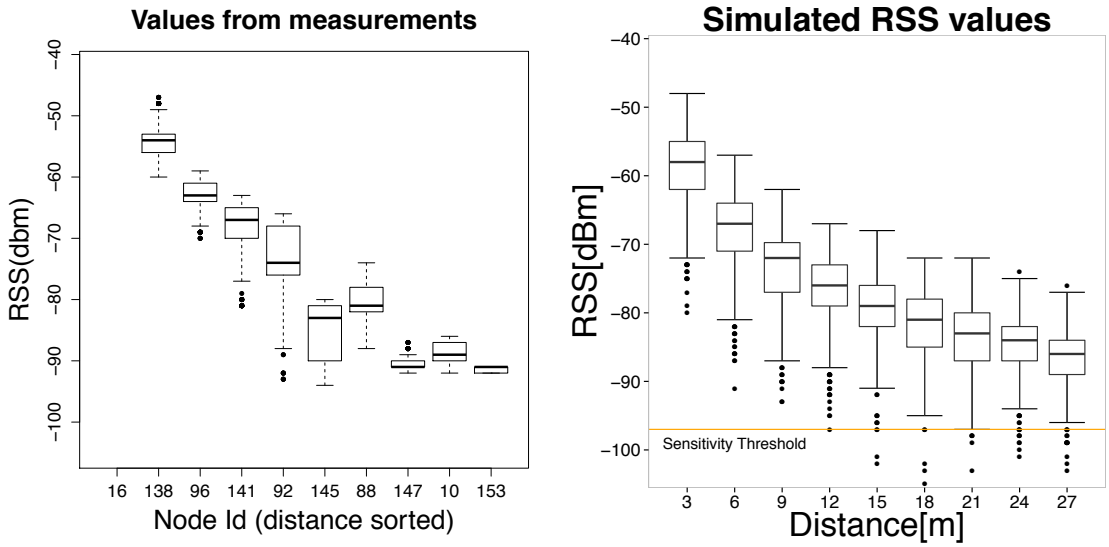


Figure 5.28: Simulated and ideal RSS values as a function of distance

The data are plotted side by side in Figure 5.29, showing that the values we produced with the simulation model above were not far from what we would expect from our real measurement environment. From the produced values, we dropped the values that were below the sensitivity level of the radio chips, which is -97 dBm for CC2420.

Finally, we produced the same number of simulated results as we had in our measurements: 2000 repetitions of ten-node setups with nodes 3 m apart from each other. On each generated set of data, we applied the methodology that we explained in Algorithm 8. The result were 100 % perfect identification of the correct sequence for either directions.

Figure 5.30 shows us, that in a propagation medium that is fairly affected by fading and multipath, our methodology remains robust, while MDS failed about 1 % of the time. We must notice that LeftToRight and RightToLeft computations are pretty much symmetrical, since time has not been a parameter in the RSS generation. For each experiment, we used same standard deviation and mean values for Rayleigh distributed channel gain and Gaussian distributed white noise. According to the Law of Large Numbers, the average of the random variables are expected to approach to their mean values, which was the case in our simulations, too. When we average a big number of values, produced with Formula 5.3, we come closer to what Free-space path loss equation would give us under ideal conditions. Hence, 100 % success with averaging all values from all channels is expected in simulations, although it is often not the case in real-world scenarios.



(a) (b)

Figure 5.29: Real measurements versus simulated RSS values

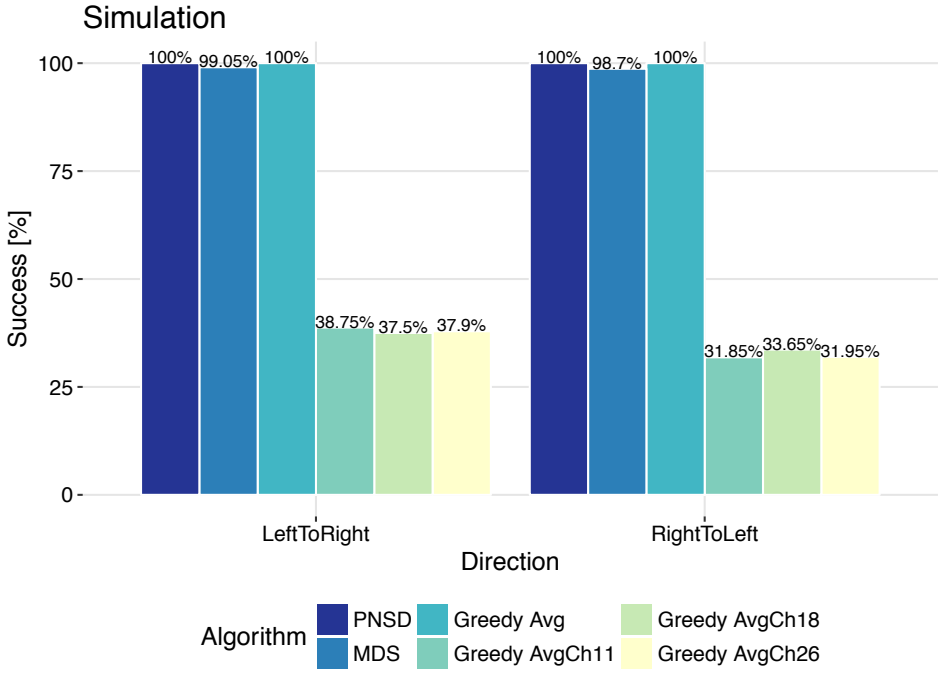


Figure 5.30: Success with simulated measurements

5.4.4 Reliability Analysis through Reverse Validation

With presented PNSD methodology we have achieved a significant improvement over alternative methods in identifying node sequence correctly with relatively small inter-node distances. We have shown that we can achieve up to 100 % correct identification of the node sequence where other approaches seemed to be statistically unstable. We have also noticed that occasionally, due to some anomalies, two adjacent nodes can be incorrectly identified in swapped spots. In this work we assume that an outcome of relative position discovery that is less than 100 % correct identification of node positions is not desired, and we want to be sure that the verdict is dependable. With this aim, we have developed a post-processing methodology to assess the reliability of the computed sequence and change the verdict if a better one exists. Hence, we will evaluate the reliability through reverse validation of the best sequence S_1 that has the highest probability and the second best sequence S_2 with second highest probability. Then we will choose the sequence that is in agreement with the reverse sequence created from the same measurement.

We have argued that the channel does not stay symmetrical even though there is very little time between measurements from node A to B and from B to A. In this case, assuming a four-node scenario (A-B-C-D), node A wants to decide between adjacent nodes B and C, and it falsely decides that C is the closer one instead of node B. In this case, the odds are very little for D to do the exact same mistake by choosing B as the closest node. Such a case would require same type of measurement distortion between different nodes and different places. But the correct verdict of the sequence in one direction is much more likely to be in agreement with the sequence constructed from the reverse direction. A similar idea, which was presented in 2011 in [109] was not known to us at the developing time of this work, but it confirms our reasoning. With this justification in mind, we construct another sequence of the nodes for validating a result, taking the node in the last position of the constructed forward sequence as the reference node of the backward sequence to be constructed.

PNSD produces a set of sequences $^*S = \{S_1,\ S_2,\ S_3,\ ..., S_u\}$, where

$$P(S_1) > P(S_2) > P(S_3) > ... > P(S_u)$$

For reliability analysis, we are interested in the two best sequences in the reverse direction that are produced by taking the node placed in the last position of S_1 as the reference node, which we denote with $S_1^{'}$ and $S_1^{''}$, where $P(S_1^{'}) > P(S_1^{''})$. We are also interested in such reverse sequences, which are produced by taking the node placed in the last position of S_2, which are $S_2^{'}$ and $S_2^{''}$, where $P(S_2^{'}) > P(S_2^{''})$.

Our probabilistic method has also shown us that imperfect results are not just random. When we sort the sequences by their probabilities ($^*S$), the second best sequence (S_2) has been in most cases the correct one if the sequence (S_1) with the highest probability was incorrect.

For reliability assessment, we are looking for finding an agreement, by comparing the most probable sequence (S_1) with its most probable reverse sequence ($S_1^{'}$) and second most probable reverse sequence ($S_1^{''}$). Then we compare second most probable sequence (S_2) with its most probable reverse sequence ($S_2^{'}$) of S_2 and second most probable reverse sequence ($S_2^{''}$) of S_2.

Our claim is, the sequence (S_i), which matches one of its reverse sequences ($S_i^{'}$ or $S_i^{''}$) is more likely to be the correct sequence (hence the verdict will be S_i) and by looking at the rank

(i) of that sequence (among other competing sequences of the same dataset) we will decide how reliable the final decision is. Here we are taking a pragmatic approach of limiting the number of candidate sequences to two in the reliability check. In the great number of observations that we had with the correct and incorrect results, the correct result could either be found in one of these two generated sequences or the correct sequence could not be found in the results at all. In the case the correct result does not appear anywhere within the generated sequences, the chances of a sequence to match with its reverse sequence is also very low.

We have 4 comparisons:

$$
\begin{aligned}
S_1 = S_1^{'} &\Rightarrow match\_rank = 1 \\
S_1 = S_1^{''} &\Rightarrow match\_rank = 1 \\
S_2 = S_2^{'} &\Rightarrow match\_rank = 2 \\
S_2 = S_2^{''} &\Rightarrow match\_rank = 2
\end{aligned}
$$

The number of comparisons (s) where we have a match:

$$
\begin{aligned}
t(S_i) &= \begin{cases} 1, \textbf{if } S_i \in \{S_i^{'}, S_i^{''}\} \\ 0, \text{otherwise} \end{cases} \\
s &= t(S_1) + t(S_2)
\end{aligned}
$$

As the final verdict, we choose between S_1 and S_2, depending on which one has a match within its reverse sequences. If both of them have a matching sequence, then we choose the one that is associated with the highest probability.

$$
finalVerdict = \begin{cases} S_i, \textbf{if } S_i \in \{S_i^{'}, S_i^{''}\} \textbf{ and } s = 1 \\ S_j = arg\ max(P(S_1), P(S_1^{'}), P(S_1^{''}), P(S_2), P(S_2^{'}), P(S_2^{''})) \textbf{ if } s \neq 1 \end{cases}
$$

We define 3 levels of reliability: High, Medium and Low. For the first two levels of reliability, we are looking for matching sequences and a significant difference in the computed probabilities.

The reliability levels are:

High:
if $s = 1$ and $match\_rank = 1$

Medium:
if $s > 1$ or $match\_rank = 2$

Low:
if $s = 0$
or
if $S_1^{'}$ or $S_2^{'}$ cannot be produced in the same length as S_1 using the same dataset. This happens in case of too many packet losses.
or

if $\frac{|P(S_1)-P(S_2)|}{min(P(S_1),P(S_2))} < 0.1$ and $s \neq 1$

If an experiment has one or two matching sequences, but none of these matching sequences have a probability difference of at least 0.1 from the competing sequence (i.e $P(S_1)$ vs $P(S_2)$), then we put this experiment into the "Low" reliability class. This value of probability difference threshold is a local parameter that is observed in many measurements that we performed and analyzed. For different sizes of networks it must be properly chosen based on a few observations. The analysis on how to mathematically calculate such a threshold is left to future work.

We applied the described reliability verification methodology to our experiments that we presented in Section 5.4.2. In the experiment, in which we had 100 % success (Figure 5.24), 9 experiments resulted in wrong verdicts in oppose to 1991 correct verdicts by PNSD. Those wrong verdicts, however, were all classified as "Medium" reliable. The other direction for the nodes in the test, where we had 93.55 % success, the success ratio was increased to 96.95 % success, changing 68 wrong verdicts with the correct ones. In both cases, 100 % of the sequences that were classified as "High" reliable, were the correct verdicts. Results are plotted in Figure 5.31.

In the second set of experiments with the nodes in Set-C from the south part of the building, in which we had lower success ratio (Figure 5.26), the verified success rate increased in both directions. More importantly, we could associate most of the wrong verdicts with "Low" reliable verdicts. Again 100 % of the "High" classified verdicts were correct. The results are shown in Figure 5.32. In this set, the success ratio was increased from 60.05 % to 86.7 % in RightToLeft direction and from 67.4 % to 89.5 % in the other direction. This was due to the nature of our "reverse validation" algorithm which chooses the more consistent result as the final verdict. Therefore, by applying reverse validation, we can go beyond assessing the reliability of a verdict, and we can replace the verdict with a better one, increasing the chances of success.

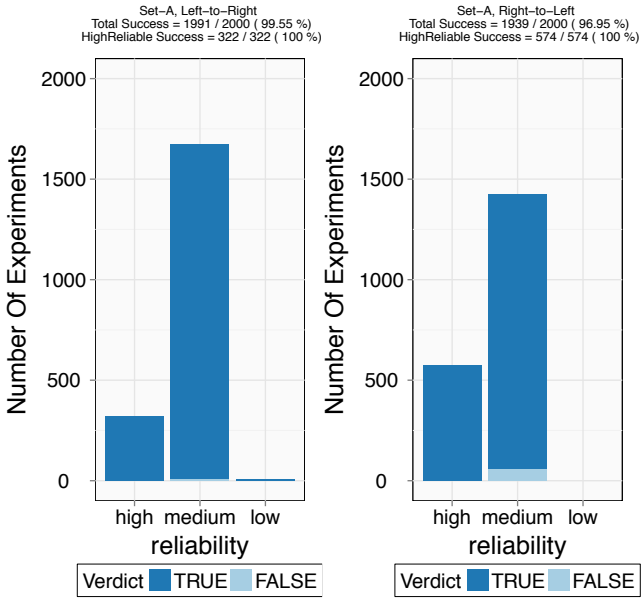


Figure 5.31: PNSD with verdict reliability classification, Set-A

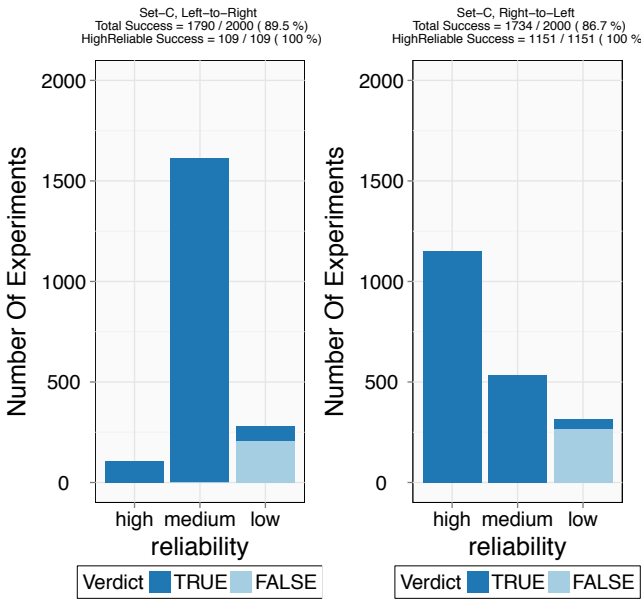


Figure 5.32: Sequence Discovery with verdict reliability classification, Set-C

5.4.5 Complexity and Overhead Reduction

Computation with pruning

Constructing a probability tree for achieving the correct sequence is possible as shown in Section 5.4.1, but unfortunately this tree construction requires high computational complexity. Theoretically, in the worst case scenario, the tree would result in $(N-1)!$ leaves for N nodes and the computational complexity would be $O(N!)$. This is not feasible if the number of nodes in a network exceeds a few. For making this probabilistic position discovery system feasible, the size of the tree needs to be reduced. One way of achieving this is by pruning the probability tree, when the weight (joint probability) of a path (root-to-leaf) becomes too small as that sequence grows into a less likely sequence. We prune the tree at every m^{th} level (in other words, after each m^{th} position in a target sequence). In our experiments, we have also observed that the competition for measuring the maximum RSS value occurs only among few closest nodes and much farther nodes do not measure an RSS value that is higher than what those closest few nodes measure, due to distance related attenuation, despite multipath and RF noise.

We define the pruning criterion at m^{th} level as "half of the maximum root-to-leaf joint probability(P^{max})". Any root-to-leaf path that has a smaller probability than $0.5 \times P^{max}$ is pruned out at each m^{th} level and the iteration continues with the remaining leaves.

We applied this pruning method on the same data that we used to create the results in Figure 5.24. As a result, the computation time was drastically improved (from 15 minutes to 3-4 minutes for one experiment on an Intel i7 architecture) while maintaining the success ratio the same. Figure 5.33 shows that the success ratio did not decrease when we pruned the probability tree with 10 nodes at levels $m = 5$ and $m = 1$. The lesser values of m causes more frequent pruning and hence the speed is also increased. One needs to keep in mind that pruning affects the number of total paths in the final tree (number of candidate sequences).

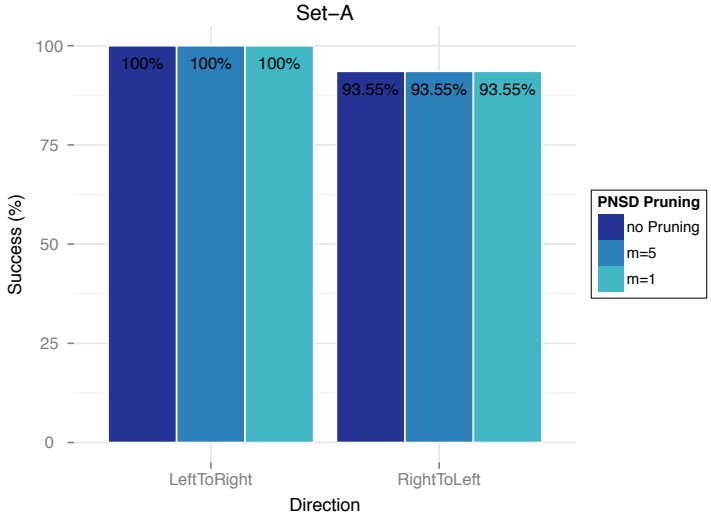


Figure 5.33: Success with pruning, Set-A

An illustration of the tree pruning, with $m = 1$ is shown in Figure 5.34.

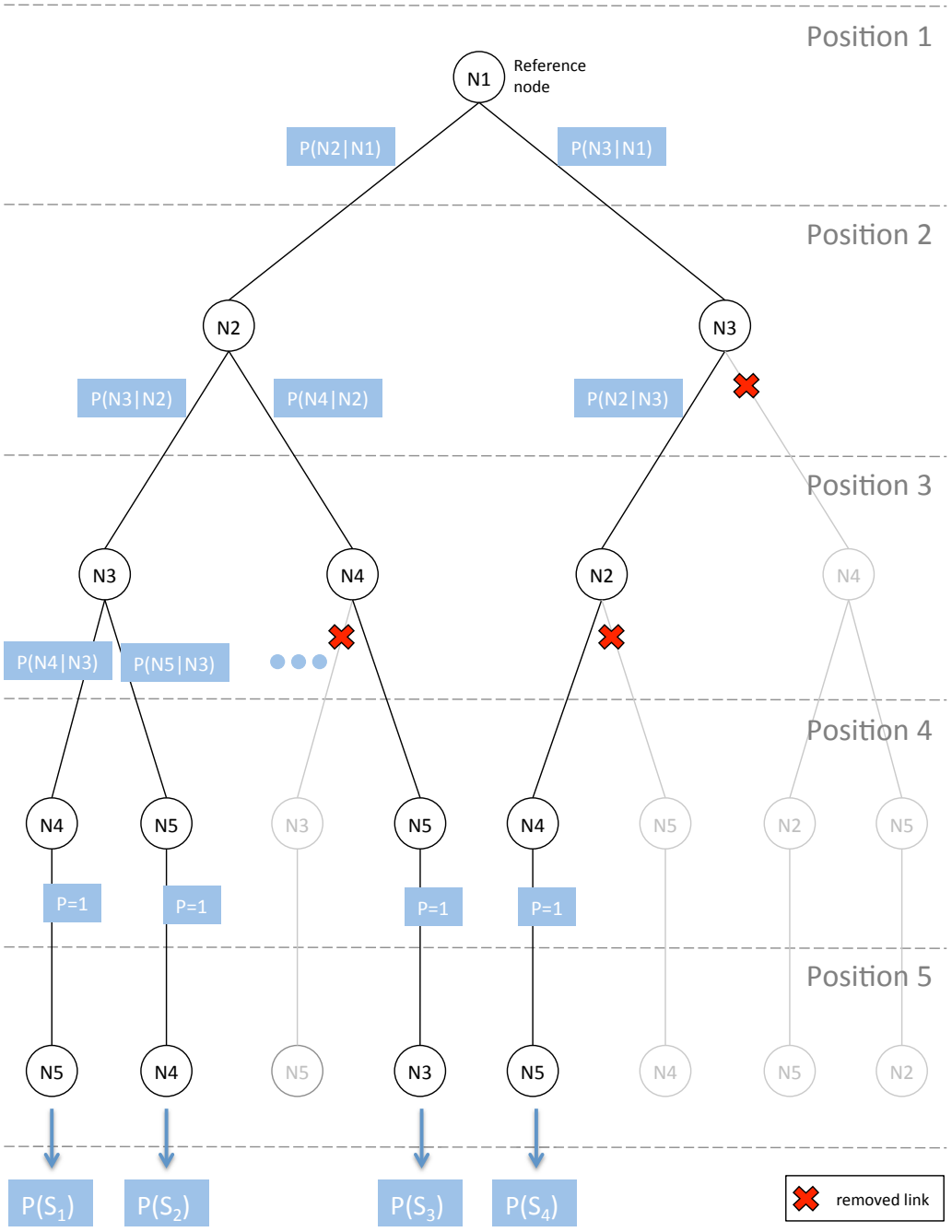


Figure 5.34: An illustration of a finalized probability tree with pruning. In this example $m = 1$ and selected sequence is $S = argmax(P(*S))$

Analysis of Required Number of Transmissions per Channel

Throughout this chapter we have validated our claims with experiments that used 40 transmissions per channel and per sender node. The computation was centralized, meaning that all the measurements needed to be collected at one processing center, which creates extra overhead. A distributed version of this system can be developed by the help of a distributed minimum spanning tree, which can be studied as a future step, but it was not considered within the current scope of this work.

Regardless of whether the implementation is distributed or centralized, it is logical to do as few transmissions as possible to save energy and time. Therefore, we downsized our measurements to fewer number of transmissions per channel while keeping the number of channels same. To analyse the impact of such a reduction in number of transmitted packets, we repeated the computations (results of which were presented in Figure 5.24) by filtering the input measurement dataset down to arbitrarily selected 1, 8 and 16 transmissions per channel. In Figure 5.35, we observe that the success ratio was not affected noticeably by the changing number of transmissions. This proves that the success we get with channel diversity is not because of the added number of transmissions into the computations (by adding more samples for different channels), but the changing of the multipath conditions by introducing the frequency diversity into the system.

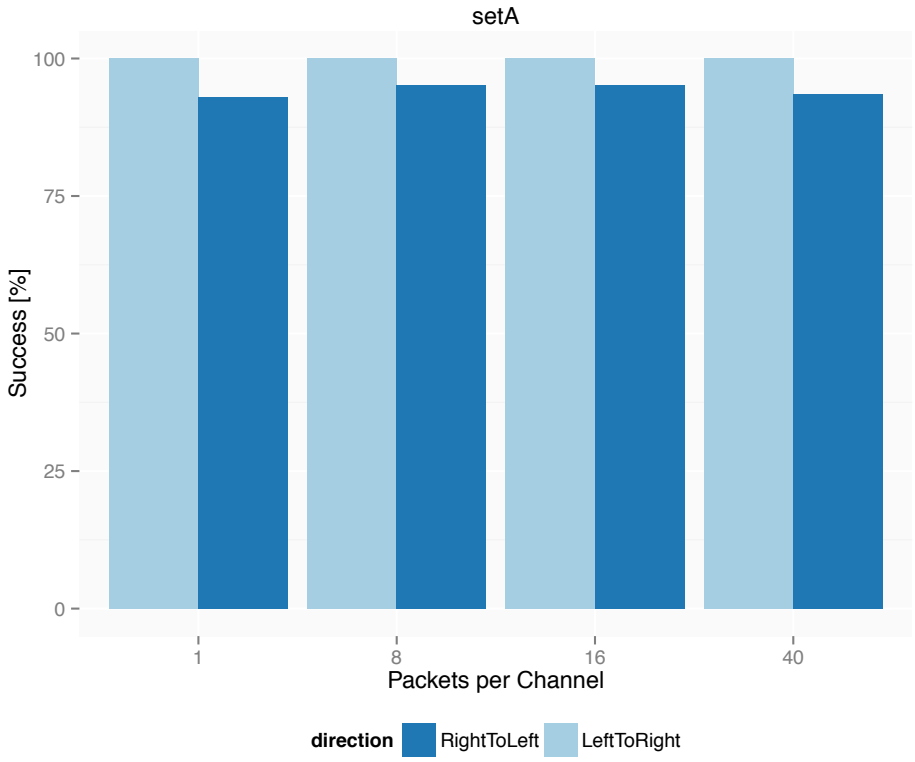


Figure 5.35: Impact of number of transmissions per channel, Set-A

We have also examined the impact of the change in the number of transmissions per channel for MDS in comparison to PNSD. In experiments with Set-A (LeftToRight direction), using fewer number of transmissions did not have a significant impact on MDS either. However, for the experiments with Set-C (RightToLeft direction), wherein the success ratio was worse for all of the algorithms in consideration, MDS got affected more severely than PNSD by the reduced number of transmissions. The comparison is shown in Figure 5.36.

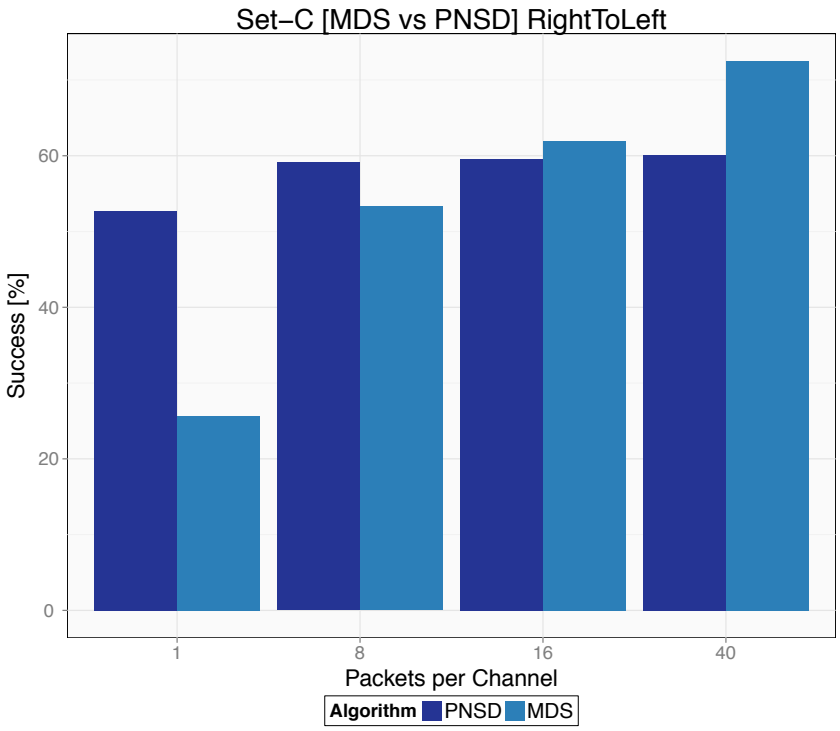


Figure 5.36: Impact of number of transmissions per channel, Set-C RightToLeft

And finally we examined the impact of different number of transmissions per channel on "reliability analysis" and we have noticed that using as many as 40 packets was indeed increasing the reliability of the results. This analysis on Set-A RightToLeft direction is shown in Figure 5.37.

To sum up, we have noticed that our suggested approach (PNSD) is considerably more robust to varying number of transmissions per channel while MDS is affected in some cases. However, PNSD benefits from larger number of transmissions per channel, resulting in increased levels of reliability.

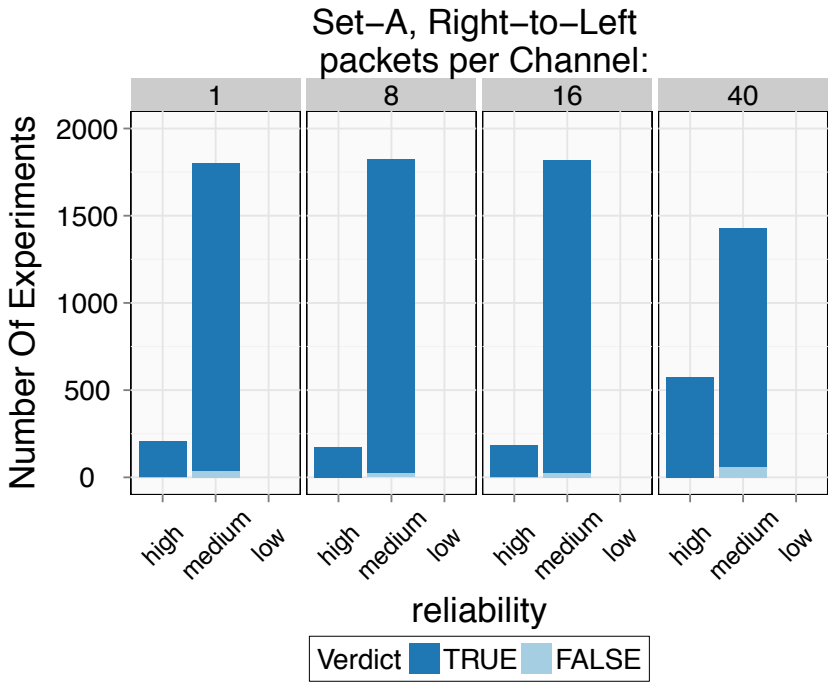


Figure 5.37: Impact of number of transmissions per channel on reliability, Set-A RightToLeft

5.4.6 Summary

In this section we presented a methodology, Probabilistic Node Sequence Discovery (PNSD), for discovering the node sequence in a linear configuration of wireless sensor networks. It was shown that, contrary to popular belief, the imprecise RSS measurements from low-power radio chips were useful for solving the problem of node sequence determination in challenging indoor environments. The results of this section demonstrated how using more of the available information (data in measurements) could allow a significant improvement in successful discovery of node positions. The tree-based computations created an added complexity, which was not the case in the previously introduced Greedy approaches. But by applying a pruning method on the probability tree, we could achieve a faster implementation of introduced node sequence discovery system.

Building on our claims and findings in the previous Sections 5.2 and 5.3, we have again shown that frequency diversity is beneficial for relating RSS measurements to position information. For the sake of fairness, the same frequency diversity was incorporated to other approaches to which our probabilistic position discovery system was compared to.

We have compared the suggested methodology (PNSD) to greedy sequence building algorithm and to well accepted MDS algorithm. The results showed that when the measurement medium was not affected heavily by external factors (as with Set-B), 100 % successful posi-

tion discovery was possible both by PNSD and MDS. However, when the measurement medium got challenged by interference or other types of anomalies, MDS and Greedy approaches got affected heavily while PNSD maintained its robustness (i.e. with Set-A, Set-C).

It was also observed that there were some occasions, in which the correct construction of the sequence was not possible. Despite hard efforts on determining the reasons for the instabilities in the raw data, leading to false sequences when our and other approaches were applied, we were not able to repeat the conditions that impaired the measurements. There have been periods of time when successful determination was possible and there have been other periods of time when correct determination of the sequence was not possible even after many repetitions. This has motivated us to create an additional quality assessment step.

By generating the reverse sequence from our initial result we are able to assess the reliability of the sequence estimates and even retroactively change the decision of the previous step to a more probable estimate. Extensive tests have shown that all the classification for which we could obtain "High" reliability verdicts have always been correct and the vast majority of those with "Medium" reliability verdicts have been correct, but rarely wrong. All the remaining wrong verdicts could be attributed to "Low" level of reliability. Using this information, users can accept the sequence with its provided reliability rank, or they might want to repeat the process until a "High" reliable sequence is found.

5.5 Chapter Summary and Conclusions

In this chapter some unconventional ways of relating RSS information to relative positions were provided, based on the findings in Section 5.2.

First, it is explained why frequency diversity can be used to determine comparative closeness of neighboring nodes. The experiment results verified that change in propagated signal frequency can cause a change in the measured strength of the received signal for as big as 20 dBm. Later, we concluded that this change has an upper limit (Ψ^{max}) and a closer node has an increased chance of measuring a greater local maximum RSS value than a node that is farther away, if the diversity in frequency is incorporated to the measurements. This epiphany has led us build node sequence discovery algorithms around this phenomenon.

Initially we used only the maximum RSS values measured by multiple receivers to find the sequence. After that, we showed how iteratively building the sequence using different ways of statistical processing could improve the results, such as through using mean of different percentiles in RSS measurements. The understanding that we gained from these techniques was; it was not always possible to hit the maximum possible RSS value within a given time and given environment. To increase the chances of finding better values to compare, averaging few highest RSS values across different receivers and across different frequencies was helpful to increase stability of the approach.

Although these methods provided good results, they were not taking some of the data, generated through the measurements, into consideration. In an attempt to increase accuracy in position discovery, we decided to increase the level of information to use and a probabilistic approach to sequence discovery, PNSD, was proposed.

Our probabilistic approach used the rankings from each measurement taken by each receiver

at each channel to calculate the empirical probabilities of closeness. Increasing the amount of information that is used to estimate the closeness of nodes also increased the chances of finding the correct sequence.

The results were verified by real-world measurements taken from TWIST testbed in a crowded office environment. Such measurements are considered being more representative for analysis of physical systems. However, it is difficult to make a generalization about the position discovery systems that are tested only in one type of environment. Although the basic version of the sequence discovery system was successfully demonstrated to live audiences multiple times, a simulation model was also developed for achieving a more general conclusion. These results confirmed that our algorithms perform better than alternative approaches, even under simulated environments.

Finally a verification system was developed in an attempt to filter the imperfect results. We call this verification system "reliability through reverse validation". The rationale we used was that if the sequence that is constructed in one direction matches another sequence that is constructed in the reverse direction, starting from the last node of the initial sequence, then the chances are high that it is the correct sequence. Likewise, if the computed sequence was incorrect, finding the exact same incorrect sequence again by starting from the other end of the initial sequence would be very low.

We confirmed this hypothesis by applying the validation algorithm on the results of PNSD. We could successfully sort out high, medium and low reliable results, by clustering the incorrect results mostly on the low rank of the reliability and the high rank of the reliability analysis contained only the correct results. The verification system was also able to select a better result, which increased the overall success rate of our algorithm.

These advancements in the node sequence discovery in a linear configuration of wireless nodes encouraged us to develop algorithms for two-dimensional configurations, in which the nodes are placed on a regular grid setting. The following chapter contains a two-dimensional adaptation of the PNSD system, which was developed in Section 5.4. Later in the following section of the next chapter, a more generalized position discovery system for equidistant two-dimensional grid settings is explained.

Chapter 6

Two-Dimensional Node Position Discovery

Methods and techniques for discovering the node sequence on a one-dimensional (1-D) setting, by using only RSS information, were developed and analyzed in the previous parts of this thesis. To map the nodes to their potential positions without using a ranging or training based system, it was necessary to extract the closeness information. It was demonstrated that utilizing frequency diversity in the measurements while collecting RSS readings was beneficial for identifying the closest node to a particular sender.

In this chapter, relative node position discovery on two-dimensional (2-D) settings will be studied. We will benefit from the findings of the previous chapter, such as data collection method and closeness estimation.

Two cases of position discovery in two dimensions will be examined. First one is an extension of previously introduced PNSD system to the two-dimensional case. Second one is a more generalized approach to relative node position discovery for equidistant two-dimensional grid settings. We measured the performance of our proposals in comparison with the MDS-MAP algorithm [131].

In this part of the work, a grid setting with cells along horizontal and vertical axes will be considered. This is a common case for many indoor environments, such as offices, hospitals or warehouses, if we take rooms or partitions of a building as cells of a virtual grid.

6.1 Extending PNSD to Rectangular Grids: Constrained Two-Dimensional Position Discovery

In the previous chapter the PNSD algorithm was introduced as a way of identifying the node sequence. In this part of the work, we will define an extension to PNSD for mapping a discovered node sequence to a specific type of two-dimensional grids, where the nodes are placed closer to each other on one dimension than on the other dimension.

This extension assumes a regular grid, each cell of which is occupied with one node of the network, and no node is left out. It is also assumed that this 2-D grid, which on x dimension

contains n_x nodes and on y dimension contains n_y nodes, has n cells. Therefore; $n = n_x \times n_y$.

We hold on to the prior prerequisite of having one reference node placed on one corner of the system. On each dimension, the distances between the adjacent nodes are roughly equal along one dimension and different than the other dimension; about d_x on the x axis, d_y on the y axis and $d_x \neq d_y$. The constrained case of 2-D setting is: d_x is sufficiently smaller than d_y, such that, of the two adjacent nodes to a node N in the system, the one on the x dimension must be able to be detected as being closer to N than the one on y dimension: $d_x < d_y$. We also need the dimensions of the grid (number of columns n_x and rows n_y) to know.

Under this constraint, a 1-D sequence of all the nodes will be populated by PNSD system. This sequence will then be "folded" at each $i \times {n_x}^{th}$ position, where $i = 1 \rightarrow n_y$, and eventually finish with a set of nodes that are distributed along the rows and columns of the grid, as given in Algorithm 9. We evaluated this method with simulations in the next part.

Algorithm 9 Folding 1-D Sequence into 2-D Grid

Step 1: Find the Sequence S using Algorithm 8

Step 2: Map S onto the Grid G of size $n_x \times n_y$

Step 3: Revert even numbered rows of G

The *constrained 2-D position discovery* algorithm first calculates a linear sequence, and then it folds the result into a two-dimensional grid. Visual illustrations of how a 1-D sequence (illustrated with red arrows) is folded into a 2-D grid are given in Figures 6.1 and 6.2. The black dots in these figures represent the nodes and the numbers next to them represent their assigned IDs. The arrows represent the direction of sequence discovery, and the corners in the red line show the folding points.

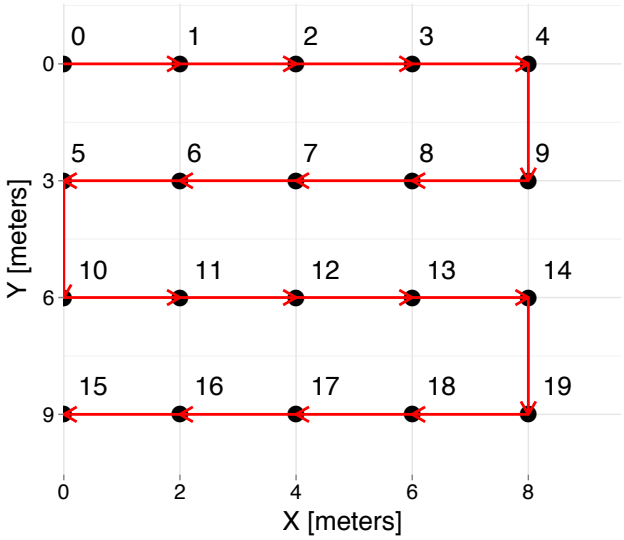


Figure 6.1: Illustration of constrained 2-D position discovery at a 5×4 setting

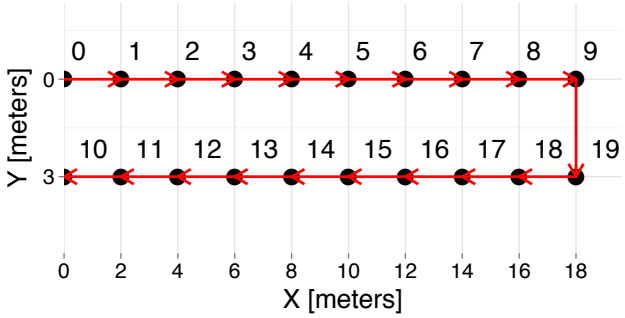


Figure 6.2: Illustration of constrained 2-D position discovery at a 10×2 setting

6.1.1 Evaluation

In the evaluation of this mechanism, two types of scenarios were assumed, in which the grids are shaped as rectangles of 10×2 and 5×4 cells and each cell contained one of the 20 nodes of a network. In each setting, the nodes were placed closer to each other on x dimension than on y dimension. Then, using the simulation model that was defined in the Section 5.4.3, RSS values corresponding to 40 measurements at each 16 channels of IEEE 802.15.4 standard were generated as a single dataset and 1000 such datasets were produced.

The simulation parameters were defined as:

- h: variance of Rayleigh distributed channel gain
- sd_n: variance of additive gaussian white noise (dBm)
- n_x: number of columns of the grid
- n_y: number of rows of the grid
- d_x: distance between each adjacent node on x dimension (meters)
- d_y: distance between each adjacent node on y dimension (meters)
- α: path loss exponent

Table 6.1: Simulation Parameters

Setting	**Grid Shape**	n_x	n_y	**Noise Level:** sd_n	h	α	d_x	d_y
1	5×4	5	4	moderate: -45 dBm	15	3	2 m	3 m
2	10×2	10	2					
3	5×4	5	4	high: -30 dBm				
4	10×2	10	2					

The parameters that were used in these simulations represented a highly challenging indoor scenario. We selected the variance of the Rayleigh channel gain as $h = 15$ which provides up to 20 dBm difference in simulated RSS values across the channels. The variance in white noise at -45 dBm level already causes up to 3 dBm fluctuation in the simulated RSS values within the same channel. The system was also tested with a greater level of white noise by setting its variance to -30 dBm. In both cases the path loss exponent α was taken 3. The parametrization of the simulations are given in the Table 6.1.

The measurements that are obtained from the simulations were used to construct the mapping of node positions with the Algorithm 9. The same measurements were also used with MDS-MAP algorithm for comparison. MDS-MAP is a connectivity based approach that utilizes MDS. Feeding connectivity data as the similarity metric to MDS gives a relative map of the node locations [131]. In our evaluations we used the average of all RSS values across all channels as the input for MDS-MAP and we used the Smacof implementation of MDS [68]. Then we converted these values to distances in meters by using Friis' path loss formula. MDS-MAP was not aware of the properties of the grid that the nodes are lying on and it generated node maps, which were often in a grid-like shape that was randomly rotated to a random direction. Following the directions in the MDS-MAP's paper, we first scaled MDS's coordinate system to match the grid coordinates of our scenario and rotated it until the reference node matched to the correct corner of the grid using procrustes analysis. Then we snapped the points in the MDS-MAP output to the closest points of the simulation nodes and we placed each node from the MDS-MAP result into a potential position in our system.

The results for both scenarios are shown in Figures 6.3 and 6.4. The plots show the success rate of constrained 2-D sequence discovery with that of MDS-MAP side-by-side. In the 5×4 scenario the performances of both algorithms did not get affected too much by the increasing white noise, but MDS-MAP has become totally unusable for the 10×2 scenario when the white noise was increased. Our algorithm provided over 81 % success rate in both grid shapes with moderate gaussian white noise. When the noise level was increased, the success rate of our algorithm did not get impacted too much and stayed over 77 % in both shapes of the grid.

MDS-MAP has performed very good, when the grid setting was of size 5×4, which resembles roughly a square. However, when the grid shape resembled a lengthy rectangle, as in 10×2 scenario, its success rate diminished. In that case, the calculated node mapping has taken a U shape rather than a rectangle, an example of which is shown in Figure 6.5.

This showed the robustness of our algorithm against to the lengthiness of the 2-D grid.

In section 5.4.4, a technique to analyse the reliability of the calculated sequence was explained. Since the 2-D grid that is generated here is a folded form of the 1-D output of the PNSD, we could apply the same algorithm on these results, too. Figures 6.6 and 6.7 show results of the success and reliability analysis when the noise level was medium. Figures 6.8 and 6.9 are the results of reliability analysis at the high noise level.

The results that were classified as "high reliable" were near 100 % correct and most of the incorrect results were successfully classified as "low reliable". These results are very promising for a system that does not rely on any preconfiguration or training.

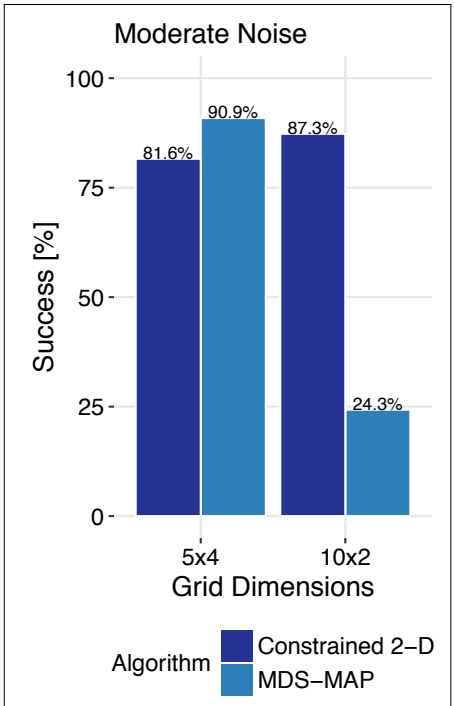


Figure 6.3: Constrained 2-D vs MDS-MAP, moderate noise

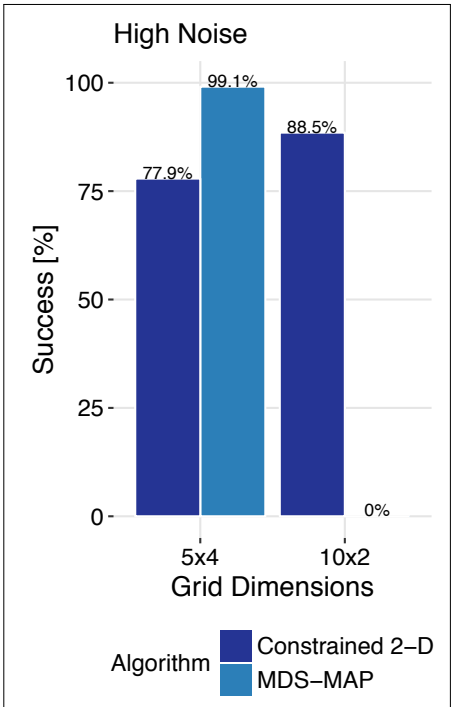


Figure 6.4: Constrained 2-D vs MDS-MAP, high noise

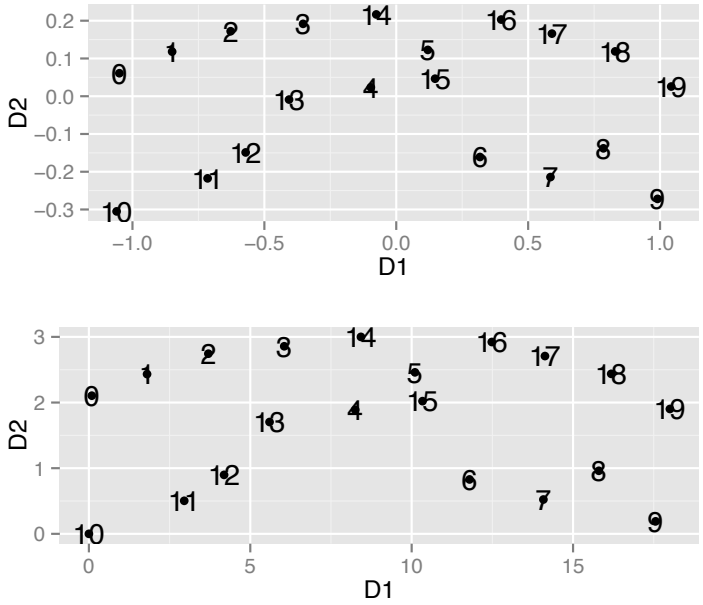


Figure 6.5: Plotted MDS-MAP output (top) and it's mapping to the metric geography (bottom), one sample

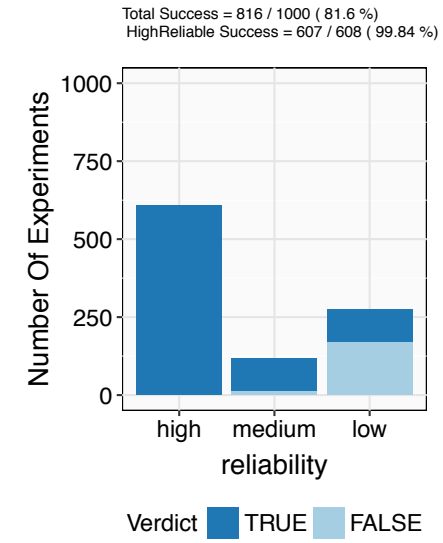


Figure 6.6: Reliability analysis at moderate noise level, 5×4 grid setting

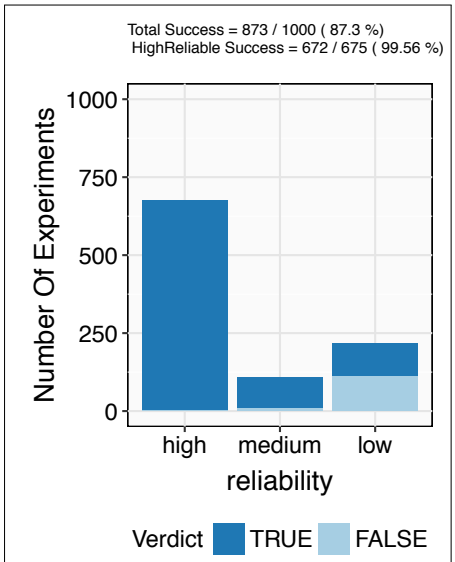


Figure 6.7: Reliability analysis at moderate noise level, 10×2 grid setting

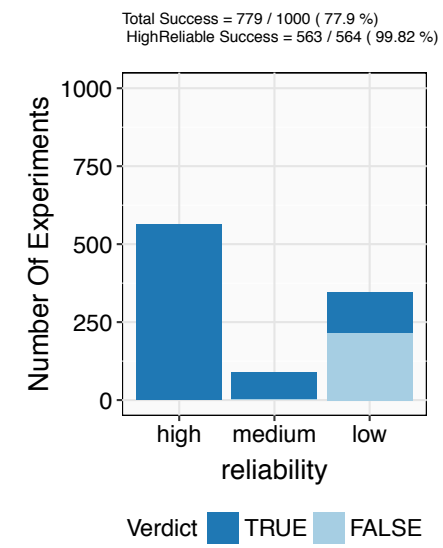


Figure 6.8: Reliability analysis at high noise level, 5×4 grid setting

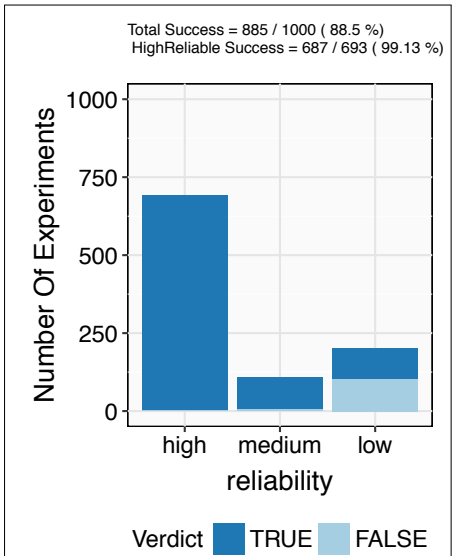


Figure 6.9: Reliability analysis at high noise level, 10×2 grid setting

6.1.2 Summary

In this part, an extension to the PNSD system is introduced for the two-dimensional settings under the condition that inter-node distances on the x-axis are different than the inter-node distances on the y-axis. The inputs to this model are 1) the prior knowledge of the node at one of the corners of the grid and 2) the knowledge of which dimension contains smaller inter-node distances.

The system works by first creating a one-dimensional sequence of all the nodes in the grid. After this sequence is created, the system folds the sequence at the ends of the rows and maps the result to the two-dimensional grid.

In this system, it is required that the inter-node distances along one axis of the grid are adequately different than the distances along the other axis. As a future work, this difference can be analysed and the limitations for this kind of applications can be identified.

In the following section, a more generalized system for discovering relative node positions on an equidistant two-dimensional grid setting will be discussed.

6.2 Two-Dimensional Node Position Discovery on an Equidistant Grid

In the previous section, the constrained 2-D node position discovery system, which was based on the PNSD methodology defined in Chapter 5, was discussed. That system required the 2-D grid setting to have different sizes in between the adjacent nodes on one axis than the other axis.

In this part of the work, we will discuss how the two-dimensional relative node position discovery can be performed on the nodes that are placed in a grid setting, where the nodes are approximately, but not strictly, at equal distances to each of their immediate neighbours at x and y dimensions. These distances between the pairs of nodes can be as small as half a meter to a few meters. A small subset of these nodes are taken as "Reference Nodes" and their positions in the grid are known to us. In this part, the subset of reference nodes will contain two or three nodes, in oppose to one node in the previous parts. The content of this section is published in our paper [132].

6.2.1 Discovering the Closest Node in a Two-Dimensional Setting

The effectiveness of frequency diversity for altering the propagation path of the transmitted signal and how combined RSS measurements from one transmitter to multiple receivers could be leveraged to detect the node that is most likely to be the closest receiver to that particular transmitter was discussed in the previous chapter.

In this part we will use a slightly different procedure to detect a set of possibly closest nodes (candidates) to a node. The procedure runs between a sender node and a set of receiver nodes. The beacon messages are sent in the same way as in the prior sections: by all nodes as senders in turns, transmitting a number of beacon messages at all channels available to the radio chip.

The receivers listen to these transmissions and record the measured RSS values for each transmission. At the end of the transmission campaign, each receiver returns a summary of the measurements, which is the mean value of measured RSS values for each channel that are greater than θ percentile of all values measured on that channel, similar to the procedure explained at the end of the Section 5.3.2. A sender transmits $e \times z$ packets at each of the total e channels, and each receiver reports e summaries back, one for each channel.

The sender node N_i compares the collected RSS summaries for each channel and puts the ID of the node that reports the biggest RSS summary for each channel into a vector, V_i. Also, the ID of any other node that reports within the RSSI resolution range of the radio chip $\pm\xi$ dBm to the biggest value of RSS summary in V_i is added into that vector V_i, because there are at least two nodes at equal distances to each sender. The mode (most frequent element) of this vector, $mode(V_i)$, is the node that is most likely be the closest node to that particular sender. In our algorithms we also use the second mode (second most frequent element) of the candidate vector as well, which we denote with $mode_2(V_i)$. Eventually, each sender generates a vector of candidates with a minimum size of e elements. This procedure is summarized in Algorithm 10.

Algorithm 10 findCandidates(Node N_i, threshold=θ, resolution=ξ, includeList = L)

Step 1: N_i transmits z packets at each one of the e channels ($z \geq 1$)

Step 2: Each receiver node, $N_r \in L$, in the communication range measures RSS at each one of e channels and reports the average of RSS measurements that are greater than θ percentile ($\overline{RSS^{\theta}}$) at each channel back to N_i

Step 3: N_i produces a vector, V_i, of receiver nodes that reported the $\pm\xi\ dBm\ of\ max(\overline{RSS^{\theta}})$ value at each channel

Return: V_i

The ratio of the frequency of a node in the vector to the length of the vector gives the probability of a node being the closest node to a particular sender, as in Equation 6.1.

$$P(N_r|N_i) = \frac{\text{frequency of } N_r \text{ in candidate vector } V_i \text{ of } N_i}{\text{size of } V_i} \tag{6.1}$$

Based on the procedure *findCandidates()* and the probability of closeness calculation above, we define two procedures for discovering the mapping of the nodes to the 2-D grid. First procedure focuses on minimizing the a priori knowledge (positions of reference nodes) required for the relative node position discovery system and second procedure focuses on increasing the robustness and the success rate of the position discovery.

6.2.2 Grid-Based Position Discovery with Two Reference Nodes: GBPD-2

This approach assumes that two nodes placed on two corners of the grid that share one edge are known to us as *reference nodes*, such as shown in Table 6.2. The number of rows and columns are initially not known. We call this system Grid Based Position Discovery with 2 reference nodes (GBPD-2).

$\boldsymbol{R1}$	$G_{1,2}$	$G_{1,3}$	...	$G_{1,m}$
$G_{2,1}$	$G_{2,2}$	$G_{2,3}$	...	$G_{2,m}$
$G_{3,1}$	$G_{3,2}$	$G_{3,3}$	...	$G_{3,m}$
...				
$G_{(n-1),1}$	$G_{(n-1),2}$	$G_{(n-1),3}$	...	$G_{(n-1),m}$
$\boldsymbol{R2}$	$G_{n,2}$	$G_{n,3}$	...	$G_{n,m}$

Table 6.2: Example placement of reference nodes R1 and R2 in Grid **G**

The system first discovers the nodes along the common edge shared with reference nodes. The discovery is initiated by one of the reference nodes selecting two candidates as the immediate neighbor along the edge. If the node that selects the candidates is positioned at $G_{r,c}$ of the grid G (cell at r^{th} row and c^{th} column), then the candidates should be on cells $G_{r+1,c}$ and

$G_{r,c+1}$ as they are the geographically closest cells. Then the other reference node selects one of these two candidates as the node closer to itself, indicating that it belongs to the common edge. This procedure starts from the first edge (column) of the grid ($c = 1$), and iterates until all nodes between two reference nodes along the same edge are discovered. We will use the notation $G_{r,c}$ both as a cell in the grid G and as the node occupying that cell, since a one-to-one mapping is targeted. Each candidate-finding includes the set of nodes that are not yet discovered in the grid, which we denote with G'. In this procedure, which is given in Algorithm 11, first node discovers two most likely closest nodes to itself as candidates and the second node chooses the closest node from these two candidates.

Algorithm 11 findEdge(Node $N1$, Node $N2$) ▷ GBPD-2

```
EdgeList ← {} ; M1 ← M2 ← null
while (M1 OR M2) ≠ N2 do
    V1 ← findCandidates(N1, θ, ξ, includeList=G')
    M1 ← mode(V1)
    M2 ← mode2(V1)
    V2 ← findCandidates(N2, θ, ξ, includeList={M1,M2})
    EdgeList.add(mode(V))
end while
Return: EdgeList
```

$\boldsymbol{R1}$				
$G_{2,1}$				
$G_{3,1}$				
...				
$G_{(n-1),1}$				
$\boldsymbol{R2}$				

Table 6.3: Discovery of first row of Grid **G** using Algorithm 11: findEdge()

Discovering the first edge of the grid provides us the number-of-rows information. Knowing that this edge is the first column of the grid, each consequent column is discovered one cell at a time, in a row-by-row basis. During the procedure, a node in $G_{r,c}$ computes its candidate list, L1, using the Algorithm 10. If L1 is bi-modal (having two *modes*), then the node in the next row of the same column, $G_{r+1,c}$ computes another candidate list, L2, and its mode is put in $G_{r+1,c+1}$. This eliminates the bi-modality of $G_{r,c}$. Then $G_{r,c}$ again computes its candidate list L1 and eventually the mode of L1 is placed in $G_{r,c+1}$. This procedure is iterated for each row in a column, until all nodes are placed in the grid. The whole procedure is given in Algorithm 12 and it is illustrated in the Table 6.4.

$\boldsymbol{R1}$	$G_{1,2}$			
$G_{2,1}$	$G_{2,2}$			
$G_{3,1}$				
...				
$G_{(n-1),1}$				
$\boldsymbol{R2}$				

Table 6.4: Discovery of further cells of Grid **G** using Algorithms 10 and 12
(solid red lines symbolize past computations, dashed red lines symbolize future computations)

Algorithm 12 buildGrid(Node $R1$, Node $R2$) ▷ GBPD-2

```
firstColumn = findEdge(R1, R2)
G(..., 1) ← firstColumn
c ← 2                                          ▷ starting from second column
while unplaced nodes exist do
    for r in Each Row do
        L1 ← findCandidates(G_{c-1,r}, θ, ξ, includeList=G')
        if L1 is bi-modal and G_{c-1,r+1} exists then
            L2 ← findCandidates(G_{c-1,r+1}, θ, ξ, includeList=G')
            G_{c,r+1} ← mode(L2)                       ▷ skip next iteration
            L1 ← findCandidates(G_{c-1,r}, θ, ξ, includeList=(G'))
        end if
        G_{c,r} ← mode(L1)
    end for
end while
```

6.2.3 Grid-Based Position Discovery with Three Reference Nodes: GBPD-3

This time the connected network of the nodes is assumed to have **three** reference nodes, $R1$, $R2$ and $R3$, and these reference nodes are assumed to occupy the positions $G_{1,1}, G_{1,2}$ and $G_{2,1}$ of the Grid G, as shown in Table 6.5. The size of at least one of the dimensions and the total number of nodes are also assumed to be known to us.

The first step is discovering the inner corner of these reference nodes (e.g. $G_{2,2}$ in Table 6.6), completing the mapping of discovered nodes to a 2×2 square grid.

Next, the adjacent neighbors on either sides of the square is discovered: 2 nodes for the vertical side ($G_{1,3}$ and $G_{2,3}$), 2 nodes for the horizontal side ($G_{3,1}$ and $G_{3,2}$). Now the system has 8 nodes of a 3×3 grid discovered as illustrated in Table 6.7.

The iteration continues with the discovery of the missing corner node ($G_{3,3}$) of the square grid, and repeats until one of the edges of the Grid G is completely covered. Then the discovery is iterated with edges only, until all the nodes are positioned. For this procedure, we define three

$\boldsymbol{R1}$	$\boldsymbol{R2}$	$G_{1,3}$	...	$G_{1,m}$
$\boldsymbol{R3}$	$G_{2,2}$	$G_{2,3}$	...	$G_{2,m}$
$G_{3,1}$	$G_{3,2}$	$G_{3,3}$	...	$G_{3,m}$
...				
$G_{(n-1),1}$	$G_{(n-1),2}$	$G_{(n-1),3}$	...	$G_{(n-1),m}$
$G_{n,1}$	$G_{n,2}$	$G_{n,3}$	...	$G_{n,m}$

Table 6.5: Example node placement in Grid **G**

$\boldsymbol{R1}$	$\boldsymbol{R2}$			
$\boldsymbol{R3}$	$G_{2,2}$			
...				

Table 6.6: Discovering a corner node in Grid **G**

$\boldsymbol{R1}$	$\boldsymbol{R2}$	$G_{1,3}$		
$\boldsymbol{R3}$	$G_{2,2}$	$G_{2,3}$		
$G_{3,1}$	$G_{3,2}$			
...				

Table 6.7: Discovering side nodes in Grid **G**

methods: **fc** (find corner), **fte** (find top edge), **fe** (find edge), which are given in Algorithms 13, 14 and 15 respectively. The procedure for building the grid (**buildGrid()**) itself is given in Algorithm 16 and illustrated in Figure 6.10 as a flow diagram.

An example iteration of buildGrid() algorithm is shown in Table 6.8. The cells are tagged with labels {iteration_number:function}.

R1	**R2**	2:fte	7:fte	14:fe
R3	1:fc	4:fc	9:fc	15:fe
3:fte	5:fc	6:fc	11:fc	16:fe
8:fte	10:fc	12:fc	13:fc	17:fe

Table 6.8: Iterations and Functions

Algorithm 13 fc(Node $N1$, Node $N2$, Node $N3$) ▷ find corner, GBPD-3

$V_1 \leftarrow$ findCandidates($N1$, θ, S, includeList = G')
$V_2 \leftarrow$ findCandidates($N2$, θ, S, includeList = V_1)
$V_3 \leftarrow$ findCandidates($N3$, θ, S, includeList = V_1)
Return:
cornerNode $\leftarrow mode(V_2 + V_3)$

Algorithm 14 fte(Node $N1$, Node $N2$) ▷ find top edge, GBPD-3

$V_1 \leftarrow$ findCandidates($N1$, θ, S, includeList = G')
$c1 \leftarrow mode(V_1)$
$c2 \leftarrow mode_2(V_1)$
Return:
$arg\,max(P(c1|N1) \times P(c1|N2), P(c2|N1) \times P(c2|N2))$

Algorithm 15 fe(Node $N1$, Node $N2$) ▷ find edge, GBPD-3

$V_1 \leftarrow$ findCandidates($N1$, θ, S, includeList = G')
if V_1 is bi-modal and $N2$ exists **then**
    $V_2 \leftarrow$ findCandidates($N2$, θ, S, includeList = G')
    $V_1 \leftarrow$ findCandidates($N1$, θ, S, includeList = (G'$-mode(V_2)$))
end if
Return:
$mode(V_1)$

Algorithm 16 buildGrid(Node $R1$, Node $R2$, Node $R3$) ▷ GBPD-3

$n_x \leftarrow$ number of Grid rows
$n_y \leftarrow$ number of Grid columns
Assumption: $n_x < n_y$
$G_{2,2} \leftarrow fc(R1, R2, R3)$ ▷ 2×2 square shape is achieved
for d in 3 to n_x **do**
    $G_{1,d} \leftarrow fte(G_{1,d-1}, G_{1,d-2})$
    $G_{d,1} \leftarrow fte(G_{d-1,1}, G_{d-2,1})$
    for j in 2 to $d-1$ **do**
        $G_{j,d} \leftarrow fc(G_{j-1,d-1}, G_{j-1,d}, G_{j,d-1})$
        $G_{d,j} \leftarrow fc(G_{d-1,j-1}, G_{d-1,j}, G_{d,j-1})$
    end for
    $G_{d,d} \leftarrow fc(G_{d-1,d-1}, G_{d-1,d}, G_{d,d-1})$
end for
for d in $n_x + 1$ to n_y **do**
    for r in 1 to n_x **do**
        $G_{r,d} = fe(G_{r,d-1}, G_{r+1,d-1})$
    end for
end for

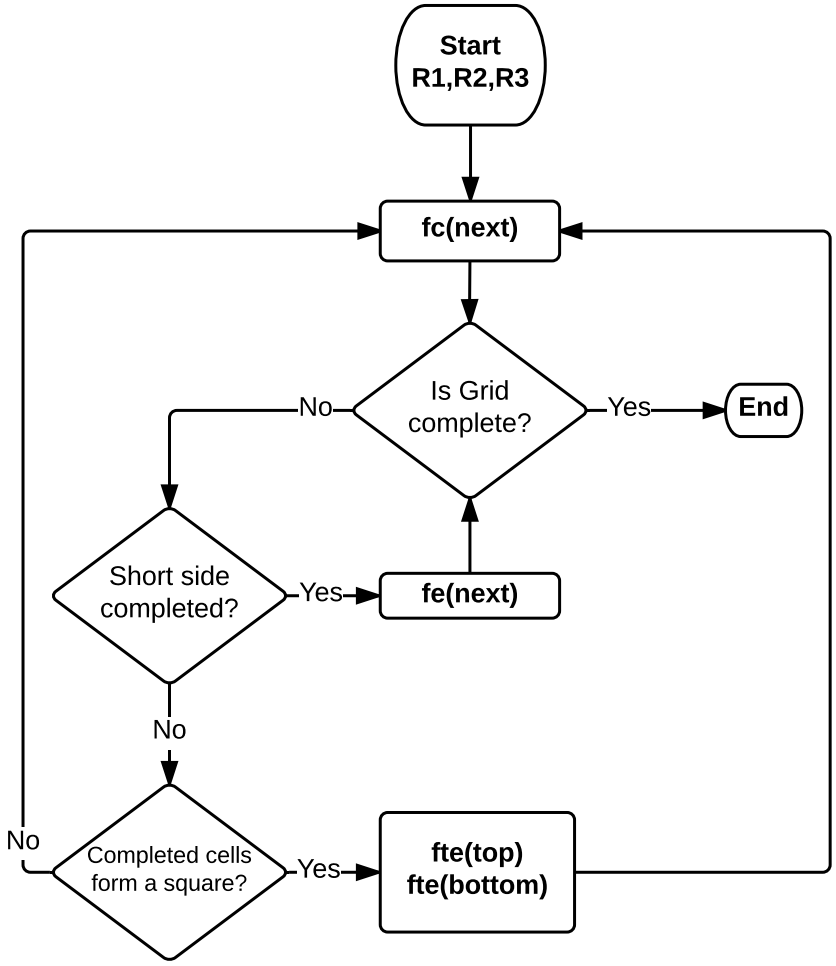


Figure 6.10: Flowchart of GBPD-3, showing how *find corner (fc)*, *find top edge (fte)* and *find edge (fe)* procedures cooperate. *next*, *top* and *bottom* stand for the next cells to be discovered.

6.2.4 Reliability Analysis

Due to the iterative nature of the process, an error in the midst of the grid would effect discovery of all the remaining nodes from that error on. We therefore verify the first and the last edges for consistency and assess how likely it is that all the nodes are positioned in the grid at the right places. Algorithm 11 ($findEdge$) is used to discover the mapping of a set of nodes to an edge of a two-dimensional grid. This algorithm takes the two ends of the edge as input and works iteratively from one end to the other end.

First, one edge is discovered, which contains two of the reference nodes, starting from one corner node towards the direction of the reference node $R1$. Then, the nodes at the opposite corners of the grid ($G_{1,m}$ and $G_{n,m}$) are taken and the last column is re-discovered using Algorithm 11 to compare with what has already been found in the grid. Finally, the last column is re-discovered in the reverse node order and the same comparison is applied. The reliability of the result is determined with a value, which we call "score" that takes 0 as its initial value. Each of these three comparisons add 1 to the current value of "score" if a match is found. At the end, the value of score corresponds to a reliability classification. This procedure is given in Algorithm 17 and it is designed for grids with more than two rows and two columns.

Algorithm 17 Calculate Reliability Score

$score = 0$
if $findEdge(G_{n,1}, G_{1,1}) == Grid[G_{1,1}, G_{n,1}]$ **then**
    score = score+1
end if
if $findEdge(G_{1,m}, G_{n,m}) == Grid[G_{1,m}, G_{n,m}]$ **then**
    score = score+1
end if
if $findEdge(G_{n,m}, G_{1,m}) == Grid[G_{n,m}, G_{1,m}]$ **then**
    score = score+1
end if
return score

Where **Grid**[a,b] is the set of nodes already mapped into one edge between the positions a and b of a grid G by one of the GBPD-# algorithms.

And the reliability value will be:

$$reliability = \begin{cases} high & \text{, if } score == 3 \\ medium & \text{, if } score == 2 \\ low & \text{, if } score \leq 1 \end{cases}$$

This reliability assessment method is applicable to both GBPD-2 and GBPD-3 algorithms.

6.2.5 Evaluation

To analyze proposed algorithms (GBPD-2 and GBPD-3) we used the same simulation model that generates RSS values between any two nodes as a function of distance and random Rayleigh channel gain, for an indoor environment. The validity of our simulation model was introduced and verified in Section 5.4.3 .

The same parametrization of the attenuation model as in the previous section was also used here. We considered two channel gain conditions, namely **moderate noise** (with variance of -45 dBm) and **high noise** (with variance of -30 dBm). In our model, -45 dBm of noise variance causes RSS values to deviate about $1-2$ dBm, which is also the precision of the CC2420 radio chip. A noise variance of -30 dBm causes the readings under the same conditions to fluctuate about $4-5$ dBm, which is possible in very noisy indoor environments. The fading gain variance h was taken as 15. Both random variables have zero mean. Path loss exponent α was taken as 3, which is typical for indoor office environments.

For our evaluations we generated RSS values according to the simulation parametrization above. We fed our GBPD algorithms, the MDS-MAP and the Fingerprinting (Nearest Neighbor) algorithms with these RSS values that are generated by the simulator. MDS-MAP does not require anchor points, but only reference points to correct the scale and orientation of its output. Therefore, the amount of required information is the same as in our GBPD algorithms. The same amount of information can also be used for applying fingerprinting, which is a technique frequently used method for indoor positioning. So it is natural to ask, whether the fingerprinting might work for this kind of 2-D scenarios. We used the Smacof implementation of MDS in MDS-MAP. The linear transformations of the MDS results using the anchor nodes, were applied using procrustes analysis [133], which is implemented in the Vegan package for R [128]. For both MDS-MAP and fingerprinting algorithms, we used the averages of the RSS values across all channels as input.

We used the original fingerprinting approach [100], since the measurement points were always same as the training points and mobility of the nodes was not considered. When we talk about node position discovery in a static setting with definite partitions, RSS-based fingerprinting is a very popular method that finds wide use. We trained the fingerprinting system with 5 % of the measurement sets and the rest of the measurements were used for testing.

In this part, too, the experiment results are identified either as a "success" or as a "fail", where "Success" indicates correct discovery of positions of **all nodes in the grid setting** and "fail" indicates otherwise. Each of the following experiment scenarios was repeated 100 times.

First we evaluated a topology of 20 simulated nodes, placed into the cells of a 5×4 grid, as shown in Table 6.9. Each node was considered to be placed 3 meters away from its immediate neighbor at the horizontal and vertical directions. We applied both GBPD-2 and GBPD-3, as well as MDS-MAP and Fingerprinting algorithms, on the same set of RSS values. The results are plotted in Figure 6.11 for two-reference-nodes scenario and in Figure 6.13 for three-reference-nodes scenario. For two-reference nodes case, we assumed that the positions of the corner nodes, N0 and N15, were known to us. Likewise, N0, N1 and N5 were assumed to be known to us for the three-reference-nodes scenario and the rest of the system was to be discovered. Other algorithms in our comparisons were also given the same set of reference (anchor) nodes for the sake of fairness.

In addition to the success ratio, we also computed the "reliability" of the results. Figure 6.12 shows the classification of position discovery verdicts for the 5×4 topology using 2 reference nodes with GBPD-2 algorithm. Figure 6.14 shows the same information for GBPD-3 algorithm, which uses 3 reference nodes. Then we increased the size of the network, as well as the grid, and performed the node position discovery on the new dataset that was produced with the same simulation parameters. Figures 6.15 and 6.17 show the performance of GBPD-2 and GBPD-3 on a 10×5 topology with 50 nodes, also in comparison with MDS-MAP and Fingerprinting algorithms. The reliability analysis results for 10×5 topology are given in Figures 6.16 and 6.17 respectively.

N0	**N1**	N2	N3	N4
N5	N6	N7	N8	N9
N10	N11	N12	N13	N14
N15	N16	N17	N18	N19

Table 6.9: 5×4 Grid of node placement for simulations

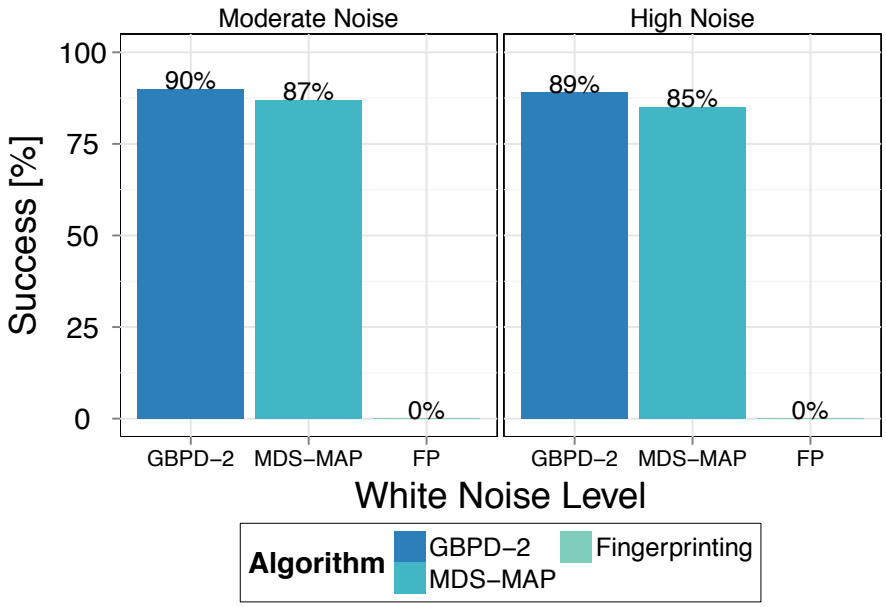


Figure 6.11: GBPD-2: Grid based position discovery with two reference nodes, 5×4 topology

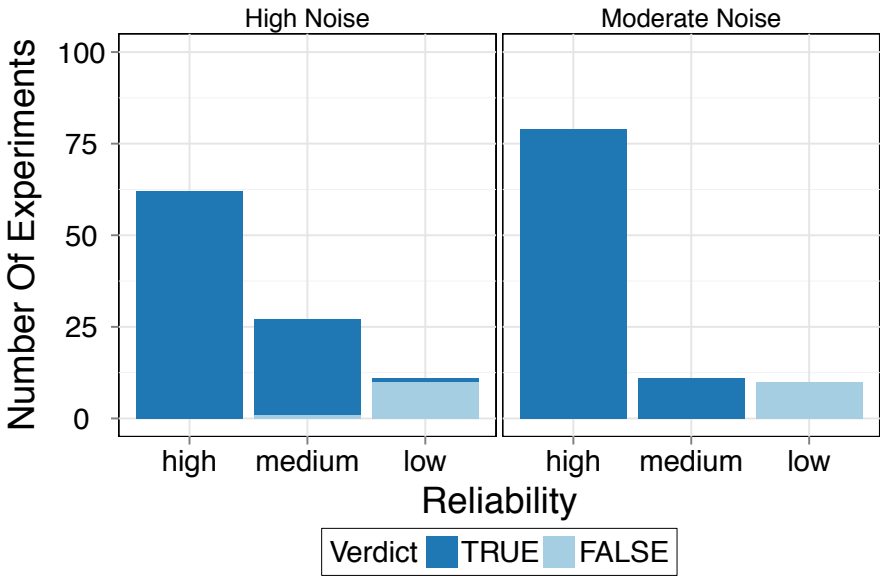


Figure 6.12: GBPD-2: Reliability analysis of the results, 5×4 topology

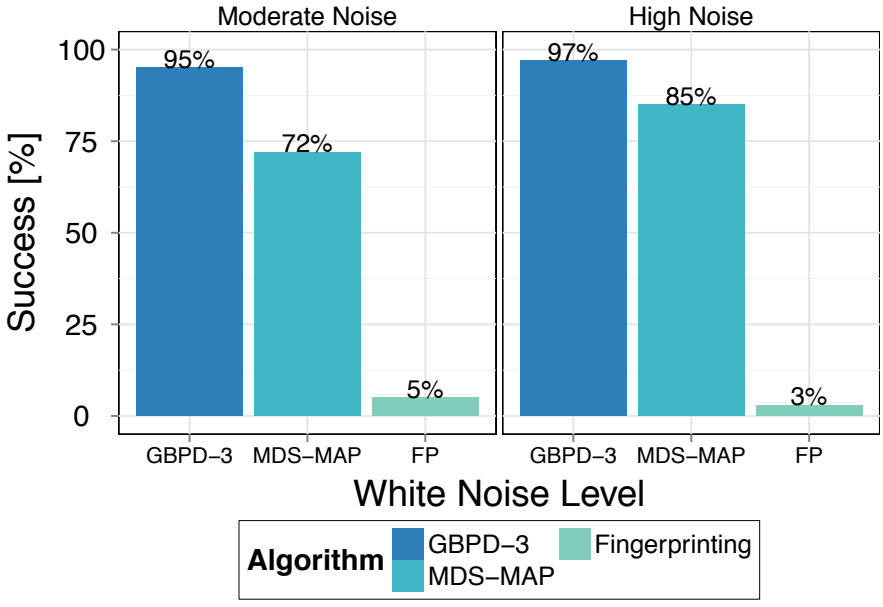


Figure 6.13: GBPD-3: Grid based position discovery with three reference nodes, 5×4 topology

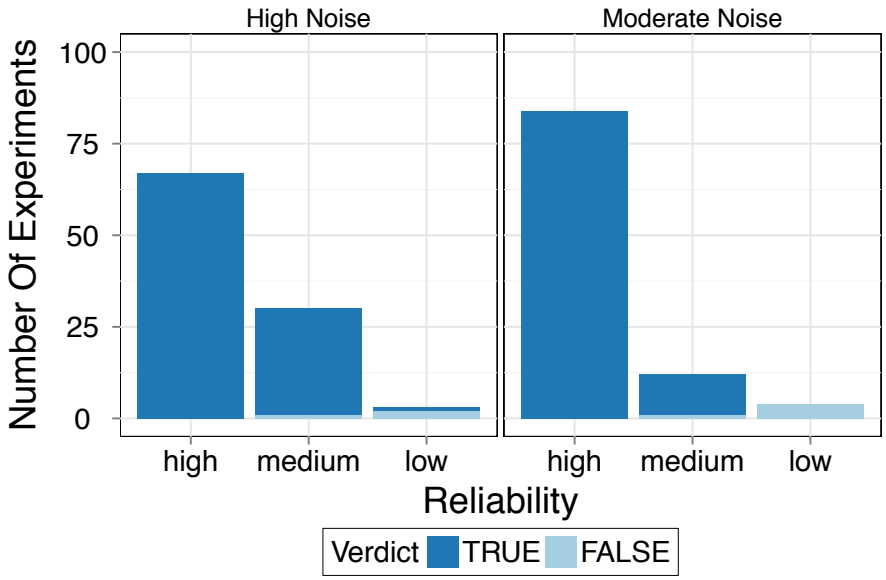


Figure 6.14: GBPD-3: Reliability analysis of the results, 5×4 topology

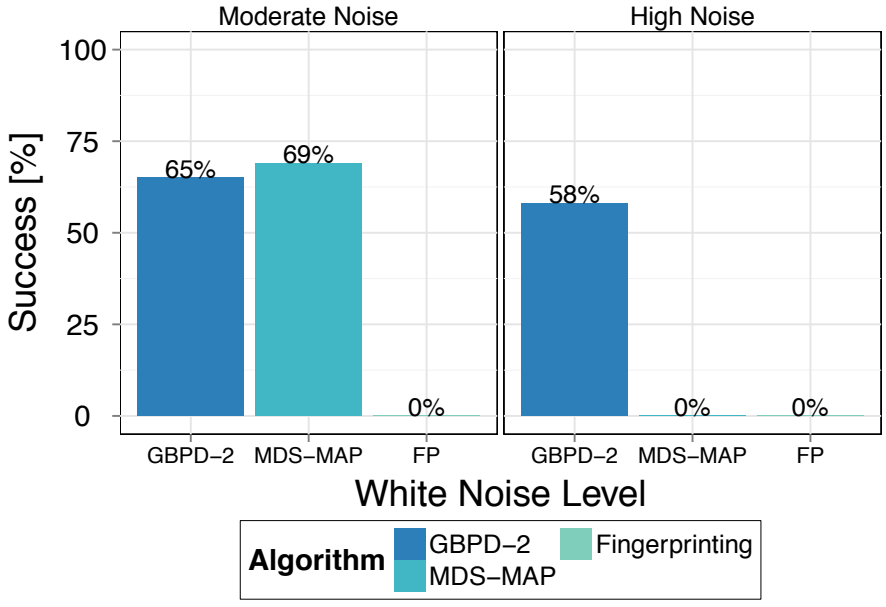


Figure 6.15: GBPD-2: Grid based position discovery with two reference nodes, 10×5 topology

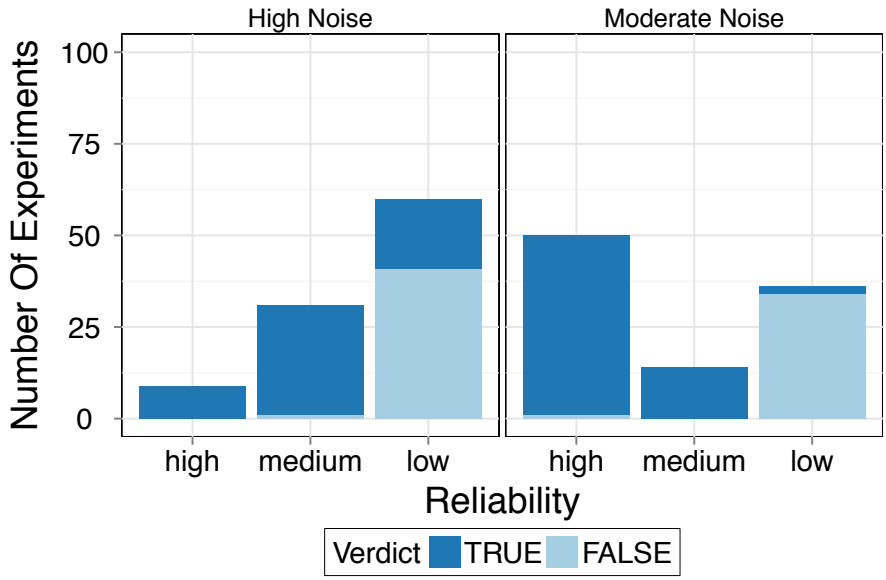


Figure 6.16: GBPD-2: Reliability analysis of the results, 10×5 topology

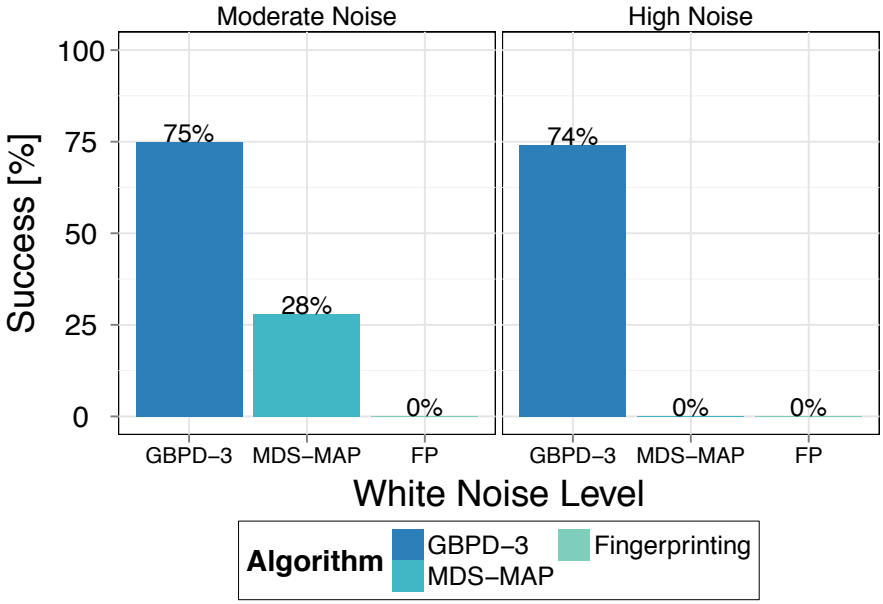


Figure 6.17: GBPD-3: Grid based position discovery with three reference nodes, 10×5 topology

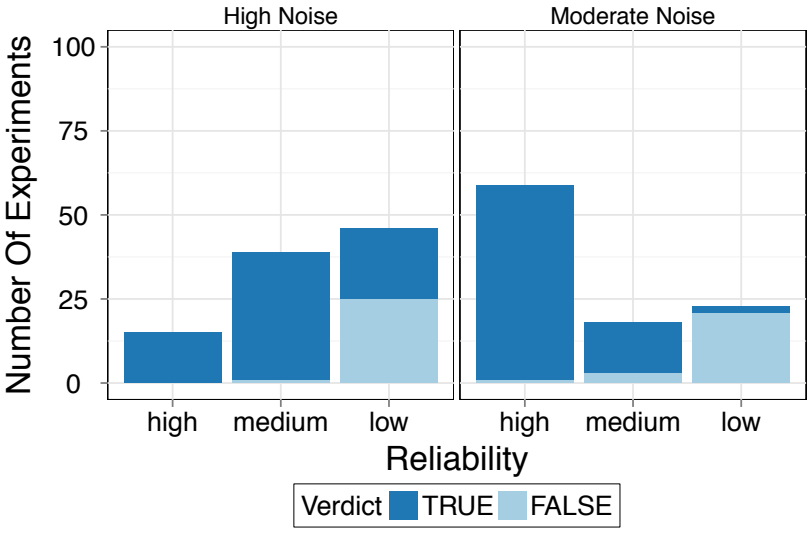


Figure 6.18: GBPD-3: Reliability analysis of the results, 10×5 topology

Results show that using such a low number of anchor points (reference nodes) for Fingerprinting is not sufficient. In certain cases, such as 5×4 topology, which is closer to a square shape, MDS-MAP offers a satisfactory performance but fails to maintain stability when the topology diverges from the square shape, such as for a 10×5 shaped topology. Also, it is more sensitive to the noise level of the wireless medium than the GBPD approaches.

Here we need to mention a special case of MDS-MAP performance: if it was given three anchor nodes at three different corners of a grid that has roughly a square shape, such as the 5×4 setting, it was able to produce 100 % correct results in moderate noise conditions. For higher noise or lengthy topology settings, such as a 10×5 grid, the performance of MDS-MAP did not improve by providing three anchor nodes at three different corners.

The reliability analysis has shown that the proposed method could successfully categorize a vast majority of imperfect mappings (Verdict: FALSE) to "low" reliability level, whereas almost 100 % of the results that were categorized in the "high" reliability level were the correct mappings (Verdict: TRUE) in both channel conditions and both topologies.

In conclusion, the GBPD algorithms are more robust to the changes in the noise levels in the environment and they scale better than the alternative approaches. Even though the increased noise levels did not immensely affect the success rates, it was visible that the results were less often in the high reliability category.

6.2.6 Summary

In this part we investigated solutions for discovering relative two-dimensional positions of multi-channel wireless sensor nodes in equidistant 2-D grid settings. The proposed systems require

the nodes' communication radio interfaces to support multiple channels and the initial position information of two or three nodes.

It has been shown that, by combining frequency diversity with statistical reasoning, discovery of relative positions of WSN nodes is possible with high precision in dense two-dimensional equidistant deployments. Based on the findings in the previous chapters, we developed a simulation to reflect the behavior of an indoor medium and we have shown that the suggested system outperforms MDS-MAP and Fingerprinting algorithms. Finally, we assessed the results of the grid building algorithms for their reliabilities (high, medium or low), which can hint to the user whether a particular measurement cycle needs to be repeated.

6.3 Chapter Summary and Conclusions

This chapter focused on discovering relative node positions on two-dimensional settings for densely deployed indoor scenarios of WSNs. The first technique that we introduced is an extension to the Probabilistic Node Sequence Discovery (PNSD) algorithm for the 2-D grid settings, where the distances between the node pairs were adequately different on the horizontal dimension than on the vertical dimension.

The second system that we proposed is a more generalized case, in which the nodes are considered to be positioned on an equidistant 2-D grid setting, in which the adjacent pairs of nodes has similar distances to each other on both vertical and horizontal dimensions.

These proposed techniques were verified using a simulation model that reflected crowded indoor office conditions. The proposed techniques were compared to the commonly accepted position discovery methods; MDS-MAP and Fingerprinting. Our algorithms demonstrated both better success in correctly identifying the node positions and also more stability over changing channel and topology conditions. In both cases, the proposed reliability computation technique allowed us to classify our position discovery results in one of the high, medium or low categories.

Chapter 7

Conclusions

Humans spend most of their time indoors. We interact with other humans, as well as objects inside buildings. The objects that we use indoors are often not placed randomly, but in an orderly fashion, if they are needed to be obtained quickly or if their positions are important information. There is an upward trend in enhancing the physical objects with Wireless Sensor Network and Internet of Things systems by providing them communicational capabilities and we can use such enhancements also for recovering the locations of the objects. There are cases, where the locations of these objects fit roughly into regular grid settings. Such grid settings can be single-dimensional in a linear form, such as along a corridor, or they can be two-dimensional, such as rooms in hospitals or racks in warehouses.

Unavailability of GPS signals indoors and added costs of deploying indoor positioning infrastructure motivated us to utilize available Received Signal Strength (RSS) information that is provided by most common radio chips during the communication packet exchange. In the literature, it is widely argued that RSS is not a good indicator of distance due to the instability of the wireless medium, multipath and other wireless interference, which we also confirmed in this work. We have, however, shown that mapping nodes of a low-power wireless sensor network to their potential positions in a regular grid can be done without calculating the geographical x, y coordinates of the nodes, but such a mapping can be achieved through available RSS data, by extracting the closeness information between the pairs of nodes in the network. We have also discussed why traditional position discovery techniques that compute geographical coordinates as the node positions are not usable in crowded indoor settings in Section 3.2.

This work proposes methods for discovering positions of the nodes relatively to each other in a densely deployed WSN, by leveraging the frequency diversity in radio communications and statistical reasoning. Unlike many range-free position discovery systems, we do not require a sparse deployment of the nodes in the network. Each node is allowed, but not required, to exist within the transmission range of any other node.

The position discovery systems that are proposed in this work produce a mapping between a set of N nodes and a set of N positions in grid shape. For measuring their performances, a binary metric is used: success of fail. Success in the results means that N nodes could be mapped into N positions without any error. Fail in the results means that the assignment of nodes to positions was not perfectly correct.

First, we developed mechanisms for detecting physical closeness of nodes to each other by leveraging frequency diversity. Using the closeness indication, techniques for matching the nodes to their potential positions were developed.

We have validated the suggested algorithms and techniques using real-world experiments and simulations. The real-world experiments were done in a crowded indoor office environment. Different floors and sides of the experimentation building were used for achieving more diverse conditions. The fact that this environment was a university building, where wireless research was in focus, made the wireless medium even more challenging. In these real-world experiments, the proposed algorithms could achieve successful discovery of relative node positions in large number of repetitions at a higher rate than the competing mechanisms. These results were supported by simulations, which represented different node placement and channel conditions.

Under realistic conditions, full success of a position discovery algorithm can never be guaranteed due to the unpredictable nature of wireless channel conditions. Therefore, the reliability assessment systems were developed as an added quality metric and the results were assigned to predefined $high$, $medium$ and low reliability categories. The results that were assigned to the $high$ reliability category were correct for more than 99 % of the time. The results that were assigned to the $medium$ reliability were correct for most of the time and the majority of the unsuccessful results were assigned to the low reliability category. This way a user can decide whether the result of the position discovery system should be accepted, or if the measurements need to be repeated until a result of category $high$ is achieved. Within the scope of this work, we did not analyze limitations on inter-node distances. Results of such an analysis would very much depend on the measurement medium (building) and the measurement hardware. Instead of such an analysis on distance and environmental limitations, we have developed the reliability analysis system and we let the users determine their own limitations based on their hardware and environment. Nevertheless, it is worth saying that, in our real-world experiments and live demonstrations we have achieved successful discovery of the nodes in a linear setting for the inter-node distances as small as 50 cm and higher.

7.1 Contributions

In the following, an outline of the major contributions of this thesis is given:

- It has been confirmed that some very small displacement of the transmitter antenna can have a significant impact on the multipath. A receiver node can measure noticeably different RSS values on the same channel as a result of such small changes in the antenna position.

- It is shown that the multipath conditions that affect the wireless signals can also be altered by changing the wavelength (frequency) of the signals. Such a change in the signal frequency can cause a variation in the measured strength of the received signal for up to 20 dBm in our observations.

- It has been shown that using only a single or a few highest measured RSS values help determining the closest receiver node to a sender node, by reducing the randomness in the

impact of multipath.

- We have presented easy-to-implement iterative sequence discovery algorithms, which leverage imprecise estimates of relative node closeness by using the proposed multichannel RSS measurement technique.
- We have proposed a probability-tree-based method —Probabilistic Node Sequence Discovery (PNSD)— for more precise detection of the node sequence at the cost of a higher computational complexity. This method was improved by a quality assessment technique, "reliability through reverse validation", to classify the reliability of the results as high, medium or low. Additionally, the speed of computations was improved by utilizing a probability tree pruning technique.
- Based on the findings and the developed techniques for the node sequence discovery, two techniques for recognition of relative node positions in a two-dimensional grid setting were proposed. First technique is a minimum-knowledge discovery system that requires only two reference nodes and the second technique is a more precise version that requires three reference nodes. These techniques were also supported by reliability assessment methods that are customized for them.

In summary, we have shown that relative node position discovery is possible with high precision in indoor environments without any preconfiguration and training phase. The proposed systems require the nodes to be deployed on a regular-grid-like setting. Only the position information from one or few of the nodes in the network is needed as a starting point. The rest of the discovery is performed with the help of the RSS information, which is often available in the regular communication of the nodes in a network. Frequency diversity substantially increases the chances of successful relative position discovery in these settings.

The results that are presented throughout this study are derived from more than 161 million separate peer-to-peer measurements in an office environment. This dataset, which is made online at [134], can be helpful for studying different fields of research, such as wireless channel modeling. Considering that the measurement time alone was above 300 hours, reutilizing this dataset can save a lot of precious research time. The developed software for performing measurements is also made available online [135] and new measurements can be collected using compatible hardware platforms, independent of the testbed.

7.2 Future Work

Progressive advancements in the algorithms and approaches that are developed during this work indicate potentials for further optimizations. For example, a cluster analysis algorithm can be used to create groups of nodes according to the estimates of relative closeness. These clusters can then internally be processed for discovering the relative positions of their nodes locally, and later the clusters can be chained to each other for generating a global view. Dividing the nodes into clusters based on physical proximity can increase the chances of a successful position discovery and decrease the speed of computations.

Another future topic can be detecting the bad channel conditions during the process of collecting RSS data and hence abruptly interrupting the position discovery system to save energy until better channel conditions are detected.

In the scope of this thesis unoccupied cells in the grids were assumed inexistent, but such cases can exist. An extension on gap detection can enable the support for more kinds of applications, in which the number of the nodes are less than the number of their potential positions.

The systems that are proposed throughout this work start with imprecise views of closeness between the reference nodes and the other nodes. Then the mapping of the nodes are performed in a one-by-one fashion. A genetic algorithm can help achieving the final mapping faster by starting from an imprecise global view and then it can mutate the current mapping into a better one. Such a technique can integrate two-way measurements from each pair of nodes and it can converge faster into a state, in which the mapping of nodes cannot be improved with further mutations.

Finally, the system can be expanded for supporting three-dimensional settings, which may require knowledge of extra reference nodes.

Appendix A

Observations and Raw Values of Single-channel RSS measurements

This Appendix contains the observations of single-channel RSS measurements. For each set of measurements, top figures show the probability density distributions[1] of the raw RSS values, which are shown in the bottom figures. The tagged data points are the relevant values for using in the sub-hypotheses in Chapter 4.

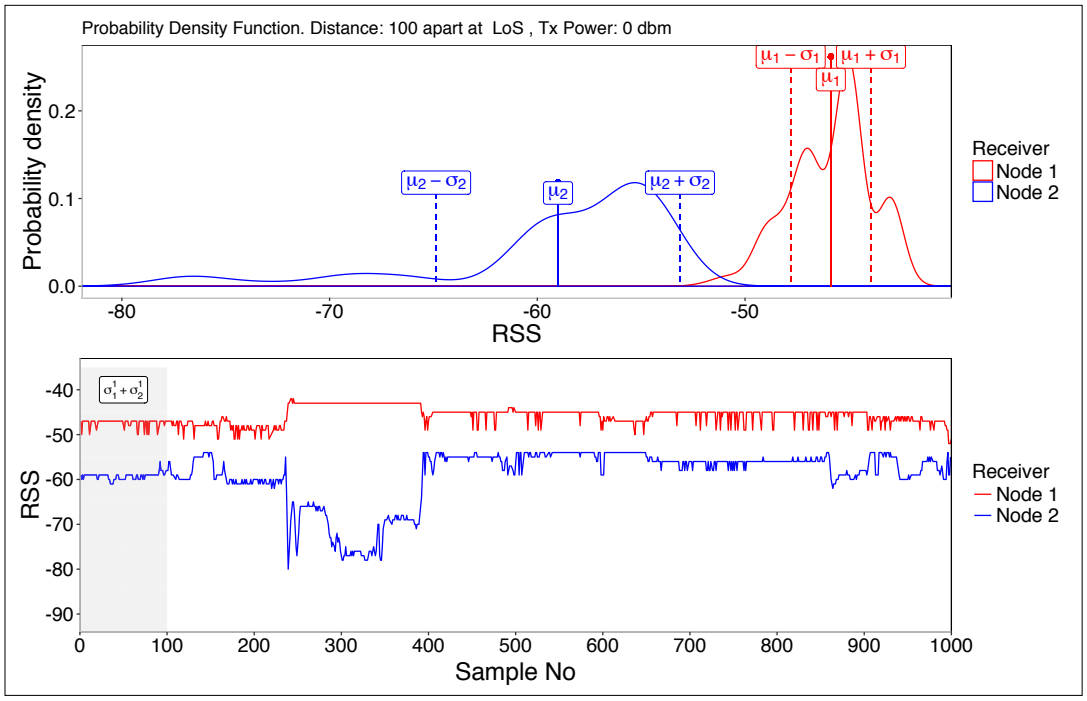


Figure 8.1: 100 cm apart nodes; reference node Radio Tx Power: 0 dBm

[1]Here it should be noted that a probability cannot be greater than 1. However some plots show values greater than one, it is due to not having enough variation in the data to generate a probability density distribution. The area under the probability density lines always add up to 1.

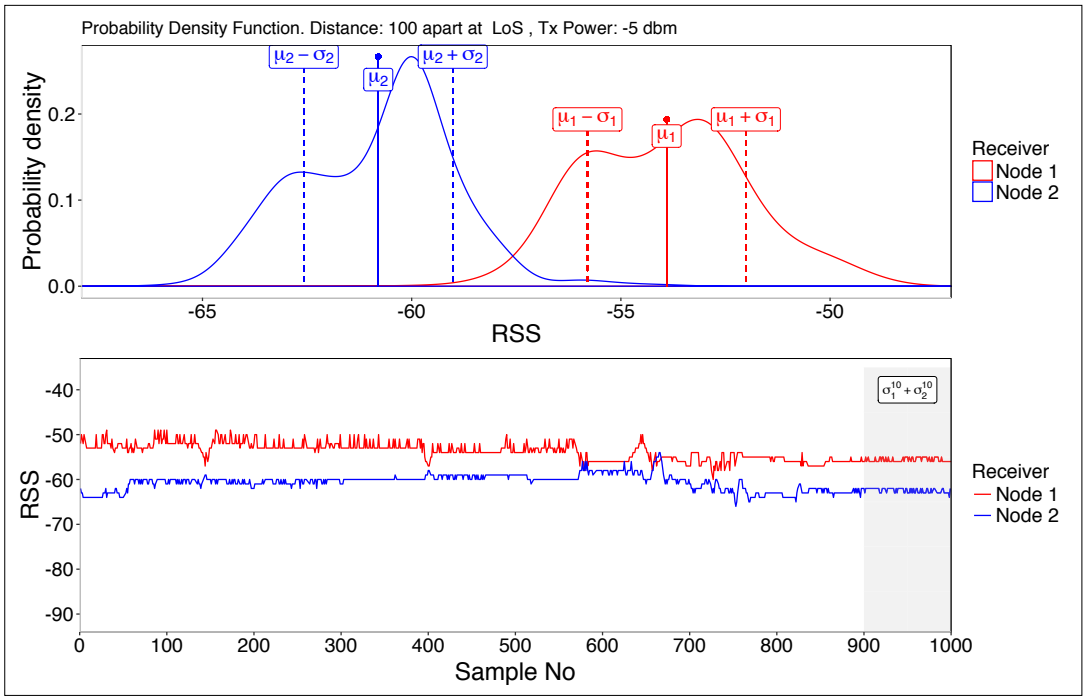


Figure 8.2: 100 cm apart nodes; reference node Radio Tx Power: -5 dBm

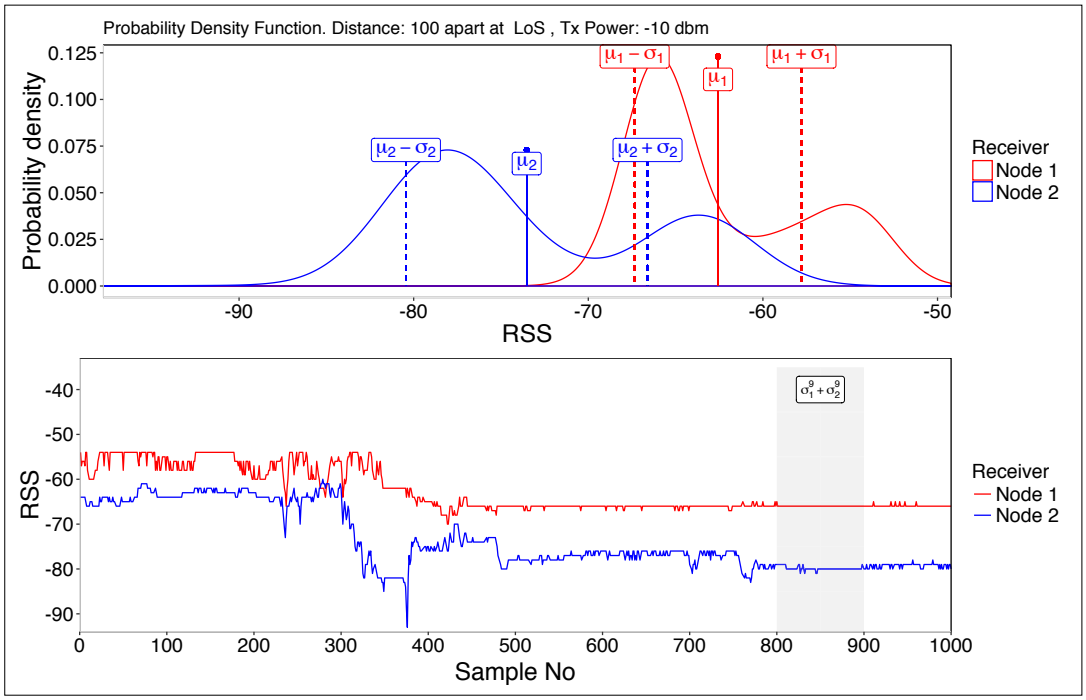


Figure 8.3: 100 cm apart nodes; reference node Radio Tx Power: -10 dBm

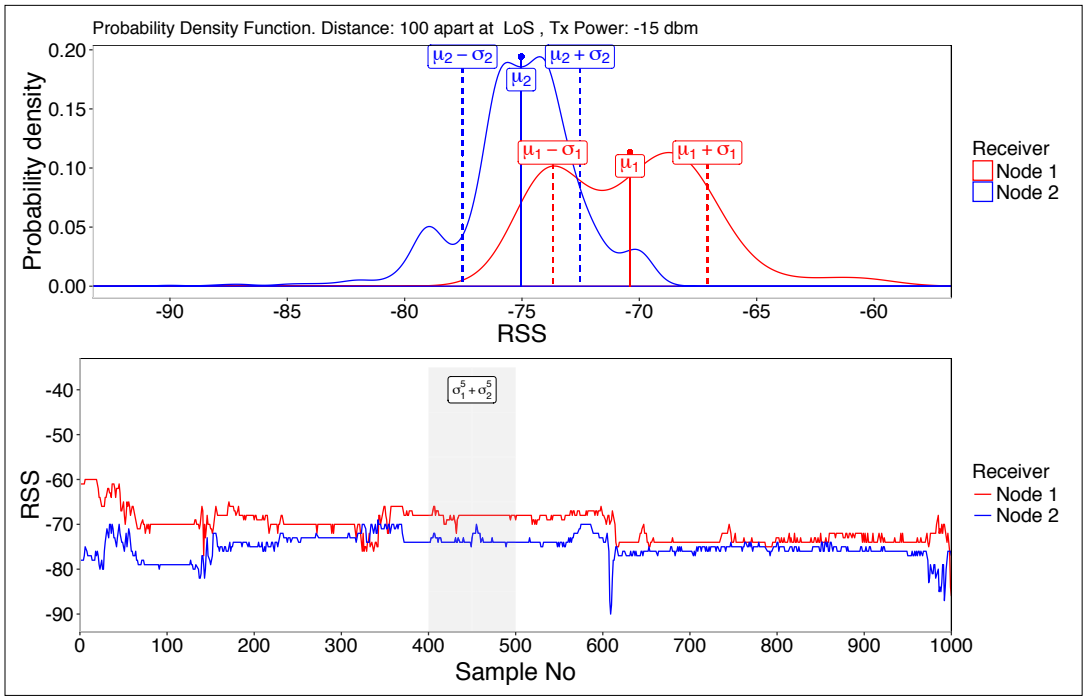


Figure 8.4: 100 cm apart nodes; reference node Radio Tx Power: -15 dBm

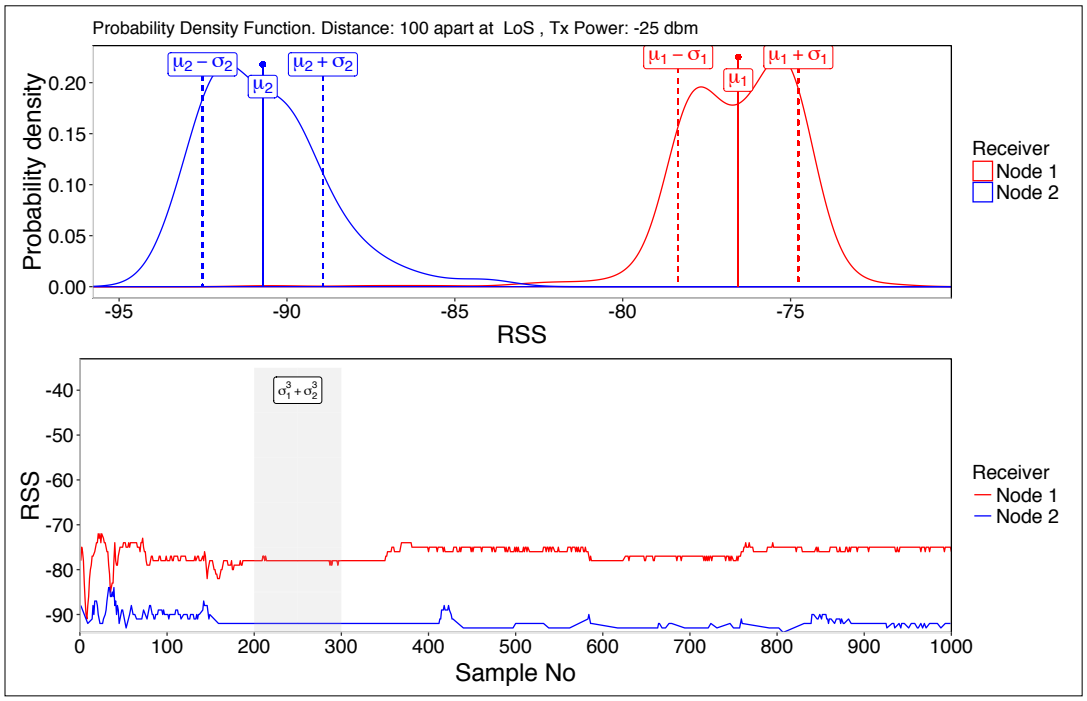


Figure 8.5: 100 cm apart nodes; reference node Radio Tx Power: -25 dBm

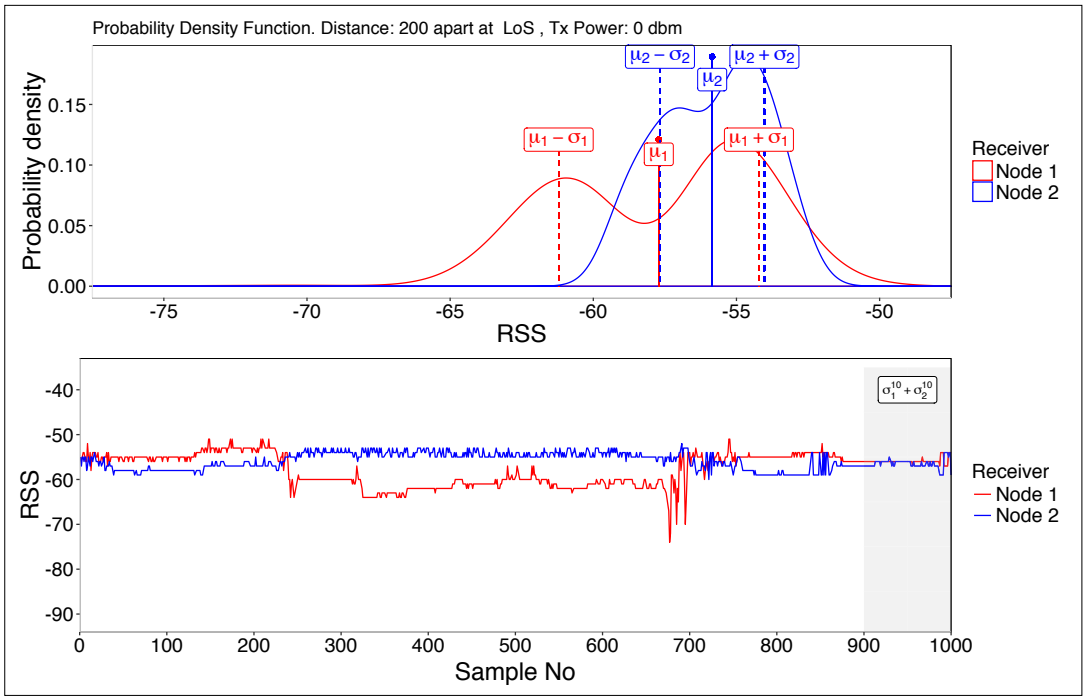


Figure 8.6: 200 cm apart nodes; reference node Radio Tx Power: 0 dBm

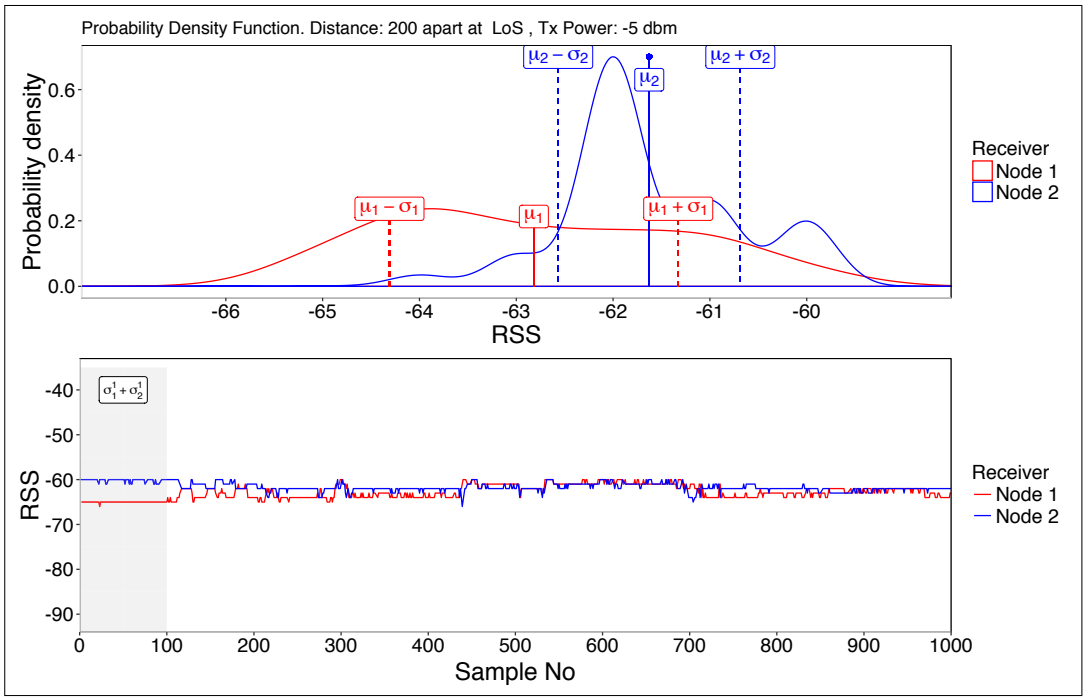


Figure 8.7: 200 cm apart nodes; reference node Radio Tx Power: -5 dBm

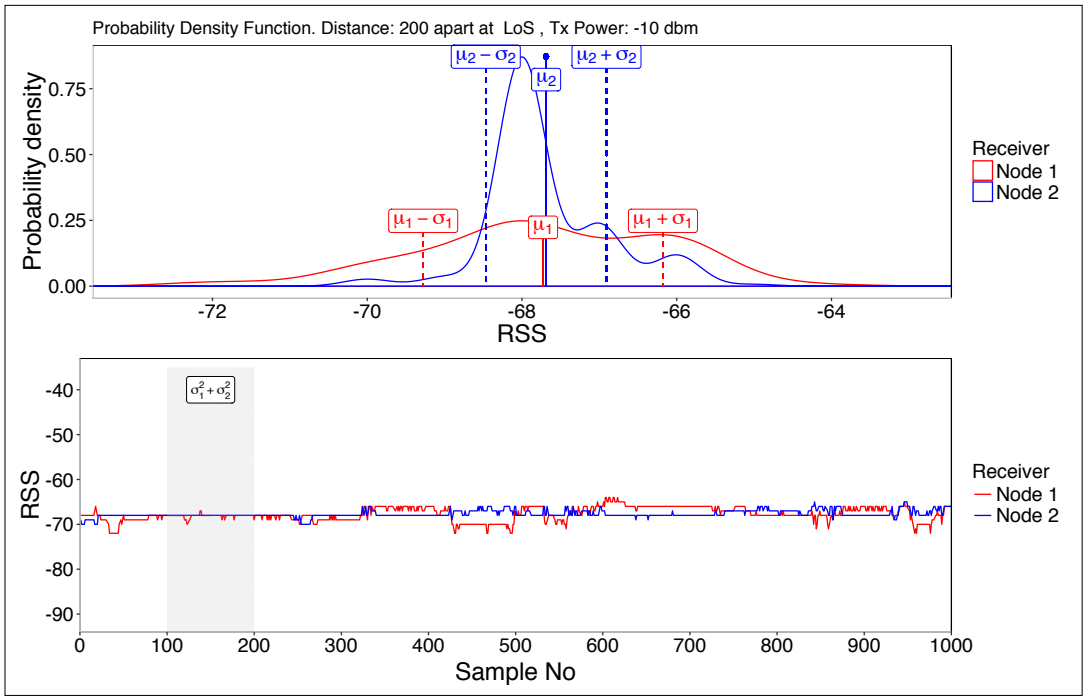


Figure 8.8: 200 cm apart nodes; reference node Radio Tx Power: -10 dBm

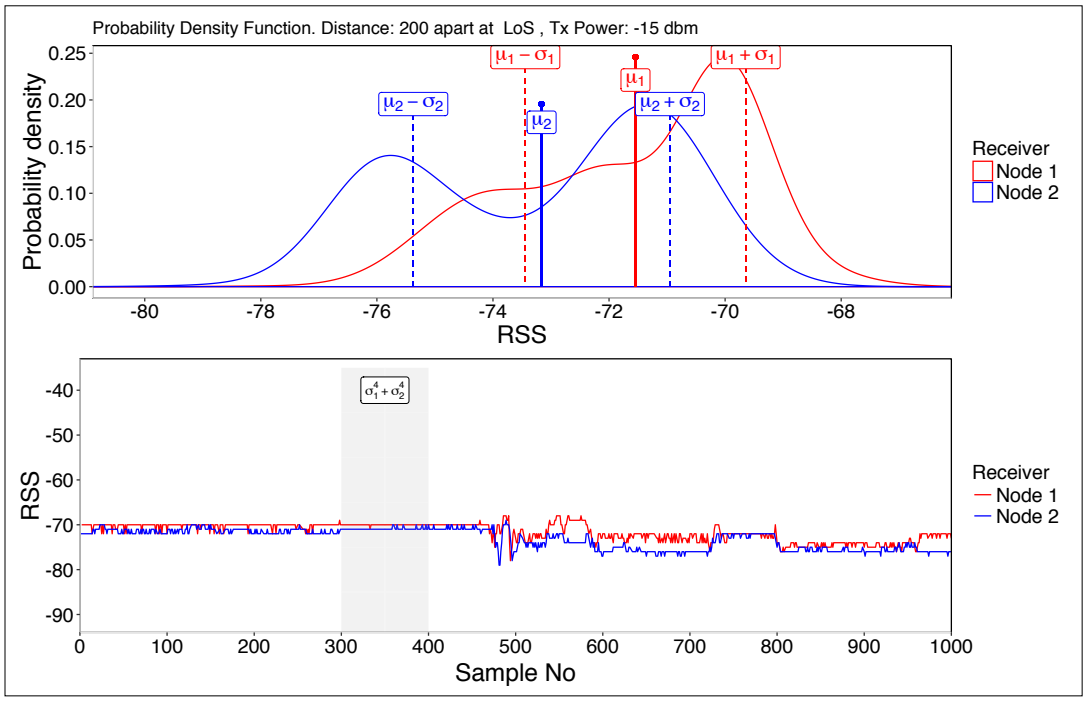


Figure 8.9: 200 cm apart nodes; reference node Radio Tx Power: -15 dBm

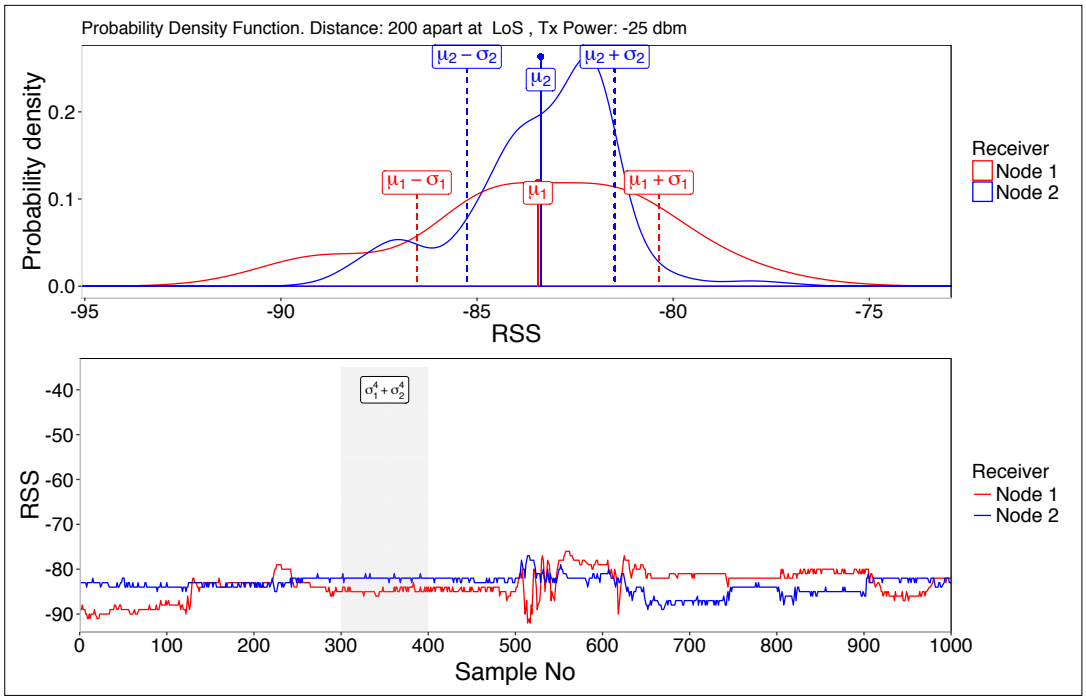


Figure 8.10: 200 cm apart nodes; reference node Radio Tx Power: -25 dBm

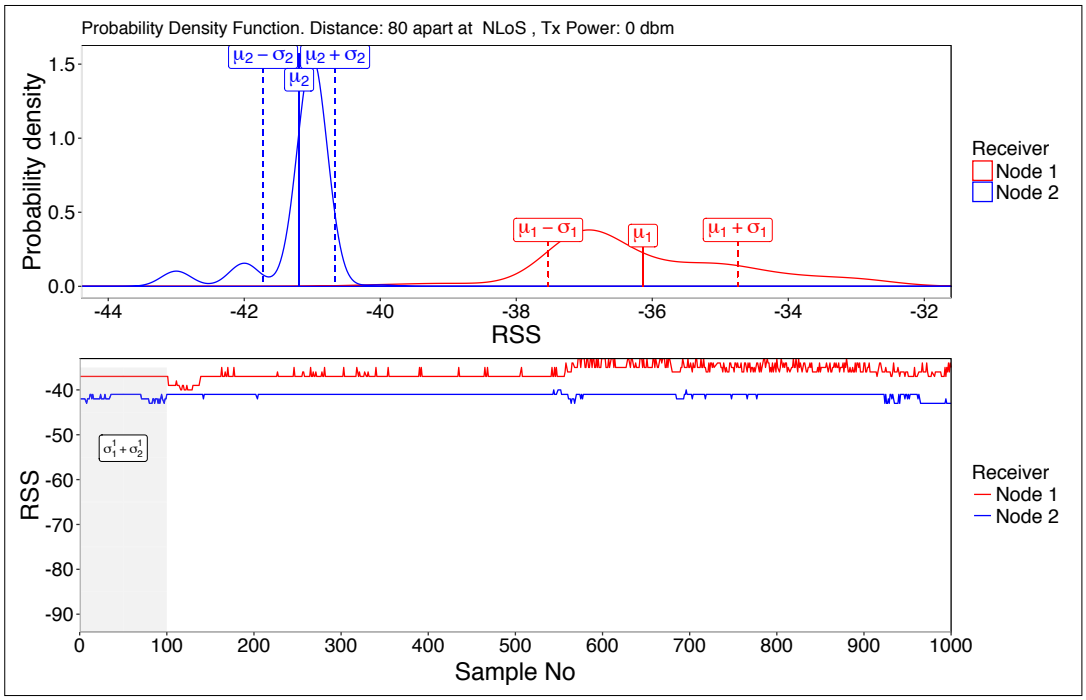


Figure 8.11: 80 cm apart nodes; reference node Radio Tx Power: 0 dBm, NLoS

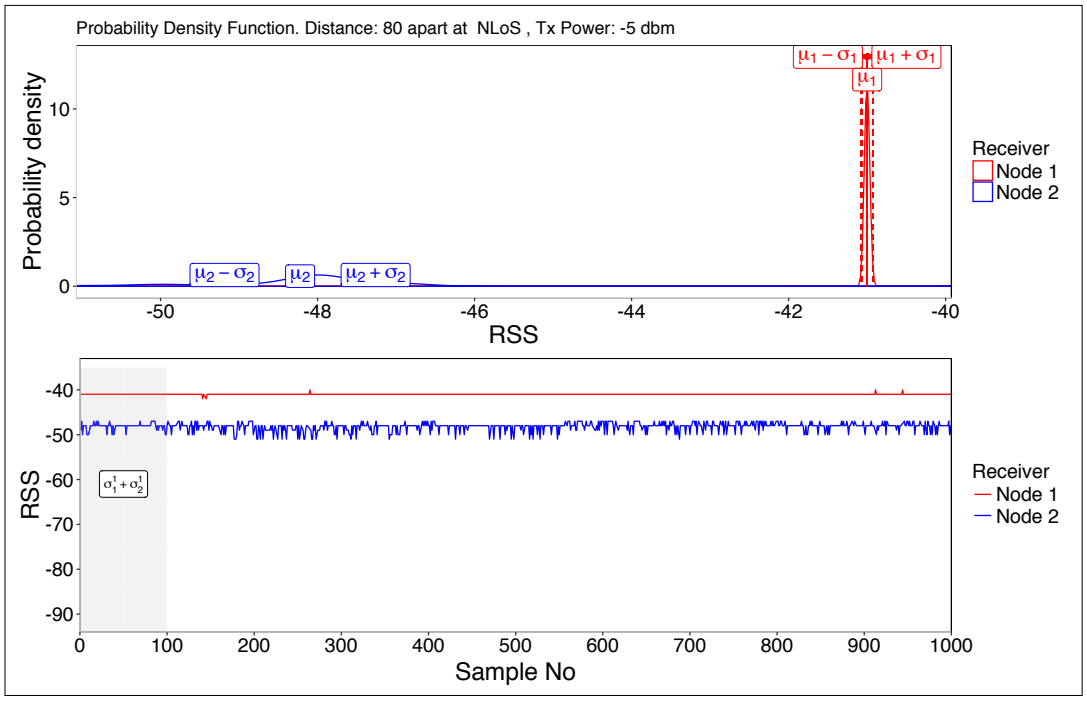


Figure 8.12: 80 cm apart nodes; reference node Radio Tx Power: -5 dBm, NLoS

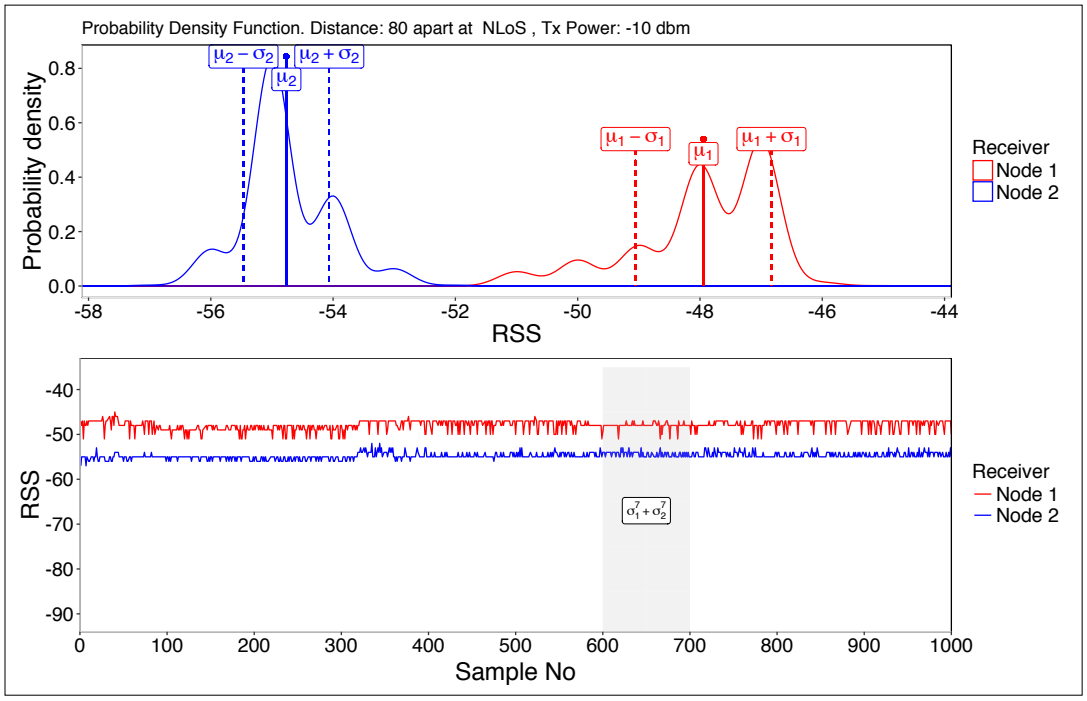


Figure 8.13: 80 cm apart nodes; reference node Radio Tx Power: -10 dBm, NLoS

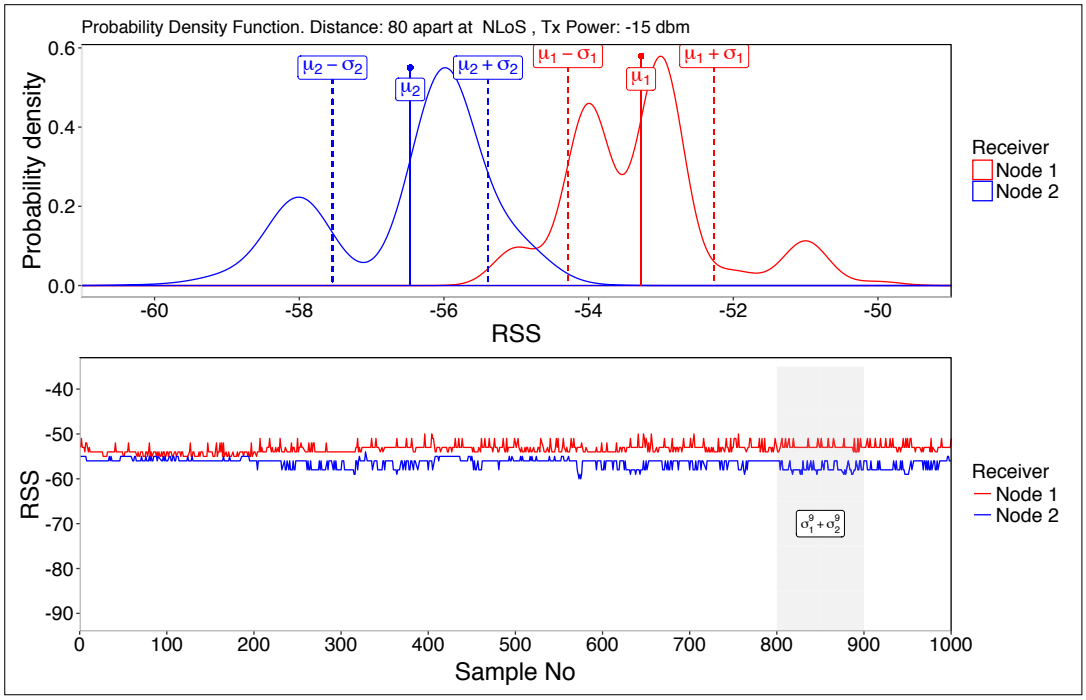


Figure 8.14: 80 cm apart nodes; reference node Radio Tx Power: -15 dBm, NLoS

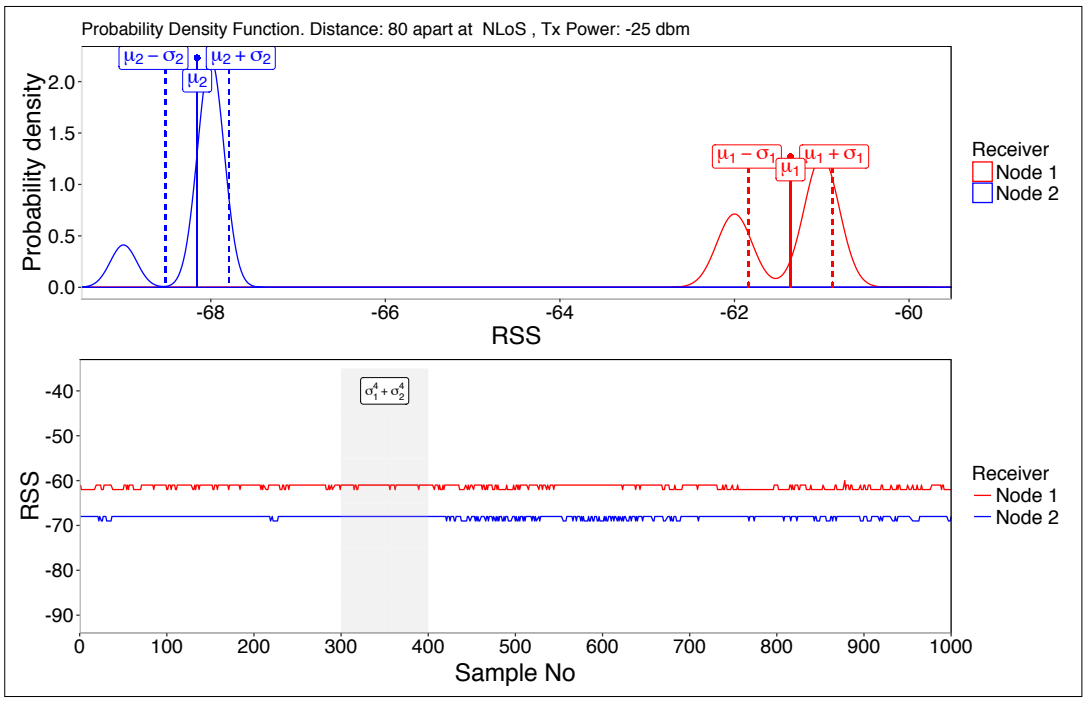


Figure 8.15: 80 cm apart nodes; reference node Radio Tx Power: -25 dBm, NLoS

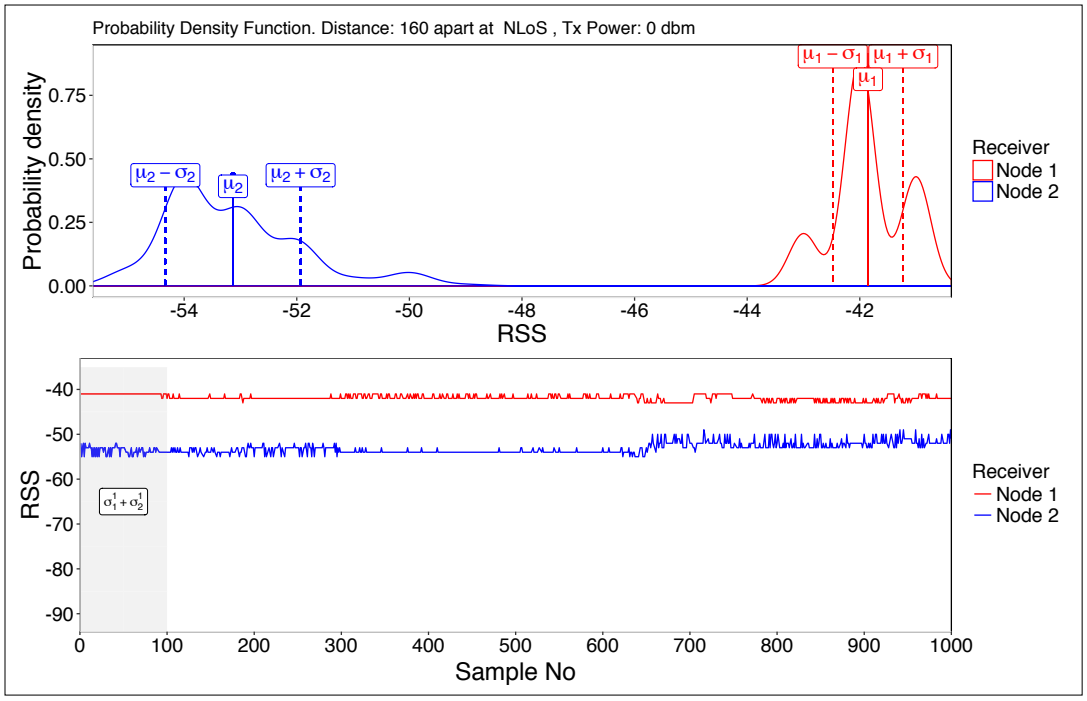


Figure 8.16: 160 cm apart nodes; reference node Radio Tx Power: 0 dBm, NLoS

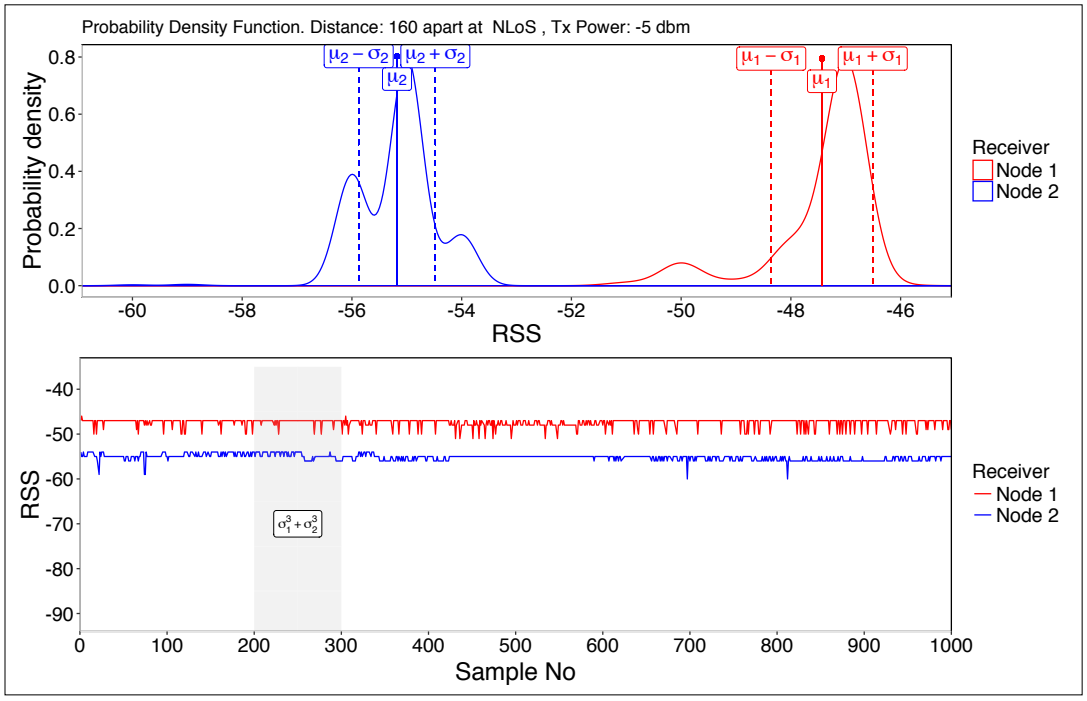


Figure 8.17: 160 cm apart nodes; reference node Radio Tx Power: -5 dBm, NLoS

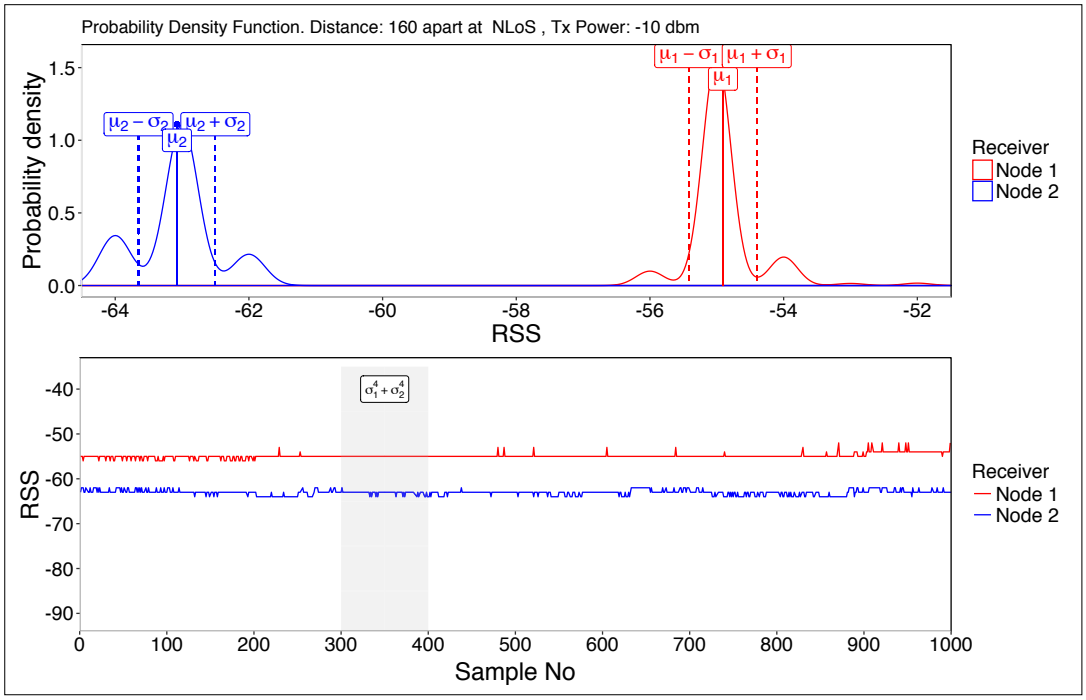


Figure 8.18: 160 cm apart nodes; reference node Radio Tx Power: -10 dBm, NLoS

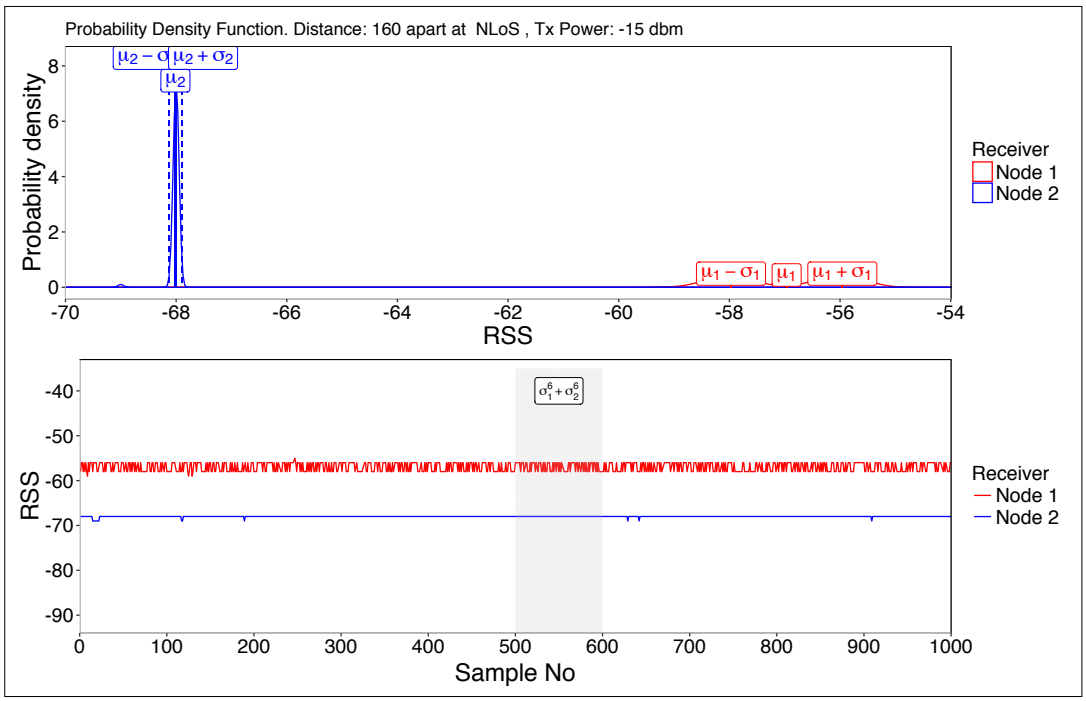


Figure 8.19: 160 cm apart nodes; reference node Radio Tx Power: -15 dBm, NLoS

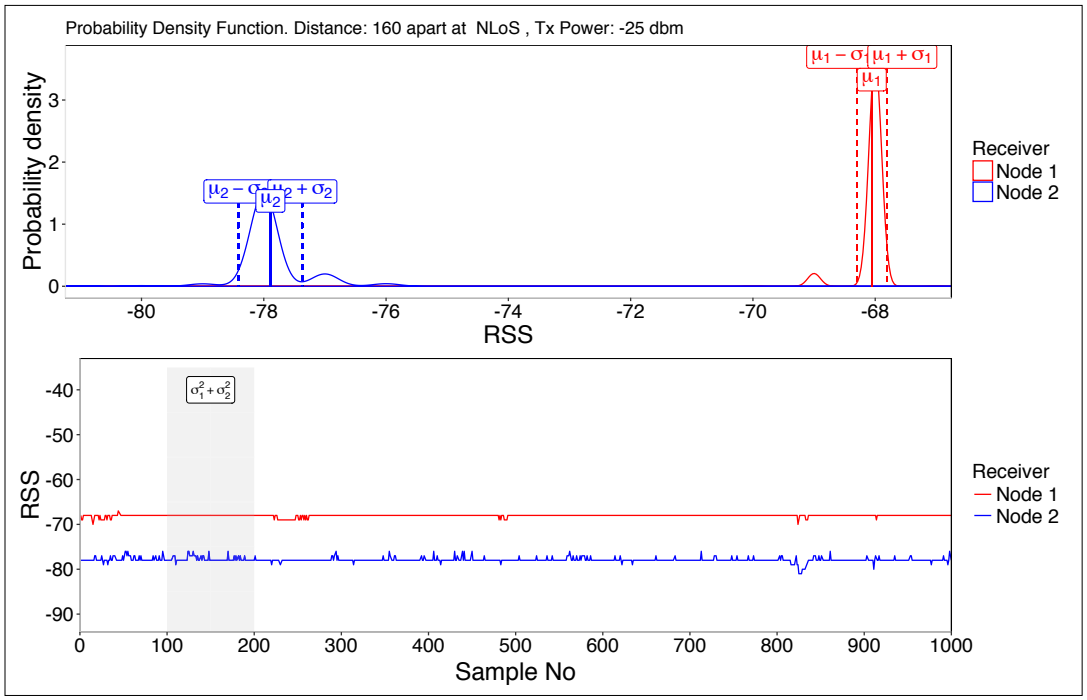


Figure 8.20: 160 cm apart nodes; reference node Radio Tx Power: -25 dBm, NLoS

Table 8.1: Line Of Sight, dS=100, dN=50

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.05	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.09	Correct
Sub-Hypothesis 2	1	0.98	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.96	Correct
Sub-Hypothesis 5	1	0.99	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.02**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.69**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.87**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.71**	**Incorrect**
Sub-Hypothesis 5	**2**	**0.90**	**Incorrect**

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.18	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.20	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Table 8.2: Line Of Sight, dS=200, dN=50

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.01**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.30**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.18**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.33**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.02**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.97**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.96**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.03**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.97**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**0.97**	**Incorrect**

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.03**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.96**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.99**	**Incorrect**
Sub-Hypothesis 5	**2**	**0.99**	**Incorrect**

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.04**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.97**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.98**	**Incorrect**
Sub-Hypothesis 5	**2**	**0.99**	**Incorrect**

Table 8.3: Line Of Sight, dS=200, dN=100

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.02	Correct
Sub-Hypothesis 2	1	0.99	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.98	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.00**	**Incorrect**
Sub-Hypothesis 2	1	0.15	Correct
Sub-Hypothesis 3	**2**	**0.25**	**Incorrect**
Sub-Hypothesis 4	1	0.06	Correct
Sub-Hypothesis 5	1	0.70	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.02	Correct
Sub-Hypothesis 2	1	0.83	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.82	Correct
Sub-Hypothesis 5	1	0.95	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.00	Correct
Sub-Hypothesis 2	1	0.28	Correct
Sub-Hypothesis 3	1	0.18	Correct
Sub-Hypothesis 4	1	0.24	Correct
Sub-Hypothesis 5	1	0.37	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.00	Correct
Sub-Hypothesis 2	1	0.20	Correct
Sub-Hypothesis 3	1	0.17	Correct
Sub-Hypothesis 4	1	0.22	Correct
Sub-Hypothesis 5	**2**	**0.12**	**Incorrect**

Table 8.4: Line Of Sight, dS=400, dN=50

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.12	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.08	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.11	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.13	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.09	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Table 8.5: Line Of Sight, dS=400, dN=100

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.18**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.09**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.99**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.99**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.05**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.24**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.13**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Table 8.6: Line Of Sight, dS=500, dN=50

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.06**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.97**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.98**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.03**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.91**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.65**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.92**	**Incorrect**
Sub-Hypothesis 5	**2**	**0.99**	**Incorrect**

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.02**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.38**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.48**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.39**	**Incorrect**
Sub-Hypothesis 5	1	0.17	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.03**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.94**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.96**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.92**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.03**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.54**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.66**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.55**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Table 8.7: Line Of Sight, dS=500, dN=100

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.06**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.08**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.11**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.13**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.08**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.97**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Table 8.8: Non Line Of Sight, dS=160, dN=80

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.06**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.65**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.65**	**Incorrect**
Sub-Hypothesis 5	**2**	**0.49**	**Incorrect**

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.23**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.28**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.30**	**Incorrect**
Sub-Hypothesis 2	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.08**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.99**	**Incorrect**
Sub-Hypothesis 3	**2**	**1.00**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.97**	**Incorrect**
Sub-Hypothesis 5	**2**	**1.00**	**Incorrect**

Table 8.9: Non Line Of Sight, dS=160, dN=160

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.01	Correct
Sub-Hypothesis 2	1	0.14	Correct
Sub-Hypothesis 3	1	0.38	Correct
Sub-Hypothesis 4	1	0.07	Correct
Sub-Hypothesis 5	1	0.87	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.03	Correct
Sub-Hypothesis 2	1	0.70	Correct
Sub-Hypothesis 3	1	0.52	Correct
Sub-Hypothesis 4	1	0.71	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	**2**	**0.03**	**Incorrect**
Sub-Hypothesis 2	**2**	**0.53**	**Incorrect**
Sub-Hypothesis 3	**2**	**0.11**	**Incorrect**
Sub-Hypothesis 4	**2**	**0.64**	**Incorrect**
Sub-Hypothesis 6	1	0.06	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.07	Correct
Sub-Hypothesis 2	1	0.94	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.90	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.04	Correct
Sub-Hypothesis 2	1	0.59	Correct
Sub-Hypothesis 3	1	0.68	Correct
Sub-Hypothesis 4	1	0.56	Correct
Sub-Hypothesis 5	1	1.00	Correct

Table 8.10: Non Line Of Sight, dS=240, dN=80

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.03	Correct
Sub-Hypothesis 2	1	0.17	Correct
Sub-Hypothesis 3	1	0.26	Correct
Sub-Hypothesis 4	1	0.06	Correct
Sub-Hypothesis 5	**2**	**0.65**	**Incorrect**

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.10	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.11	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.12	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.11	Correct
Sub-Hypothesis 2	1	1.00	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Table 8.11: Non Line Of Sight, dS=240, dN=160

Tx: 0 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.08	Correct
Sub-Hypothesis 2	1	0.96	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.98	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -5 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.08	Correct
Sub-Hypothesis 2	1	0.99	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	1.00	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -10 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.03	Correct
Sub-Hypothesis 2	1	0.73	Correct
Sub-Hypothesis 3	1	0.71	Correct
Sub-Hypothesis 4	1	0.75	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -15 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.06	Correct
Sub-Hypothesis 2	1	0.94	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.98	Correct
Sub-Hypothesis 5	1	1.00	Correct

Tx: -25 dBm

Sub-Hypotheses:	Decision	Trust Level	Result
Sub-Hypothesis 1	1	0.04	Correct
Sub-Hypothesis 2	1	0.98	Correct
Sub-Hypothesis 3	1	1.00	Correct
Sub-Hypothesis 4	1	0.98	Correct
Sub-Hypothesis 5	1	1.00	Correct

Appendix B

Raw Multichannel RSS Measurements

Here two sample datasets from raw measurements are provided. First dataset is from a measurement that could be successfully identified by the PNSD algorithm. The second dataset is from a measurement that was not perfectly identified.

Each plot represents the RSS values from one sender node and two receiver nodes that are closest to it, starting from one end of a sequence. The node with ▲ (red color top on the legend) is the closer node.

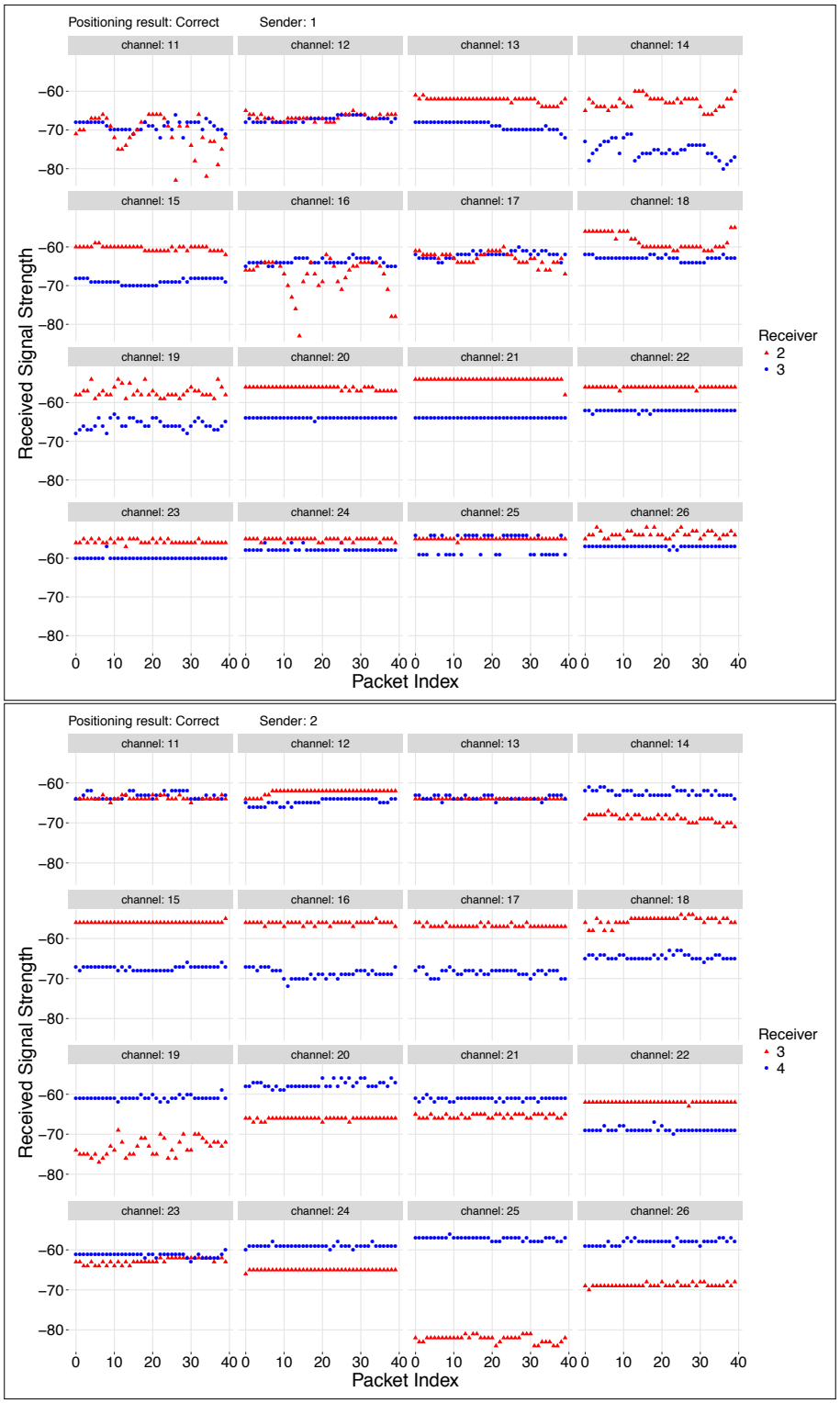


Figure 9.21: Raw measurement from a successful sequence identification. Nodes 1 & 2 in the sequence

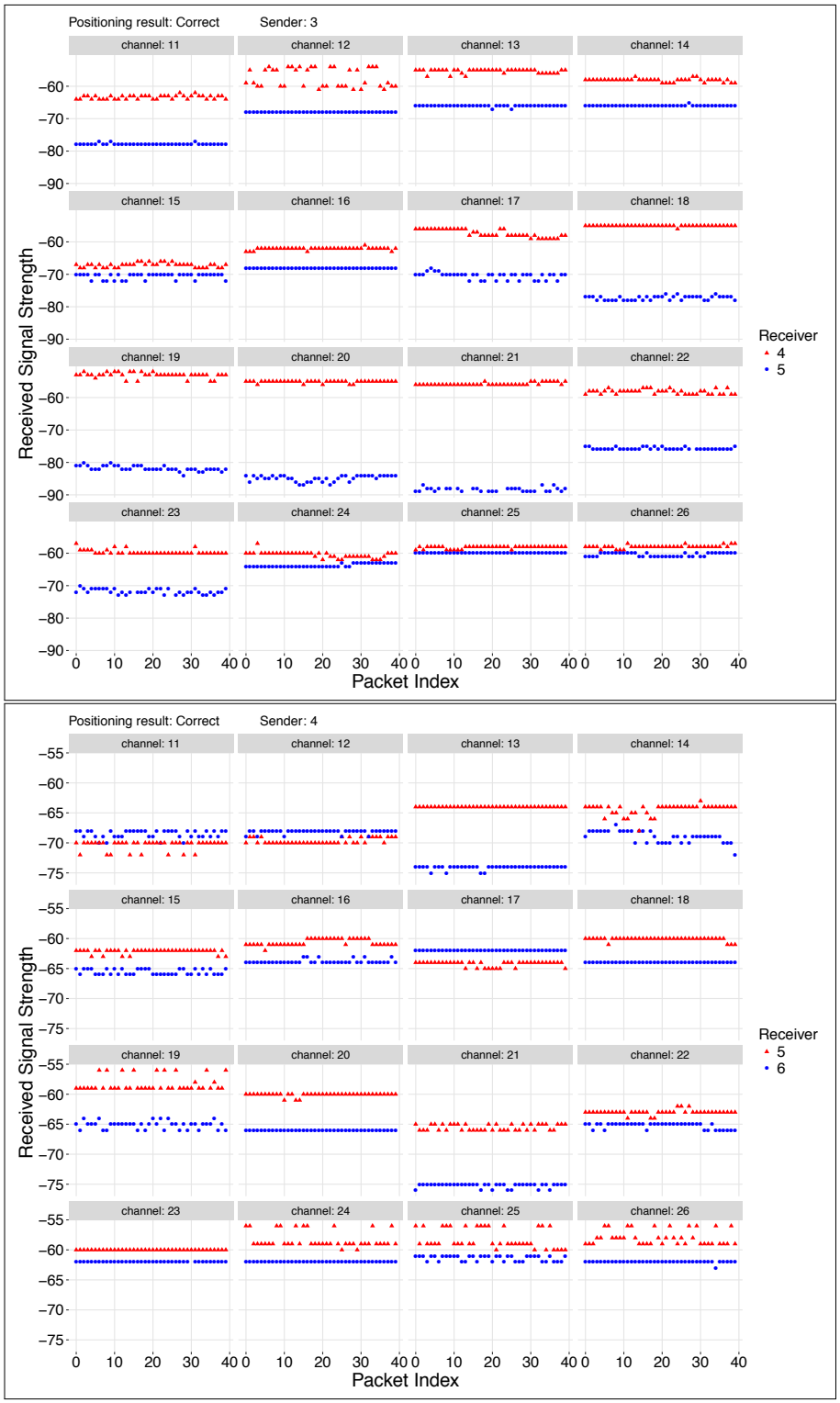


Figure 9.22: Raw measurement from a successful sequence identification. Nodes 3 & 4 in the sequence

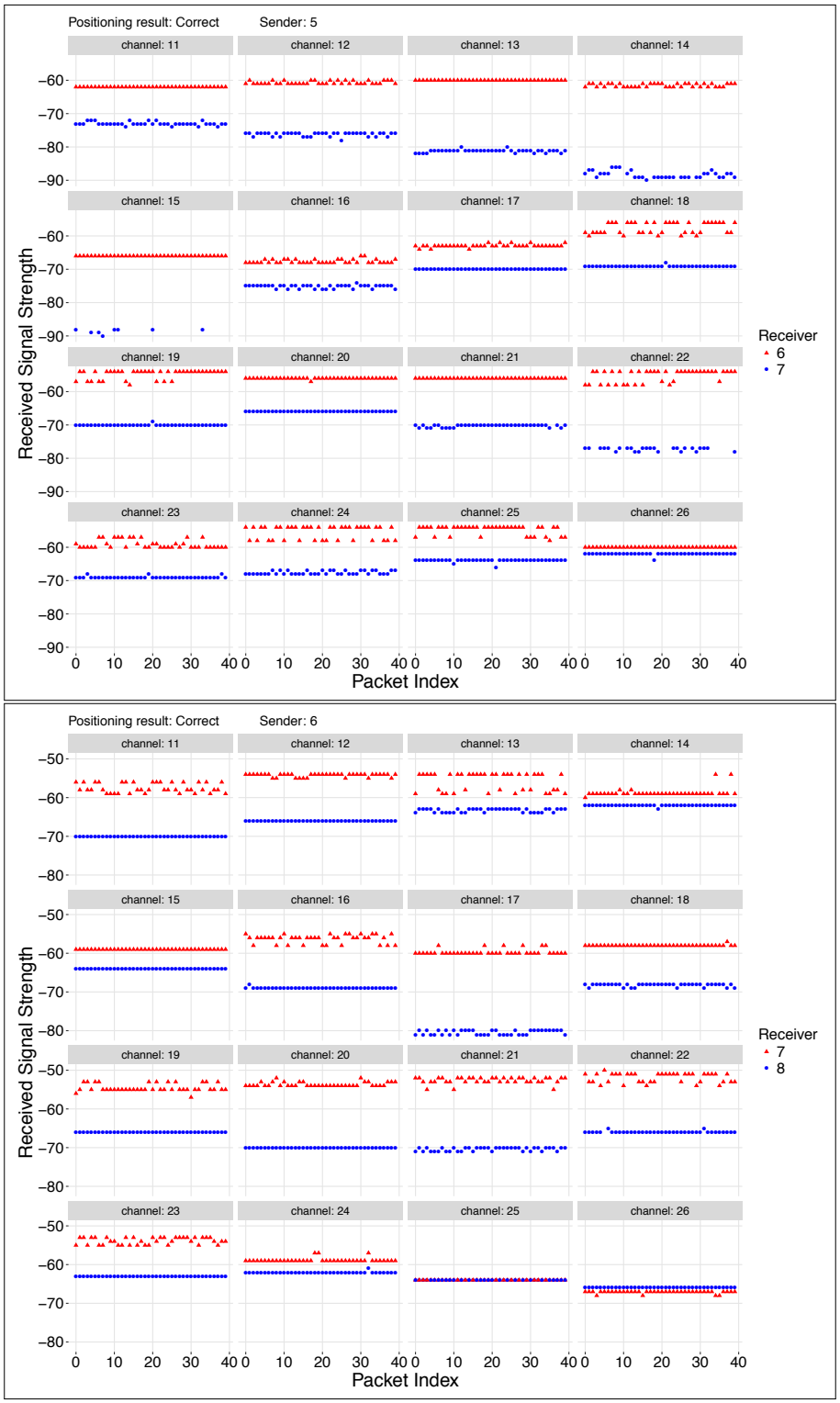


Figure 9.23: Raw measurement from a successful sequence identification. Nodes 5 & 6 in the sequence

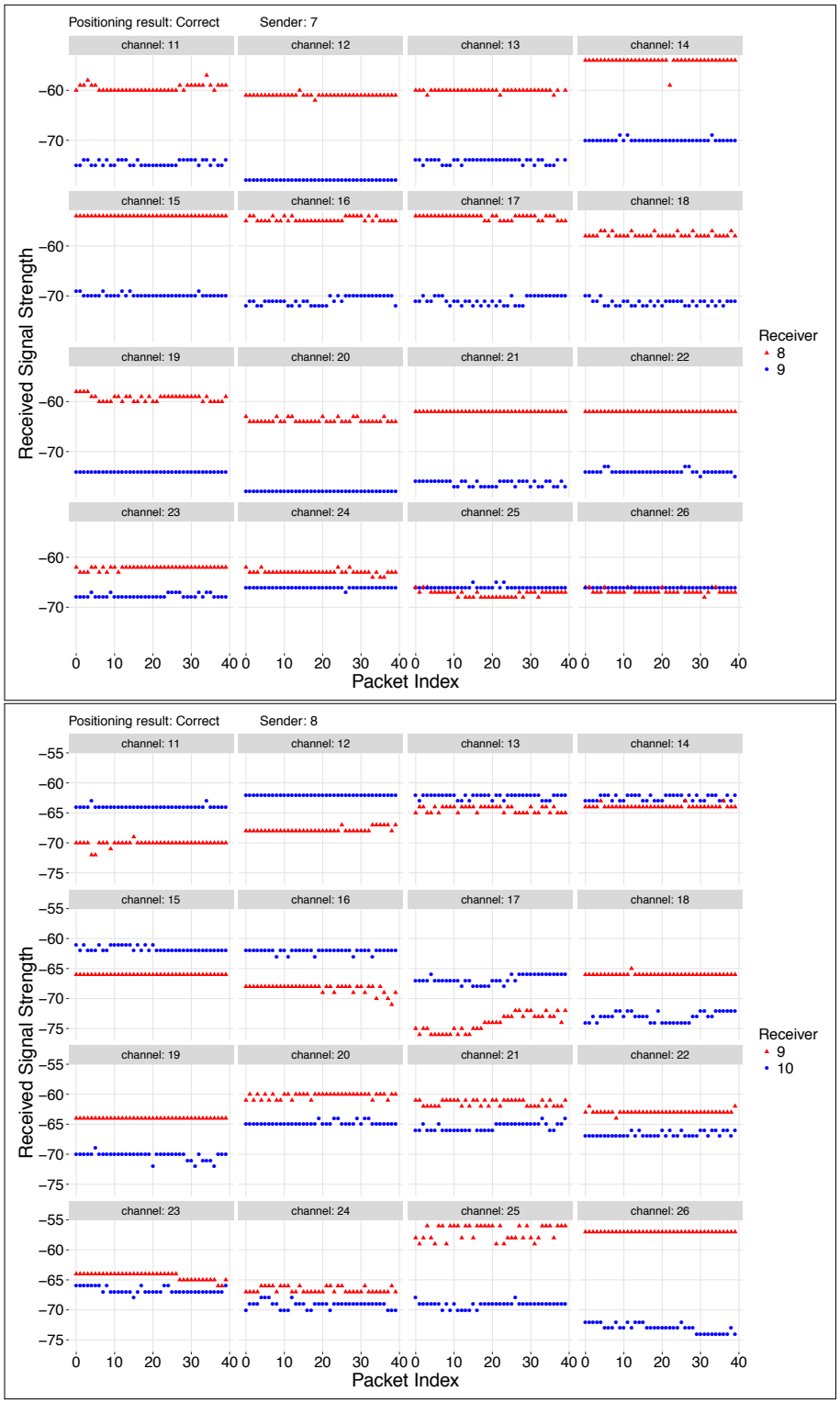


Figure 9.24: Raw measurement from a successful sequence identification. Nodes 7 & 8 in the sequence

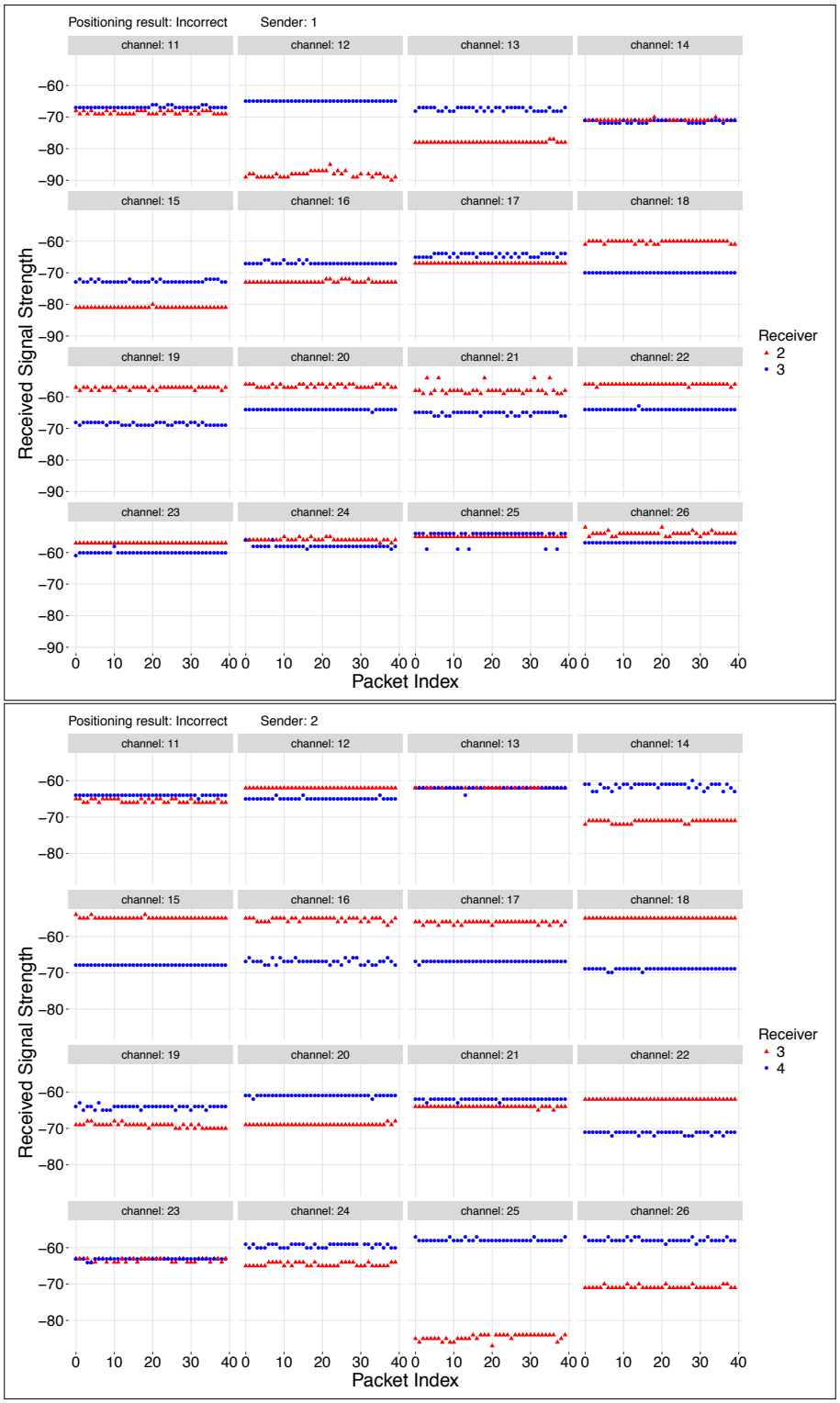


Figure 9.25: Raw measurement from an unsuccessful sequence identification. Nodes 1 & 2 in the sequence

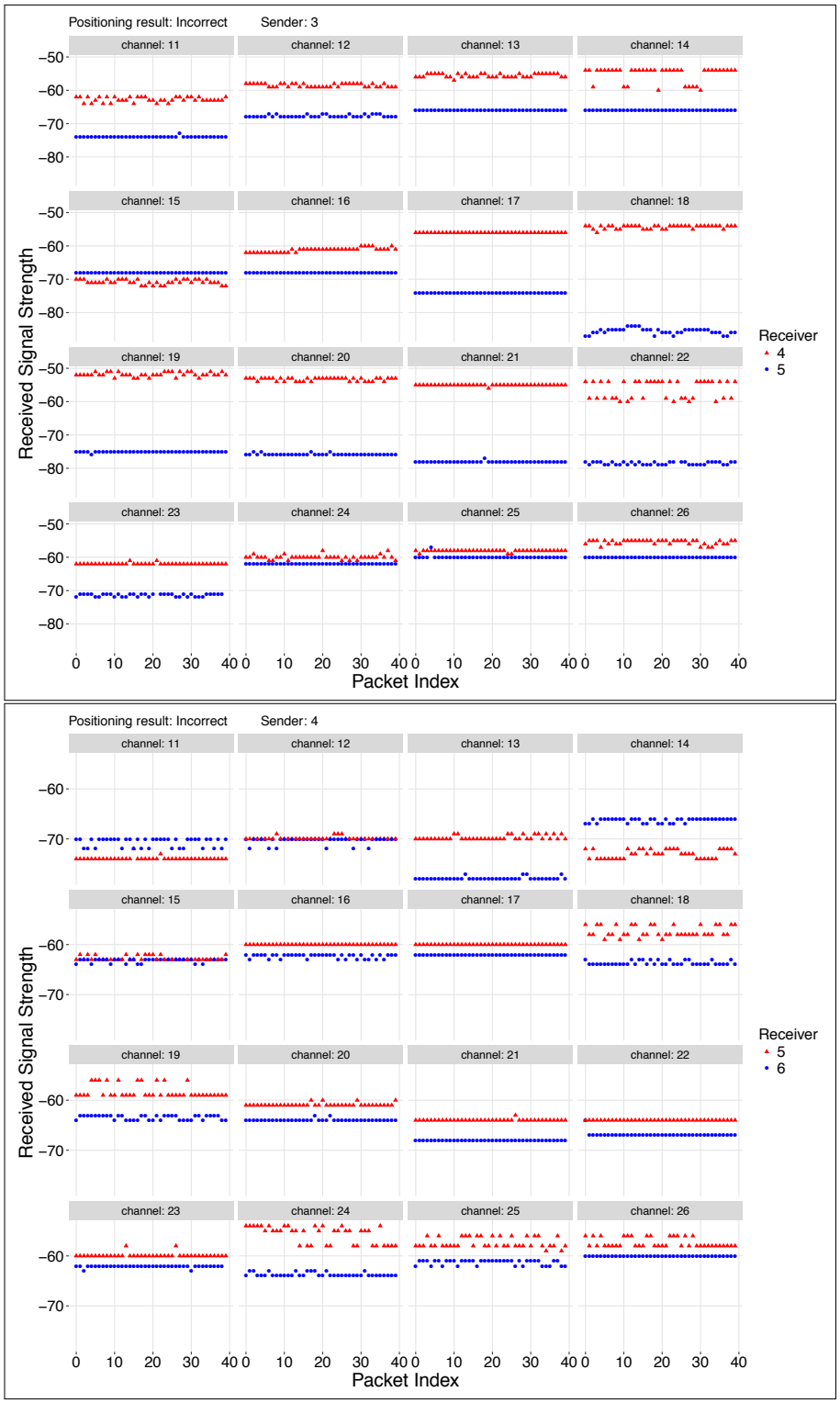


Figure 9.26: Raw measurement from an unsuccessful sequence identification. Nodes 3 & 4 in the sequence

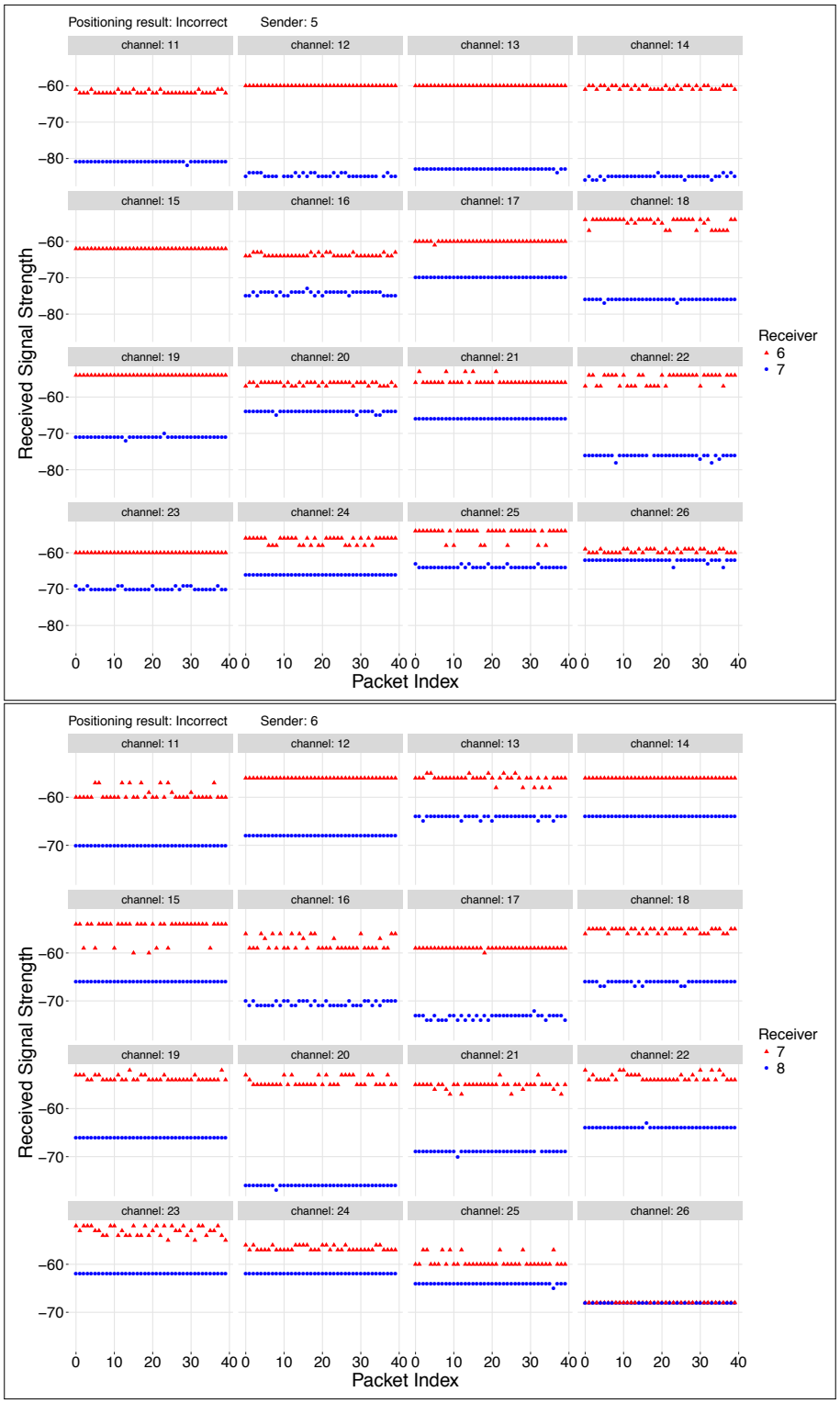


Figure 9.27: Raw measurement from an unsuccessful sequence identification. Nodes 5 & 6 in the sequence

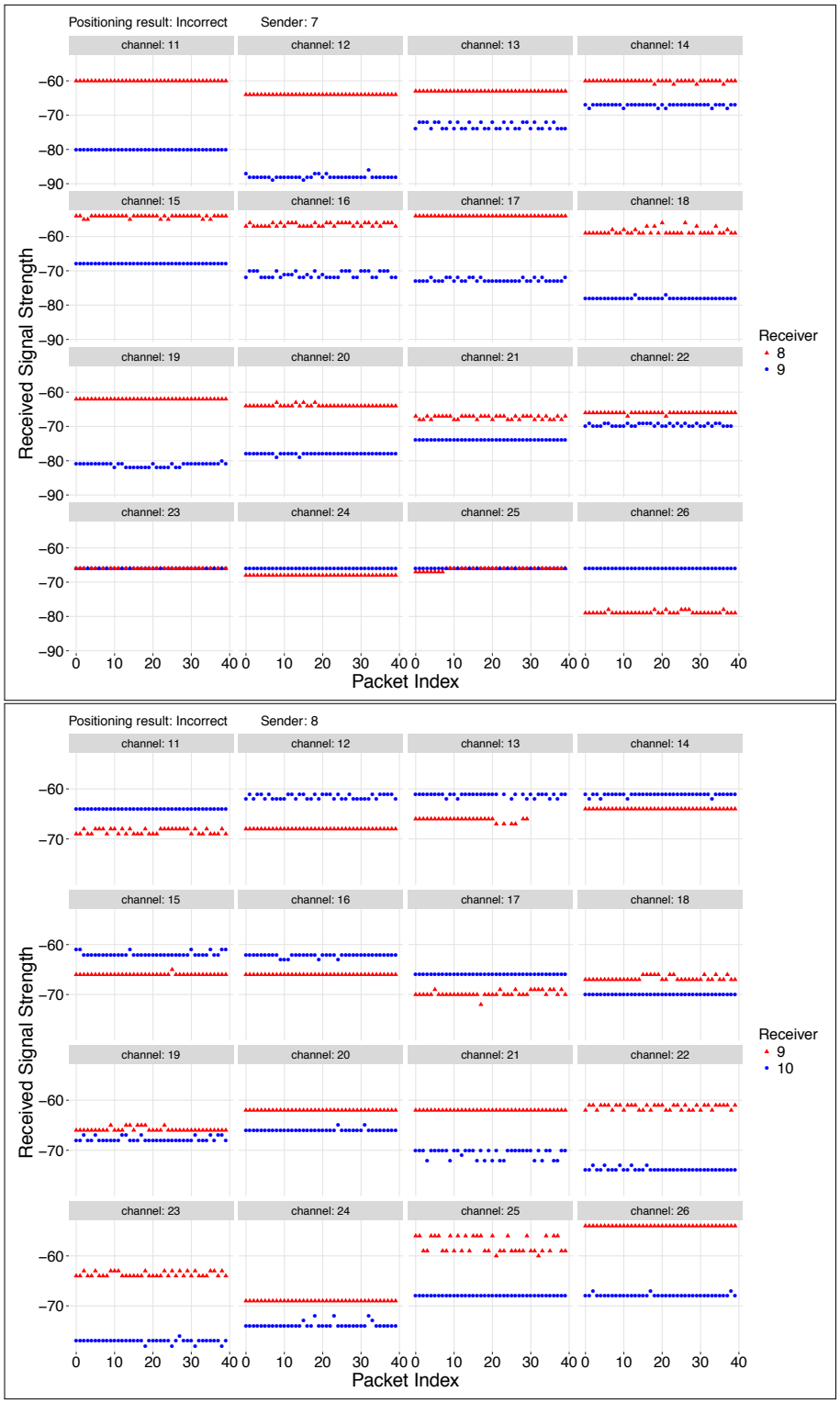


Figure 9.28: Raw measurement from an unsuccessful sequence identification. Nodes 7 & 8 in the sequence

Appendix C

Authors Publications

Thesis Related

2016 **Ergin O.**, Handziski V., Wolisz A., "**Grid-Based Node Position Discovery**", *Localization and GNSS (ICL-GNSS'16), 2016 IEEE Sixth International Conference on, 28-30 Jun 2016*

2014 **Ergin O.**, Handziski V., Behboodi A. , Wolisz A., "**Determining Node Sequence in a Linear Configuration**", Indoor Positioning and Indoor Navigation (IPIN), 2014 IEEE Fifth International Conference on, 27-30 Oct 2014

2013 **Ergin, O.**; Handziski, V.; Wolisz, A, "**Node Sequence Discovery in Wireless Sensor Networks**", Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on , vol., no., pp.394,401, 20-23 May 2013

2012 **Ergin, O.**; Wolisz, A, "**Node position discovery in wireless sensor networks**", Positioning Navigation and Communication (WPNC), 2012 9th Workshop on , vol., no., pp.157,162, 15-16 March 2012

Others

2016 Moayeri N., **Ergin O.**, Lemic F., Handziski V., Wolisz A., "**PerfLoc (Part 1): An Extensive Data Repository for Development of Smartphone Indoor Localization Apps**", *Personal, Indoor and Mobile Radio Communications - (PIMRC): Services Applications and Business, 2016 IEEE 27th International Symposium on, 4-7 Sep 2016*

2012 Marrón, P.J. et al. eds: **"The Emerging Domain of Cooperating Objects"**. Springer (2011). ISBN 978-3-642-16945-8.

2010 Sebnem Baydere, Erdal Cayirci, **Onur Ergin**, Isa Hacioglu, Anibal Ollero, Ivan Maza, Antidio Viguria, Phillipe Bonnet and Maria Lijding, "**Cooperating Embedded Systems and Wireless Sensor Networks**", Chapter 2 (pg 25-113), Wiley, ISBN 978-1-84821-000-4, 2010.

2009 P. J. Marrón, S. Karnouskos, and D. Minder, Eds.,**"Research Roadmapon Cooperating Objects"**, European Commission, Office for OfficialPublications of the European Communities, July 2009, no. ISBN: 978-92-79-12046-6.

2007 **Ergin, O.**; Baydere, S., "**ONE: Adaptive One-to-N Error Recovery in Wireless Sensor Networks**", Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on , vol., no., pp.31,36, 3-6 Dec. 2007

2007 MB. Mutluoglu, **O.Ergin**, S.Baydere, "**TAP: Tree Based Code Dissemination for Wireless Sensor Networks**", Proc. of the Int. Conf. WORLDCOM ICWN 2007, USA

2006 Sebnem Baydere, **Onur Ergin** and Embedded WiSeNts Consortium, "**Embedded WiSeNts Research Roadmap**", Logos Verlag, Berlin 2006, ISBN 3-8325-1424-4.

Bibliography

[1] "Ieee standard for low-rate wireless networks– – part 15.4," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, April 2016.

[2] Guoqiang Mao, Barış Fidan, and Brian D. O. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, July 2007.

[3] J.K. Cavers, *Mobile Channel Characteristics*, Kluwer Academic, 2000.

[4] R. Bajaj, S.L. L Ranaweera, and D.P. P Agrawal, "GPS: location-tracking technology," *Computer*, vol. 35, no. 4, pp. 92–94, apr 2002.

[5] Holger Karl and Andreas Willig, "Localization and positioning," *Protocols and Architectures for Wireless Sensor Networks*, pp. 231–249, 2005.

[6] 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, nov 2007.

[7] L. Chen, S. Thombre, K. Järvinen, E. S. Lohan, A. Alén-Savikko, H. Leppäkoski, M. Z. H. Bhuiyan, S. Bu-Pasha, G. N. Ferrara, S. Honkala, J. Lindqvist, L. Ruotsalainen, P. Korpisaari, and H. Kuusniemi, "Robustness, security and privacy in location-based services for future iot: A survey," *IEEE Access*, vol. 5, pp. 8956–8977, 2017.

[8] R. Shea, D. Fu, A. Sun, C. Cai, X. Ma, X. Fan, W. Gong, and J. Liu, "Location-based augmented reality with pervasive smartphone sensors: Inside and beyond pokemon go!," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017.

[9] D Gorecky, M Schmitt, M Loskyll, and D Zühlke, "Human-machine-interaction in the industry 4.0 era," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, jul 2014, pp. 289–294.

[10] Peter R Wurman, Raffaello D'Andrea, and Mick Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI magazine*, vol. 29, no. 1, pp. 9, 2008.

[11] S. A. Dale and P. Daly, "The soviet union's glonass navigation satellites," *IEEE Aerospace and Electronic Systems Magazine*, vol. 2, no. 5, pp. 13–17, May 1987.

[12] J Benedicto, S Dinwiddy, G Gatti, R Lucas, and M Lugert, "Galileo: Satellite system design," *European Space Agency*, 2000.

[13] Shaofeng Bian, Jihang Jin, and Zhaobao Fang, "The beidou satellite positioning system and its positioning accuracy," *Navigation*, vol. 52, no. 3, pp. 123–129, 2005.

[14] Ferdinando Urbano, Francesca Cagnacci, Clément Calenge, Holger Dettki, Alison Cameron, and Markus Neteler, "Wildlife tracking data management: a new vision," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 365, no. 1550, pp. 2177 LP – 2185, jun 2010.

[15] Rainer Mautz, "The challenges of indoor environments and specification on some alternative positioning systems," *Proceedings - 6th Workshop on Positioning, Navigation and Communication, WPNC 2009*, vol. 2009, pp. 29–36, 2009.

[16] Zhengjie Wang, Xiaoguang Zhao, and Xu Qian, "The application and issuse of linear wireless sensor networks," in *2011 International Conference on System science, Engineering design and Manufacturing informatization*, Oct 2011, vol. 2, pp. 9–12.

[17] Imad Jawhar, Nader Mohamed, and Dharma P. Agrawal, "Linear wireless sensor networks: Classification and applications," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1671 – 1682, 2011, Dependable Multimedia Communications: Systems, Services, and Applications.

[18] I. Jawhar, M. Ammar, S. Zhang, J. Wu, and N. Mohamed, "Ferry-based linear wireless sensor networks," in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec 2013, pp. 304–309.

[19] A. Baechler, L. Baechler, S. Autenrieth, P. Kurtz, T. Hoerz, T. Heidenreich, and G. Kruell, "A comparative study of an assistance system for manual order picking – called pick-by-projection – with the guiding systems pick-by-paper, pick-by-light and pick-by-display," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, Jan 2016, pp. 523–531.

[20] S. Lanzisera, D. Zats, and K. S. J. Pister, "Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 837–845, March 2011.

[21] KSAP Levis, "Rssi is under appreciated," in *Proceedings of the Third Workshop on Embedded Networked Sensors, Cambridge, MA, USA*, 2006, vol. 3031, p. 239242.

[22] K. Heurtefeux and F. Valois, "Is rssi a good choice for localization in wireless sensor network?," in *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, March 2012, pp. 732–739.

[23] D. Konings, N. Faulkner, F. Alam, F. Noble, and E. Lai, "Do rssi values reliably map to rss in a localization system?," in *2017 2nd Workshop on Recent Trends in Telecommunications Research (RTTR)*, Feb 2017, pp. 1–5.

[24] A Zanella and A Bardella, "Rss-based ranging by multichannel rss averaging," *Wireless Communications Letters, IEEE*, vol. 3, no. 1, pp. 10–13, February 2014.

[25] A. Zanella, "Best practice in rss measurements and ranging," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2662–2686, Fourthquarter 2016.

[26] Ambili Thottam Parameswaran, Mohammad Iftekhar Husain, and Shambhu Upadhyaya, "Is rssi a reliable parameter in sensor localization algorithms ? an experimental study," in *28th International Symposiym on Reliable Distributed Systems*, 2009.

[27] Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Souvik Sen, and Vlado Handziski, "Microsoft indoor localization competition: Experiences and lessons learned," *GetMobile: Mobile Comp. and Comm.*, vol. 18, no. 4, pp. 24–31, Jan. 2015.

[28] D. Lymberopoulos and J. Liu, "The microsoft indoor localization competition: Experiences and lessons learned," *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 125–140, Sept 2017.

[29] Vlado Handziski, Andreas Köpke, Andreas Willig, and Adam Wolisz, "Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks," in *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, 2006, REALMAN '06.

[30] Holger Karl and Andreas Willig, *Protocols and Architectures for Wireless Sensor Networks*, John Wiley & Sons, 1st edition, 2005.

[31] Michel Banatre, Pedro Jose Marron, Anibal Ollero, Adam Wolisz, Michel Banatre, Pedro Jose Marron, Anibal Ollero, and Adam Wolisz, *Cooperating Embedded Systems and Wireless Sensor Networks*, Wiley-IEEE Press, 1 edition, 2008.

[32] Vidyasagar Potdar, Atif Sharif, and Elizabeth Chang, "Wireless sensor networks: A survey," in *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*. IEEE, 2009, pp. 636–641.

[33] A. Berger, L. B. Hörmann, C. Leitner, S. B. Oswald, P. Priller, and A. Springer, "Sustainable energy harvesting for robust wireless sensor networks in industrial applications," in *2015 IEEE Sensors Applications Symposium (SAS)*, April 2015, pp. 1–6.

[34] B. A. Nahas, S. Duquennoy, V. Iyer, and T. Voigt, "Low-power listening goes multichannel," in *2014 IEEE International Conference on Distributed Computing in Sensor Systems*, May 2014, pp. 2–9.

[35] P.J.M. Havinga, L. Evers, J. Wu, Holger Karl, Andreas Koepke, Vlado Handziski, and M. Zorzi, "Snapshots of the eyes project," in *Wireless Ad-Hoc Networks, 2004 International Workshop on*, May 2004, pp. 45–52.

[36] Jan Beutel, Oliver Kasten, Friedemann Mattern, Kay Römer, Frank Siegemund, and Lothar Thiele, "Prototyping wireless sensor network applications with btnodes," *Wireless Sensor Networks*, pp. 323–338, 2004.

[37] Jochen Schiller, Achim Liers, and Hartmut Ritter, "Scatterweb: A wireless sensornet platform for research and teaching," *Computer Communications*, vol. 28, no. 13, pp. 1545–1551, 2005.

[38] Lama Nachman, Jonathan Huang, Junaith Shahabdeen, Robert Adler, and Ralph Kling, "Imote2: Serious computation at the edge," in *Wireless Communications and Mobile Computing Conference, 2008. IWCMC'08. International*. IEEE, 2008, pp. 1118–1123.

[39] Brendan Cody-Kenny, David Guerin, Desmond Ennis, Ricardo Simon Carbajo, Meriel Huggard, and Ciaran Mc Goldrick, "Performance evaluation of the 6lowpan protocol on micaz and telosb motes," in *Proceedings of the 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*. ACM, 2009, pp. 25–30.

[40] Toshiaki Miyazaki, Shoichi Yamaguchi, Koji Kobayashi, Junji Kitamichi, Song Guo, Tsuneo Tsukahara, and Takafumi Hayashi, "A software defined wireless sensor network," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE, 2014, pp. 847–852.

[41] Jonny Johansson, Matthias Völker, Jens Eliasson, Åke Östmark, Per Lindgren, and Jerker Delsing, "Mulle: a minimal sensor networking device: implementation and manufacturing challenges," in *IMAPS Nordic Annual Conference: 26/09/2004-28/09/2004*. International Microelectronics and Packaging Society, Nordic chapter, 2004, pp. 265–271.

[42] Geoffrey Werner-Allen, Konrad Lorincz, Mario Ruiz, Omar Marcillo, Jeff Johnson, Jonathan Lees, and Matt Welsh, "Deploying a wireless sensor network on an active volcano," *IEEE internet computing*, vol. 10, no. 2, pp. 18–25, 2006.

[43] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, April 2005, pp. 463–468.

[44] Min Goo Lee, Yong Kuk Park, Kyung Kwon Jung, and June Jae Yoo, "Wireless electricity monitoring system for smart house using smart plug," *Trans Tech Publications, USA*, 2011.

[45] Adrian Burns, Barry R Greene, Michael J McGrath, Terrance J O'Shea, Benjamin Kuris, Steven M Ayer, Florin Stroiescu, and Victor Cionca, "Shimmer–a wireless sensor platform for noninvasive biomedical research," *IEEE Sensors Journal*, vol. 10, no. 9, pp. 1527–1534, 2010.

[46] "Zolertia RE-Mote platform," online, https://github.com/Zolertia/Resources/wiki/RE-Mote Last accessed: 2017-09-17.

[47] "TI Multi-standard SensorTag," online, http://www.ti.com/product/CC2650 and http://www.ti.com/ww/en/wireless_connectivity/sensortag/tearDown.html Last accessed: 2017-09-17.

[48] "The SEED-EYE Board," online, http://www.evidence.eu.com/products/seed-eye.html Last accessed: 2017-09-17.

[49] "nRF52 DK," online, https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF52-DK Last accessed: 2017-09-17.

[50] TinyOS Alliance, "Tinyos 2.1 adding threads and memory protection to tinyos," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, New York, NY, USA, 2008, SenSys '08, pp. 413–414, ACM.

[51] Adam Dunkels, BjÃ¶rn GrÃ¶nvall, and Thiemo Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the First IEEE Workshop on Embedded Networked Sensors (Emnets-I)*, Tampa, Florida, USA, Nov. 2004.

[52] Chih-Chun Chang, Sead Muftic, and David J Nagel, "Measurement of energy costs of security in wireless sensor nodes," in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*. IEEE, 2007, pp. 95–102.

[53] Rahul Mangharam, Anthony Rowe, and Raj Rajkumar, "Firefly: a cross-layer platform for real-time embedded wireless networks," *Real-Time Systems*, vol. 37, no. 3, pp. 183–231, 2007.

[54] TGH Basten and FBA Schiphorst, *Structural health monitoring with a wireless vibration sensor network*, Katholieke Universiteit Leuven, 2012.

[55] Pavan Sikka, Peter Corke, Leslie Overs, Philip Valencia, and Tim Wark, "Fleck-a platform for real-world outdoor sensor networks," in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*. IEEE, 2007, pp. 709–714.

[56] Christian Decker, Albert Krohn, Michael Beigl, and Tobias Zimmer, "The particle computer system," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*. IEEE, 2005, pp. 443–448.

[57] Anupama Sahu, Eduardo B Fernandez, Mihaela Cardei, and Michael VanHilst, "A pattern for a sensor node," in *Proceedings of the 17th conference on pattern languages of programs*. ACM, 2010, p. 7.

[58] Aliaksei Andrushevich, Rolf Kistler, Marcel Bieri, and Alexander Klapproth, "Zigbee/ieee 802.15. 4 technologies in ambient assisted living applications," in *3rd European ZigBee Developers? Conference (EuZDC)*, 2009.

[59] Vladimir Vujovic and Mirjana Maksimovic, "Raspberry pi as a wireless sensor node: performances and constraints," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE, 2014, pp. 1013–1018.

[60] "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.1a: Wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (wpan)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–700, June 2005.

[61] "Ieee standard for high data rate wireless multi-media networks– – part 15.3," *IEEE Std 802.15.3-2016 (Revision of IEEE Std 802.15.3-2003*, pp. 1–510, July 2016.

[62] "Ieee standard for local and metropolitan area networks - part 15.6: Wireless body area networks," *IEEE Std 802.15.6-2012*, pp. 1–271, Feb 2012.

[63] "Chipcon CC2420 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver," online, http://www.ti.com/lit/ds/symlink/cc2420.pdf Last accessed: 2017-06-25.

[64] V. Handziski, J. Polastre, J.-H. Hauer, C. Sharp, A. Wolisz, and D. Culler, "Flexible hardware abstraction for wireless sensor networks," in *Proceedings of Second European Workshop on Wireless Sensor Networks (EWSN 2005)*, Istanbul, Turkey, Feb. 2005.

[65] Mark L Davison, Cody S Ding, and Se-Kang Kim, "Multidimensional scaling," in *The Reviewer's Guide to Quantitative Methods in the Social Sciences*, chapter 20, p. 265. Routledge, 2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN, 2010.

[66] Warren S Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[67] "Psychometrika," (1936 - today), https://link.springer.com/journal/11336 (Last accessed: 2017-06-25).

[68] Jan de Leeuw and Patrick Mair, "Multidimensional scaling using majorization: Smacof in r," *Journal of Statistical Software*, vol. 31, no. 3, pp. 1–30, 8 2009.

[69] Publications Office of the European Union, "Gnss market report issue 5," Tech. Rep., The European GNSS Agency (GSA), 2017, https://www.gsa.europa.eu/system/files/reports/gnss_mr_2017.pdf, Last accessed: 24.09.2017.

[70] Xiang-Yang Li, *Wireless Ad Hoc and Sensor Networks: Theory and Applications*, Cambridge University Press, New York, NY, USA, 2008.

[71] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1327–1346, Secondquarter 2017.

[72] Paweł Kułakowski, Javier Vales-Alonso, Esteban Egea-López, Wiesław Ludwin, and Joan García-Haro, "Angle-of-arrival localization based on antenna arrays for wireless sensor networks," *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1181–1186, 2010.

[73] D. Koks, "Numerical calculations for passive geolocation scenarios," Tech. Rep. DSTO-RR-0319, Department of Defence, Australian Government, January 2007.

[74] T. S. Rappaport, J. H. Reed, and B. D. Woerner, "Position location using wireless communications on highways of the future," *IEEE Communications Magazine*, vol. 34, no. 10, pp. 33–41, Oct 1996.

[75] Yang Sun Lee, Jang Woo Park, and Leonard Barolli, "A localization algorithm based on aoa for ad-hoc sensor networks," *Mob. Inf. Syst.*, vol. 8, no. 1, pp. 61–72, Jan. 2012.

[76] Jehoshua Bruck, Jie Gao, and Anxiao (Andrew) Jiang, "Localization and routing in sensor networks by local angle information," *ACM Trans. Sen. Netw.*, vol. 5, no. 1, pp. 7:1–7:31, Feb. 2009.

[77] S. Lanzisera, D. T. Lin, and K. S. J. Pister, "Rf time of flight ranging for wireless sensor network localization," in *2006 International Workshop on Intelligent Solutions in Embedded Systems*, June 2006, pp. 1–12.

[78] "Nanotron Technologies," http://www.nanotron.com, Last accessed: 2017-01-22.

[79] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2000, MobiCom '00, pp. 32–43, ACM.

[80] Andreas Savvides, Heemin Park, and Mani B. Srivastava, "The n-hop multilateration primitive for node localization problems," *Mob. Netw. Appl.*, vol. 8, no. 4, pp. 443–451, Aug. 2003.

[81] S. Gezici, Zhi Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, July 2005.

[82] Kay Römer, "The lighthouse location system for smart dust," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, New York, NY, USA, 2003, MobiSys '03, pp. 15–30, ACM.

[83] Theodore Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[84] J. J. Robles, M. Deicke, and R. Lehnert, "3d fingerprint-based localization for wireless sensor networks," in *2010 7th Workshop on Positioning, Navigation and Communication*, March 2010, pp. 77–85.

[85] L. Gogolak, S. Pletl, and D. Kukolj, "Indoor fingerprint localization in wsn environment based on neural network," in *2011 IEEE 9th International Symposium on Intelligent Systems and Informatics*, Sept 2011, pp. 293–296.

[86] R. Priwgharm and P. Chemtanomwong, "A comparative study on indoor localization based on rssi measurement in wireless sensor network," in *2011 Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE)*, May 2011, pp. 1–6.

[87] Rong Peng and Mihail L. Sichitiu, "Probabilistic localization for outdoor wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, 2007.

[88] M. Garcia, C. Martinez, J. Tomas, and J. Lloret, "Wireless sensors self-location in an indoor wlan environment," in *2007 International Conference on Sensor Technologies and Applications (SENSORCOMM 2007)*, Oct 2007, pp. 146–151.

[89] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 961–974, Nov. 2004.

[90] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, Oct 2000.

[91] D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, 2001, vol. 5, pp. 2926–2931 vol.5.

[92] Y. Wang, X. Wang, D. Wang, and D. P. Agrawal, "Range-free localization using expected hop progress in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1540–1552, Oct 2009.

[93] N. Moayeri, M. O. Ergin, F. Lemic, V. Handziski, and A. Wolisz, "Perfloc (part 1): An extensive data repository for development of smartphone indoor localization apps," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2016, pp. 1–7.

[94] "Microsoft indoor localization competition," http://research.microsoft.com/en-us/events/ipsn2014indoorlocalizatinocompetition/, Last accessed: 2017-10-03.

[95] Kamin Whitehouse, Chris Karlof, and David Culler, "A practical evaluation of radio signal strength for ranging-based localization," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 1, pp. 41–52, Jan. 2007.

[96] J. Kang, D. Kim, and Y. Kim, "Rss self-calibration protocol for wsn localization," in *2007 2nd International Symposium on Wireless Pervasive Computing*, Feb 2007.

[97] Peter De Cauwer, Tim Van Overtveldt, Jeroen Doggen, Filip Van der Schueren, Maarten Weyn, and Jerry Bracke, "Study of rss-based localisation methods in wireless sensor networks," in *ECUMICT, European Conference on the Use of Modern Information and Communication Technologies*. January, 2010.

[98] Giovanni Zanca, Francesco Zorzi, Andrea Zanella, and Michele Zorzi, "Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks," in *Proceedings of the Workshop on Real-world Wireless Sensor Networks*, New York, NY, USA, 2008, REALWSN '08, pp. 1–5, ACM.

[99] K. Yu, I. Sharp, and Y.J. Guo, *Ground-based wireless positioning*, vol. 5, Wiley-IEEE Press, 2009.

[100] P. Bahl and V.N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, vol. 2.

[101] Giovanni Zanca, Francesco Zorzi, Andrea Zanella, and Michele Zorzi, "Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks," in *Proceedings of the workshop on Real-world wireless sensor networks*, 2008, REALWSN '08.

[102] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, MobiCom '03.

[103] Dian Zhang, Yunhuai Liu, Xiaonan Guo, Min Gao, and L.M. Ni, "On distinguishing the multiple radio paths in rss-based ranging," in *INFOCOM, 2012 Proceedings IEEE*, Mar. 2012.

[104] P. Pettinato, N. Wirström, J. Eriksson, and T. Voigt, "Multi-channel two-way time of flight sensor network ranging," in *Proceedings of the 9th European Conference on Wireless Sensor Networks*, Feb. 2012, EWSN 2012.

[105] Andrea Bardella, Nicola Bui, Andrea Zanella, and Michele Zorzi, "An experimental study on ieee 802.15.4 multichannel transmission to improve rssi-based service performance," in *Proceedings of the 4th international conference on Real-world wireless sensor networks*, 2010, REALWSN'10.

[106] I. Jawhar and N. Mohamed, "A hierarchical and topological classification of linear sensor networks," in *2009 Wireless Telecommunications Symposium*, April 2009, pp. 1–8.

[107] S. Varshney, C. Kumar, and A. Swaroop, "Linear sensor networks: Applications, issues and major research trends," in *International Conference on Computing, Communication Automation*, May 2015, pp. 446–451.

[108] R. D. Komguem, R. Stanica, M. Tchuente, and F. Valois, "Node ranking in wireless sensor networks with linear topology," in *2017 Wireless Days*, March 2017, pp. 127–132.

[109] X. Zhu and G. Chen, "Spatial ordering derivation for one-dimensional wireless sensor networks," in *2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications*, May 2011, pp. 171–176.

[110] Xiaojun Zhu, Xiaobing Wu, and Guihai Chen, "Relative localization for wireless sensor networks with linear topology," *Computer Communications*, vol. 36, no. 15-16, pp. 1581 – 1591, 2013.

[111] Yong Cui, Qiusheng Wang, Haiwen Yuan, Xiao Song, Xuemin Hu, and Luxing Zhao, "Relative localization in wireless sensor networks for measurement of electric fields under hvdc transmission lines," *Sensors*, vol. 15, no. 2, pp. 3540–3564, 2015.

[112] M.O. Ergin, V. Handziski, and A Wolisz, "Node sequence discovery in wireless sensor networks," in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, May 2013, pp. 394–401.

[113] K. Yedavalli and B. Krishnamachari, "Sequence-based localization in wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 1, 2008.

[114] Ziguo Zhong and Tian He, "Achieving range-free localization beyond connectivity," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, 2009, SenSys '09.

[115] Ziguo Zhong and Tian He, "Rsd: A metric for achieving range-free localization beyond connectivity," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 11, pp. 1943–1951, Nov 2011.

[116] A El Assaf, S. Zaidi, S. Affes, and N. Kandil, "Hop-count based localization algorithm for wireless sensor networks," in *Mediterranean Microwave Symposium (MMS), 2013 13th*, Sept 2013, pp. 1–6.

[117] S. Shioda and K. Shimamura, "Anchor-free localization: Estimation of relative locations of sensors," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, Sept 2013, pp. 2087–2092.

[118] Santar Pal Singh and S.C. Sharma, "Range free localization techniques in wireless sensor networks: A review," *Procedia Computer Science*, vol. 57, pp. 7 – 16, 2015, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

[119] Radu Stoleru, Tian He, and JohnA. Stankovic, "Range-free localization," in *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, Radha Poovendran, Sumit Roy, and Cliff Wang, Eds., vol. 30 of *Advances in Information Security*, pp. 3–31. Springer US, 2007.

[120] Asma Mesmoudi, Mohammed Feham, and Nabila Labraoui, "Wireless sensor networks localization algorithms: a comprehensive survey," *CoRR*, vol. abs/1312.4082, 2013.

[121] Alexander H Henderson, Gregory D Durgin, and Christopher J Durkin, "Measurement of small-scale fading distributions in a realistic 2.4 ghz channel," Tech. Rep., Georgia Institute of Technology, 2007.

[122] T. Chrysikos, G. Georgopoulos, and S. Kotsopoulos, "Attenuation over distance for indoor propagation topologies at 2.4 ghz," in *2011 IEEE Symposium on Computers and Communications (ISCC)*, June 2011, pp. 329–334.

[123] C. Phillips, D. Sicker, and D. Grunwald, "Bounding the error of path loss models," in *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on*, May 2011.

[124] Gomes Pedro Henrique, Ying Chen, Thomas Watteyne, and Bhaskar Krishnamachari, "Insights into frequency diversity from measurements on an indoor low power wireless network testbed," in *IEEE Global Telecommunications Conference (GLOBECOM), Workshop on Low-Layer Implementation and Protocol Design for IoT Applications (IoT-LINK)*, 2016.

[125] Z. Fang, Z. Zhao, D. Geng, Y. Xuan, L. Du, and X. Cui, "Rssi variability characterization and calibration method in wireless sensor network," in *The 2010 IEEE International Conference on Information and Automation*, June 2010, pp. 1532–1537.

[126] M.O. Ergin and A. Wolisz, "Node position discovery in wireless sensor networks," in *Positioning Navigation and Communication (WPNC), 2012 9th Workshop on*, march 2012, pp. 157 –162.

[127] M. Onur Ergin, Vlado Handziski, Arash Behboodi, and Adam Wolisz, "Determining node sequence in a linear configuration," in *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, Oct 2014, pp. 534–543.

[128] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2012, ISBN 3-900051-07-0.

[129] David Tse and Pramod Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, New York, NY, USA, 2005.

[130] Andrea Goldsmith, *Wireless Communications*, Cambridge University Press, New York, NY, USA, 2005.

[131] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus PJ Fromherz, "Localization from mere connectivity," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2003, pp. 201–212.

[132] M. O. Ergin, V. Handziski, and A. Wolisz, "Grid-based position discovery," in *2016 International Conference on Localization and GNSS (ICL-GNSS)*, June 2016, pp. 1–8.

[133] J.C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[134] "Deposit Once: Repository for Research Data and Publications," https://depositonce.tu-berlin.de/, Last accessed: 2017-12-19.

[135] "NodeIF: Node InterFace, a software for tinyOS(c) to command telosB-family nodes as radio interfaces from one Java process.," https://github.com/OnurErgin/NodeIf, Last accessed: 2017-09-10.