HYBRIDISATION OF GNSS WITH OTHER WIRELESS/SENSORS TECHNOLOGIES ONBOARD SMARTPHONES TO OFFER SEAMLESS OUTDOORS-INDOORS POSITIONING FOR LBS APPLICATIONS

By

HALGURD SARHANG MAGHDID

Department of Applied Computing

The University of Buckingham

United Kingdom

Thesis submitted for the Degree of Doctor of Philosophy in Computer Science to the School of Science in the University of Buckingham

December 2015

ABSTRACT

Location-based services (LBS) are becoming an important feature on today's smartphones (SPs) and tablets. Likewise, SPs include many wireless/sensors technologies such as: global navigation satellite system (GNSS), cellular, wireless fidelity (WiFi), Bluetooth (BT) and inertial-sensors that increased the breadth and complexity of such services.

One of the main demand of LBS users is always/seamless positioning service. However, no single onboard SPs technology can seamlessly provide location information from outdoors into indoors. In addition, the required location accuracy can be varied to support multiple LBS applications. This is mainly due to each of these onboard wireless/sensors technologies has its own capabilities and limitations. For example, when outdoors GNSS receivers on SPs can locate the user to within few meters and supply accurate time to within few nanoseconds (e.g. ± 6 nanoseconds). However, when SPs enter into indoors this capability would be lost. In another vain, the other onboard wireless/sensors technologies can show better SP positioning accuracy, but based on some pre-defined knowledge and pre-installed infrastructure. Therefore, to overcome such limitations, hybrid measurements of these wireless/sensors technologies into a positioning system can be a possible solution to offer seamless localisation service and to improve location accuracy.

This thesis aims to investigate/design/implement solutions that shall offer seamless/accurate SPs positioning and at lower cost than the current solutions. This thesis proposes three novel SPs localisation schemes including WAPs synchronisation/localisation scheme, SILS and UNILS. The schemes are based on hybridising GNSS with WiFi, BT and inertial-sensors measurements using combined localisation techniques including time-of-arrival (TOA) and dead-reckoning (DR). The first scheme is to synchronise and to define location of WAPs via outdoors-SPs' fixed location/time information to help indoors localisation. SILS is to help locate any SP seamlessly as it goes from outdoors to indoors using measurements of GNSS, synched/located WAPs and BT-connectivity signals between groups of cooperated SPs in the vicinity. UNILS is to integrate onboard inertial-sensors' readings into the SILS to provide seamless SPs positioning even in deep indoors, i.e. when the signals of WAPs or BT-anchors are considered not able to be used.

Results, obtained from the OPNET simulations for various SPs network size and indoors/outdoors combinations scenarios, show that the schemes can provide seamless and locate indoors-SPs under 1 meter in near-indoors, 2-meters can be achieved when locating SPs at indoors (using SILS), while accuracy of around 3-meters can be achieved when locating SPs at various deep indoors situations without any constraint (using UNILS). The end of this thesis identifies possible future work to implement the proposed schemes on SPs and to achieve more accurate indoors SPs' location.

ACKNOWLEDGEMENTS

First, **ALLAH THE MOST GRACIOUS AND MERCIFUL**: Who gave me this opportunity for doing research at this level. In addition, there are many people that I have to thank because without them, I would not be the PhD student that I am today.

My family: I would like to dedicate this thesis to my darling SHATU (my daughter) who has been my inspiration to be a successful person she can be proud of. I would also like to thank my soul mate, my best friend, and my wife SHNO who has been there for me every step of the way during this PhD. Without her continuous encouragement, my PhD would never be completed….

My mother, my brother and my sister: In these moments I cannot forget soulful persons who have given me everything, and have prayed day and night for me and for my family… thanks for all of you.

My supervisors: I would like to express my gratitude to my supervisors (Dr. Ali Al-Sherbaz, Dr. Naseer Al-Jawad and Dr. Ihsan Lami) for their guidance throughout this research study. I highly appreciate all their valuable comments, suggestions, and discussions.

Staff and Colleague: I am highly indebted and thoroughly grateful to staff at Applied Computing Department and my colleagues. Also, I would like express my sincerest gratitude Prof. Sabah Jassim, who offered me this great opportunity and many thanks for his continuous motivation & support…I would also like to thank my personal tutor (Dr. Naseer Al-jawad) for being a very good listener and for his continuous support and encouragement…

Last but not least: I am thankful to Koya University in Kurdistan-Iraq for offering me the scholarship…

ABBREVIATIONS

AAL	Ambient Assisted Living
AM	Acceleration-Magnitude
AMV	Acceleration-Moving Variance
AP	Access-point
ACK	Acknowledged
AGPS	Aided-GPS
AOA	Angle-of-arrival
ARE	Angular Rate Energy
BLE	Bluetooth Low Energy
BS	Base-station
BT	Bluetooth
Cell-ID	Cell-identification
CTS	Clear to Send
CDMA	Code Division Multiple Access
DB	Database
DR	Dead-reckoning
DOP	Dilution-of-precision
DWC	Dynamic WAP clock
EGPS	Enhanced GPS
E-OTD	Enhanced observed-time-difference
EKF	Extended Kalman filter
FCC-E911	Federal communications commission enhanced-911
GLRT	Generalized likelihood ratio test
GLONASS	GLObal NAvigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HW	Hardware
HDOP	Horizontal Dilution of Precision
HR	Human Resources

IMES	Indoors messaging system
ISM	Industrial, Scientific, and Medical
KBLS	Knowledge-based logistics system
LLA	Latitude-longitude-altitude
LLS	Linear-least-squares
LOS	Line-of-sight
LMU	Location Measurement Units
LBS	Location-Based Services
LTE	Long Term Evolution
MM	Map-matching
NFC	Near Field Communication
NLOS	Non-line-of-sight
ONS	Online Social Networking
OTDOA	Observed-TDOA
ppb	Part per-billion
ppm	Part per-million
PRP	Permutation Reference-Positions
POI	Point-Of-Interest
PSAP	Public Safety Answering Point
RBS	Reference Broadcast Synchronisation
RTLS	Real-time location system
RSS	Received signal strength
RTS	Request to Send
RMSE	Root mean square error
RTT	Round-trip-time
SV	Satellite Vehicle
2G	Second Generation
SUPL	Secure User Plane Location
SNR	Signal-to-noise-ratio
SNTP	Simple Network Time Protocol
SLAM	Simultaneous localisation and mapping

SILS	Smart indoors localisation scheme
SPs	Smartphones
SW	Software
SMSR	Switching Master/Slave Role
3G	Third Generation
3D	Three dimensions
TA	Time-advance
TDOA	Time-difference-of-arrival
TOA	Time-of-arrival
TOF	Time-of-flight
TTFF	Time-to-first-fix
2D	Two dimensions
UNILS	Unconstrained indoors localisation scheme
U-TDOA	Uplink-TDOA
VL	Very Low
VH	Very High
WAP	WiFi access point
WPS	WiFi Positioning System
WiFi-A-GPS	WiFi-Assisted-GPS
WiFi	Wireless Fidelity
WLAN	Wireless local area network
ZUPT	Zero-velocity-Update

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DECLARATION

I hereby declare that all the work in my thesis entitled (HYBRIDISATION OF GNSS WITH OTHER WIRELESS/SENSORS TECHNOLOGIES ONBOARD SMARTPHONES TO OFFER SEAMLESS OUTDOORS-INDOORS POSITIONING FOR LBS APPLICATIONS) is my own work except where due reference is made within the text of the thesis.

I also declare that, to the best of my knowledge, none of the material has ever previously been submitted for a degree in the University of Buckingham or any other University.

Halgurd Sarhang Maghdid

CHAPTER1. INTRODUCTION

1.1 Overview

SPs and tablets, driven by mobile-services/applications such as LBS, are becoming important to our communication, localisation and information needs. The need for advanced LBS in terms of applications variety and accuracy is increasing every year since the emergence of the SPs few years ago [1].

Equally, SPs manufacturers are mounting powerful processing capability and several wireless communication and localisation technologies to cater for such LBS applications. The SPs include GPS and GLONASS receivers and WiFi, BT, NFC, LTE/3G/2G transceivers as well as Octa-core 2.1 GHz processors dedicated for applications including ones that try to continuously locate the geographical position of the SPs [2].

The availability of LBS applications on SPs has expanded the challenges for offering diverse applications and precise/accurate localisation. Examples of such LBS applications include tracking users via Telemetric for security and safety, helping the user navigating outdoors-indoors of large buildings such as hospitals, or applications for advertising and management, billing, gaming, social networking, finding nearest restaurant/shop and other POI information [3].

However, LBS applications that depends only on GNSS receivers will be restricted to provide seamless positioning due to the weakened/limited GNSS signals reception in high-dense urban areas and indoors [4]. Therefore, most of the current LBS applications do attempt to hybridise multi-GNSS signals such as GPS plus GLONASS with cellular, WiFi, BT, NFC and inertial-sensors on SPs with other pre-installed infrastructure sensors.

The focus of my research study is to offer seamless positioning solutions capitalising on the use of existing infrastructure and onboard SPs devices/sensors such as GNSS, WiFi, BT and inertial-sensors as well as the cooperation with other SPs available in the vicinity.

This thesis explores the problem of seamless positioning off-the-shelf SPs and offers solution which is more accurate, on-the-go and at lower cost than the contemporary localisation solutions currently available on SPs. Available solutions and techniques that are used to locate SPs are explained in chapter-2 and they are evaluated based on location accuracy. As well as, the advantages and disadvantages of each solution in general and limitations of utilised hardware are presented. My choice of hybridisation onboard SPs technologies to provide seamless positioning as the primary research direction is explained followed by an overview of my novel localisation schemes. In addition, to prove and to implement the proposed schemes, Android-based SPs are selected because Android is fully supported, widely used and proved, open and free mobile platform [5].

The purpose of this chapter is to understand the problem background, to highlight the challenges associated with SP based localisation solutions and to state the research motivation. After highlighting these, this chapter introduces research objectives and contributions as well as addresses the most significance of three published novel schemes and organises the structure of the rest chapters of the thesis.

1.2 Problem Background

As a result of the literature survey of recent published localisation solutions/techniques used in LBS for SPs, the followings are the most important challenges in designing new generation of such localisation solutions:

1. Supplying positioning performance with reliable and high location accuracy. Generally, indoor environments require high accuracy to be useful for practical LBS purposes. This is because, when indoors-SPs are dealing with objects and distances at a smaller scale. However, due to complex indoors structure, multipath and reflected signal issues irrespective of the single/hybridised techniques used and cumulative positioning errors the target accuracy remains unresolved [6].

2. Providing seamless and on-the-go positioning from outdoors to indoors. Most recent LBS applications need continuous positioning to perform their services

such as patient monitoring, museum and fire-fighter [7]. However, outdoors and indoors positioning are basically considered as two separate problems. The reason being a solution designed for one environment either will not work in the other or will require significant hardware and/or software modification of the solution.

3. Providing low-cost positioning solution by utilising only the onboard GNSS receivers, wireless transceivers and inertial-sensors and using only the existing wireless infrastructure, i.e. without dedicated hardware/networks. For example, RSS fingerprinting technique that is typically associated with indoors positioning solutions, SPs can define their location within few meters accuracy. However, dedicated hardware such as host server, sensor network on premises, and calibration to survey positioning area for such localisation technique deduces huge cost [8].

Current onboard SP GNSS receivers have been developed with increasing performance and its accuracy and are now able to operate in much more severe signal-degraded environments than before [9]. Following these achievements of GNSS-based location services in outdoors, the challenges of LBS applications have shifted to supply of such services for indoors. However, still the GNSS performance is degraded in dense-urban areas or indoors due to the availability of weak GNSS-signals or multipath signal propagation issue.

Other outdoor localisation systems usually are utilised to cover only densely urban areas and are designed to either replace GNSS or enhance the GNSS-receiver accuracy. They use alternative signal sources such as Wi-Fi or LTE/3G [10], because they are abundant in urban environments while GNSS suffers from various problems caused by the density of high buildings around the user. However, the obtained accuracy, for example 5 to 25 meters through these solutions is not acceptable by most of the LBSs [11].

For indoors, localisation solution has a different picture and there is no single accepted commercial solution for SPs, yet [12]. That could be because most of such solutions do not function well in a completely new environment as each localisation solution has to be carefully adapted for each new area. Although SPs are not designed to define their location other than with GNSS receiver, other wireless technologies including BT, WiFi and LTE and inertial-sensors onboard could be used for this purpose as well. For example, every single SP has WiFi transceiver which can be used to receive WAPs beacon signals. WAPs broadcasts these beacon-signals and these beacons travel relatively through air, and by using the difference between when the time signal was generated at multiple WAPs and the time it was received at SP it is possible to determine the SP's position through Trilateration. Under ideal circumstances, for example: perfect clock synchronisation between SPs and WAPs, LOS signal, and accurate calibration, this technique can in theory deliver sub-meter accuracy. However, inaccuracy of WAPs clocks and indoors complex structure are the main sources of error that can degrade the accuracy of WiFi-based positioning and it was therefore necessary to research how to mitigate or even eliminate them. For example synchronisation can be completely solved by using an accurate reference time such as the GNSS time on SPs, when outdoors. Another example, a localisation solution may need its own set of deployed transmitters to broadcast beacons around the environment where a more dense distribution will result in higher accuracy (like iBeaconing [13]). In addition, it has also been shown that location accuracy around 3 meters can be achieved through the use of electromagnetic waves such as BT and Wi-Fi signals, although so far such results were obtained only with custom built hardware and are not available to general users [14].

Another completely different approach utilises onboard SP inertial-sensors such as accelerometer and gyroscope. No beacons are needed but these inertial-sensors' measurements typically result in large positioning error over short time due to large cumulative sensor-drift errors unless these errors are bounded by measurements from other localisation systems [15].

1.3 My Research Motivation

Ever since I have worked on designing and implementing network security tools on mobile-devices within my MSc degree in applied computing and my BSc in software engineering. In 2010, the University of Buckingham offered me a place as a PhD student in the Department of Applied Computing to conduct my own research. I was delighted by this offer and the opportunity was to contribute to the research in mobile-services' application development especially with the LBS applications.

I started by studying the applications development based on hybridisation of different wireless and sensor technologies onboard SPs to offer outdoors-indoors seamless positioning, something that some companies in my country and from abroad can benefit from. Taking into consideration the limitations associated with current localisation solutions, I believe that developing a solution works by using onboard SPs devices will make such solutions inexpensive, accessible, and easier to deploy.

I believed that, when different wireless and sensor technologies are integrated, a sufficient LBS application can be developed to offer not only seamless positioning, but also to offer accurate and provide low-cost solution which are the main LBS users' demands. I.e. I imagined that this research work with this technologies integration can help impact the current and future SP-LBS due to being easier to deploy and offer high localisation performance at lower cost when compared with current localisation solutions.

I have chosen this research field to help myself understand the concepts of the new SP localisation solutions when deployed by hybridising GNSS with other wireless and/or inertial-sensors technologies. This is to meet the aforementioned LBS users' requirements, since current SP localisation solutions don't satisfy all these requirements in a single solution yet. Therefore, this research study shows how to design and implement for such novel solutions on SPs. I have thoroughly enjoyed this research experience and I would like to continue further work on this hybridisation-technologies solution in the future.

1.4 Research Objectives and Contributions

The main aim of this research study is to design and implement seamless, accurate, and low cost SP positioning solutions that withstand some of the challenges illustrated in

(section 1.2) and meets the main emerging challenges explained earlier in previous sections. The objectives of this study can be summarised in the following four key points:

1. To perform theoretical and empirical investigation of wireless and inertial-sensors technologies onboard SPs, which aims to properly assess the limitations and capabilities in terms of SP positioning. Furthermore, analysing and understanding the trade-off between localisation performance metrics including accuracy, applicability, cost, battery-power consumption, time to fix, coverage and robustness.

2. To supply novel schemes that improves accuracy of indoors-SPs location using multi devices/inertial-sensors measurements.

3. To provide outdoors-indoors seamless localisation by hybridising of GNSS with other onboard wireless/sensor technologies including WiFi, BT, and inertial-sensors to aid indoors-SPs localisation for LBS applications.

4. To provide practical low-cost and easy-deployable solution, i.e. proposing SP localisation schemes that will only capitalise the existing infrastructure, neither need survey of the area nor need for special external hardware/sensors as well as cooperate with other SPs available in the vicinity, yet offering high localisation performance solution.

The above objectives can be summarised in the following research questions:

1. Question 1: What are the abilities and limitations of SPs localisation techniques which are associated with current available solutions? (**addressed in section 2.2**)

2. Question 2: What are the main error sources of onboard SP wireless/sensor technologies for localisation? (**addressed in section 2.3**)

3. Question 3: Do the existing SP localisation solutions based on particular localisation technique offer seamless outdoors-indoors positioning service? (**addressed in section 2.4**)

4. Question 4: What is the accuracy of the available timestamp function sources on SPs for estimating TOF? (**addressed in section 3.2.2**)

5. Question 5: How obtained GNSS time on SPs is reliably used to synchronise WAPs in the vicinity? (**addressed in section 3.3**)

6. Question 6: how can synchronised-WAPs be accurately located? (**addressed in section 4.3**)

7. Question 7: How accurately can indoors-SPs position be defined using hybrid onboard SP wireless technologies? (**addressed in section 5.4**)

8. Question 8: Can BT-Hop synchronisation between the connected SPs in BT-network provide pseudorange measurements? (**addressed in section 5.4.1.1**)

9. Question 9: How can the position of indoors-SPs be obtained without using any localisation infrastructure, unconstrained and cooperatively? (**addressed in section 5.5**)

10. Question 10: Can fused onboard SPs wireless transceivers and sensor measurements prevent the accumulated error of the defined indoors-SPs position? (**addressed in section 5.5.1.3**)

Answers to these questions will produce the following distinct contributions to the field of seamless outdoors-indoors SPs localisation: (the main contributions of this study are illustrated in Figure 1-1)

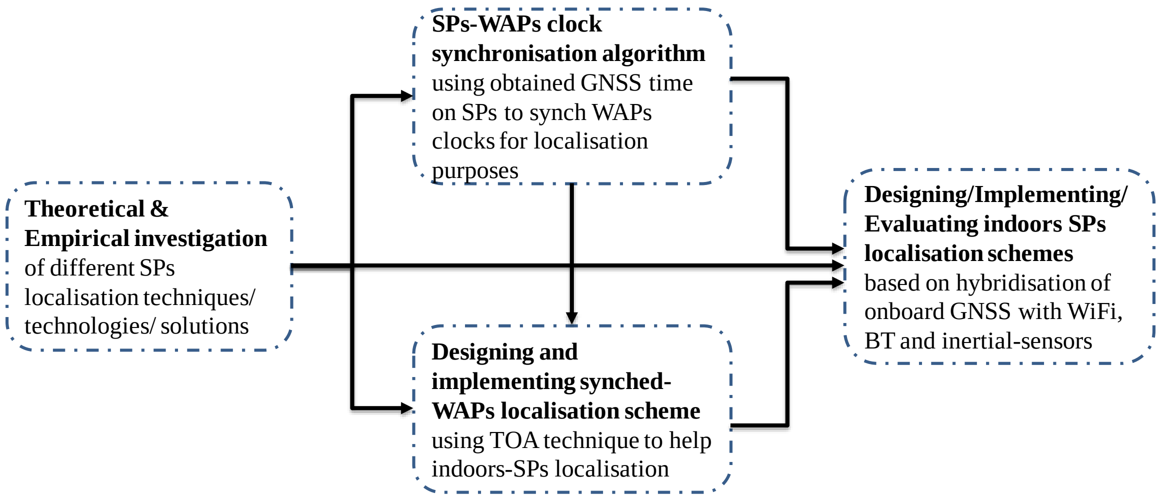


Figure 1-1: Research contributions.

1. This research study is started by presenting theoretical and empirical analysis of different localisation techniques as well as SPs localisation solutions using different technologies which are available on SPs including GNSS, LTE/3G, WiFi, BT, and inertial-sensors. Furthermore, highlights some advantages as well as limitations of each solution. We established the following findings:

   a. Current localisation solutions available on SPs have own capabilities and limitations. Without establishing additional hardware or without pre-defined constraints, no one of these solutions can offer seamless positioning on-the-go, anywhere, anytime.

   b. Accurate indoors-SPs localisation solutions are most likely to be based on hybridisation of GNSS with other wireless technologies such as Wi-Fi and BT technologies as well as integrated with local inertial-sensors.

   c. Achieving accuracy of SPs location in localisation solutions is strongly related to: pseudorange measurements, environmental complexity, using localisation techniques as a standalone or as a combined approach, hardware or software designed of the solutions and estimating/calculating 'SPs location' methods including Proximity, Trilateration and fingerprinting.

2. To develop different localisation hypothesis and various scenarios by contributing group of SPs and WAPs using TOA technique, this research study proposes SPs-WAPs clocks synchronisation algorithm based on a developed WAPs clock model. The WAPs clock model uses obtained GNSS time on SPs as the reference clock for the clock synchronisation, together with WAP clock offset and accurate clock drift model based on real noise factors such as white noise, flicker noise and random walk noise. The model is also designed to be dynamic to accommodate various WAPs hardware and their position.

3. Proposes a hybridisation scheme of GNSS with WiFi technology to help seamless SP outdoors-indoors positioning on-the-go, anywhere, anytime and to design independent of pre-installed infrastructure solution. The proposed scheme uses any outdoors-available GNSS-enabled SPs within reach of WAPs to locate the synchronised WAPs to aid indoors-SPs localisation using Trilateration. This proposed hybridisation solution is also providing a low-cost and easy to deploy localisation solution, since it utilises the existing WAPs in/around buildings and it doesn't need any dedicated hardware.

4. This study also proposes two indoors-SPs localisation schemes including SILS and UNILS. The schemes are to locate indoors-SPs and to get better position accuracy. These schemes calculate location of SPs seamlessly from outdoors into indoors based on hybridisation of GNSS, WiFi and BT measurements as well as utilising local inertial-sensor readings.

During this research study, the following papers were published:

1. Halgurd S. Maghdid, Ali Al-Sherbaz, Naseer Al-Jawad and Ihsan .A. Lami "UNILS: **Un**constrained **I**ndoors **L**ocalization **S**cheme based on cooperative smartphones networking with onboard inertial, Bluetooth and GNSS devices," IEEE/ION PLANS, Savannah, Georgia-USA, April 11-14, 2016.

2. Halgurd S. Maghdid, Ihsan .A. Lami, Kayhan Z. Ghafoor and Jaime Lloret

"Seamless Outdoors-Indoors Localization Solutions on Smartphones: Implementation and Challenges",

ACM Computing Survey Journel, Volume 48 Issue 4, March 2016. **Impact Factor for 2015 is 3.373.**

3. Ihsan .A. Lami, Halgurd S. Maghdid, and Torben Kuseler

   "SILS: a Smart Indoors Localization Scheme based on on-the-go cooperative Smartphones networks using onboard Bluetooth, WiFi and GNSS,"

   ION GNSS+ 2014, At Tampa Convention Center, Tampa, Florida-USA, 8-12 Sept 2014.

4. Ihsan A. Lami and Halgurd S. Maghdid

   "Synchronising WiFi access points with GPS time obtained from smartphones to aid localisation,"

   International Conference on Computer Applications Technology (ICCAT), 2013, Sousse, 20-22 Jan. 2013.

5. Halgurd S. Maghdid and Ihsan .A. Lami

   "Dynamic clock-model of Wi-Fi access-points to help indoors localisation of smartphones,"

   4th International Congress on Ultra Modern Telecommunications ICUMT), 2012.

1.5 Significance of Study

The proposed schemes in this study are significantly contributed to offer continuous SP positioning from outdoors into indoors. This is achieved by proposing SPs-WAPs clock synchronisation algorithm using obtained GNSS time, on SPs when outdoors, as an accurate reference time. Moreover, locating these synched WAPs in the vicinity that contributes in eliminating a big part of the SP positioning cost and makes solution would be easily applicable. In another vain, the scheme neither needs any pre-defined WAP location information nor deploys additional hardware to survey/collect WAP location information. This study is also important to provide a better accuracy indoors-SP localisation solution in comparison with the current solutions. This is done by using the proposed of indoors localisation schemes based on multi-measurements functions including:

1. BT to BT relative-pseudoranges based on TOA technique between the participated SPs based on hop-synchronisation and Master-Slave role switching to minimise the pseudoranges error.

2. Selecting fixed GNSS location of outdoors-SPs with good geometric reference positions to solve DOP issue which has influence on indoors-SPs position accuracy.

3. WAPs-SPs Trilateration estimates for deep indoors localisation.

4. New unconstraint fusion measurements by using only available devices/sensors on SPs including GNSS, BT and inertial-sensors, i.e., even when communication with existing infrastructures such as WAPs or BT-anchors is considered unavailable. The fusion measurements include relative- pseudoranges using TOA, distance-displacement and heading estimation using DR technique of the networked SPs. This fusion by using Kalman Filter exploits the advantages of each of these techniques while compensating for their limitations.

Thus, all proposed schemes including the developed algorithms in this research study, as a localisation solution, is a good candidate for seamless LBS localisation applications on SPs. Since the proposed localisation solution is accurate, low-cost, on-the-go, anytime, anywhere and without using any pre-installed and calibrated infrastructure or dedicated Internet based data/server.

1.6 Thesis Organisation

The rest of this thesis is divided into the following chapters:

Chapter-two provides a background, problems and literature reviews of the area of study as well as evaluates the current SP localisation techniques, technologies and solutions. Later, the strengths and weaknesses of the reviewed localisation solutions in relation to SP localisation approaches are highlighted.

Chapter-three addresses the main challenges and factors of time-based measurements, clock synchronisation and pseudorange estimation. This study is important to introduce the design detail of the proposed SPs-WAPs clock synchronisation algorithm based on the developed WAPs clock modelling and using GNSS time that is obtained from SPs to aid localisation.

Chapter-four explains my first published scheme to obtain WAPs locations and mitigate DOP issues when performing TOA technique. This chapter details the results of simulation experiments, obtained from OPNET, that I performed using appropriate scenarios to validate the scheme, including using single-SP and multiple-SPs in the vicinity.

Chapter-five focuses on explaining of the cooperative localisation solutions using wireless/sensor technologies onboard SPs to locate SPs, in **section 5.3**. This work has lead to published scheme SILS to locate the indoors-SPs. The SILS implementation algorithms and simulation results, obtained from OPNET, will be evaluated in various scenarios and experiments based on hybridisation of GNSS, BT and WiFi measurements in **section 5.4**. Then in **section 5.5**, the proposed UNILS using the fused onboard wireless technology and inertial-sensors measurements is presented. Finally, **section 5.6** gives a summary on the achievements from both SILS and UNILS as well as provides a comparison with the current localisation solutions.

Chapter-six summarises the research’s achievements and suggests directions for future research.

CHAPTER2. BACKGROUND AND LITERATURE REVIEW ON SP LOCALISATION SOLUTIONS

This chapter starts by presenting background description of the SPs location determination and imperfection localisation factors relevant to the context of the thesis. It also reviews the current literature and gives an overview of the related work in the field of SP localisation. The review provides localisation techniques and solutions which are adapted on SPs via onboard wireless and sensor technologies including GNSS, cellular, WiFi, BT and inertial sensors.

This chapter is divided into four sections: overview on SPs localisation, localisation techniques, onboard SPs wireless/sensors technologies for localisation and SPs localisation solutions. In case of overview on SPs localisation the questions are:

1. How can an SP define its position? (**see section 2.1.1**)

2. What are the metrics used to evaluate the performance of SPs localisation solution? (**see section 2.1.3**)

3. What are the main demands for LBS-SPs users? (**see section 2.1.4**)

In relation to localisation techniques section two questions are raised for each technique: (**see section 2.2**)

1. How could the technique be performed to define SP's location?

2. What are the advantages and constrains of the technique?

For onboard localisation technologies section the questions are: (**see section 2.3**)

1. What is the ability of each technology?

2. What are the error sources of the technology which are contributed for SP localisation?

And finally for localisation solution section, the question will be: does the solution can offer accurate and seamless positioning at reasonable cost? (**see section 2.4**)

The combination of all these aspects examines the current SPs localisation solution.

2.1 SP Localisation Concepts

This section aims: to give an overview of SPs localisation, to address error sources and to present some metrics that can be used to evaluate the performance of localisation solutions on SPs including: location accuracy and precision, time to fix, battery-power consumption, applicability, cost, robustness, and coverage.

2.1.1 Overview

This subsection addresses the question of "How can an SP define its position?"An SP location is calculated, typically, by considering measurements of distances or angles (referred to observations) to reference points whose positions are known. SPs can measure these observations from onboard wireless chipsets and sensors through running a particular localisation application. For example, an SP can measure the distances from multiple WAPs through the RSS measurements or TOF of the WAPs-received signals. These measured distances together with known WAP position then can be used to define the SP location.

An SP location may be relative or absolute. A relative location is depicted by a distance and/or heading in relation to a particular known position. The known position may be other SP's position or a BS/WAP/BT-anchor's position. While an absolute location has 2D or 3D coordinates that are common to a large defined region, for example the globe.

2.1.2 Error Sources

The aforementioned observations, however, are liable to errors coming from different factors that impact on the result of SP location estimation including:

1. Complex/harsh structure of the positioning area, for example indoors structure, will introduce multipath and NLOS signal issue when location estimation will based on wireless signal-measurements [16],

2. Measurement noise due to low material quality of the sensors and wireless transceivers on SPs which are introducing big error in localisation such as: WiFi transceivers-clock time error [17] or drift issue of inertial-sensor readings [18],

3. Errors due to techniques/mathematical calculations to define SP position such as: Trilateration, Triangulation, Fingerprinting, and DR [19].

4. Constellation of reference positions such as BSs/WAPs (or satellites for GNSS systems), i.e. DOP issue, when Trilateration process is applied to calculate SP location [20].

The detailed analysis of the first three error sources is presented separately in the next sections. Also, the geometric shape of reference positions (as they are presented to an SP) affects the obtained accuracy of the SPs position. In an ideal localisation solution, SPs will be designed to receive signals from available references positions in a manner that minimise the DOP issue.

Figure 2-1 illustrates an example of DOP issue where BSs/WAPs being tracked are installed near to each other, i.e. approximately are parallel on a line. As it can be observed, it is difficult to define where the circles (through their pseudoranges) intersect. Therefore, it is not easy to define the SP location where BSs/WAPs are clustered closely together.

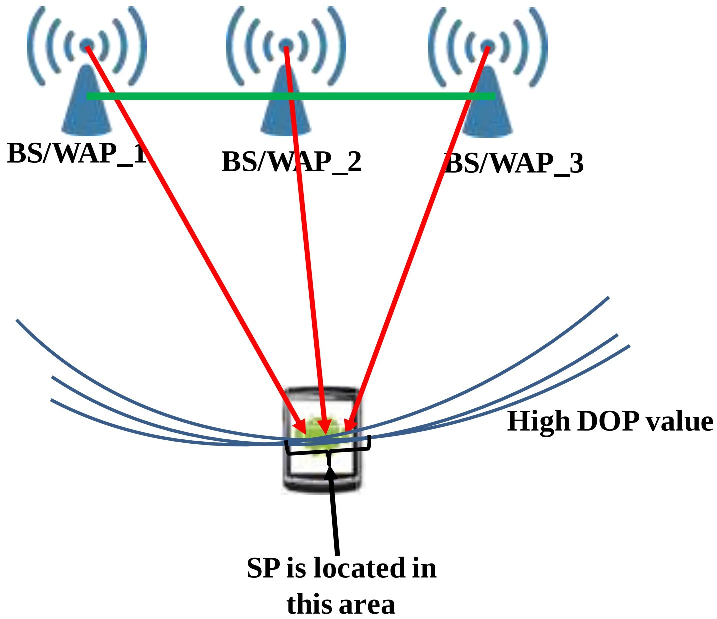


Figure 2-1: Poor reference positions installation.

To reduce the impact of DOP issue, as shown in Figure 2-2, one extra of a pseudorange measurement to a BS/AP that is angularly spaced from the group of tracked BSs/WAPs allows SP to determine its position more accurately.

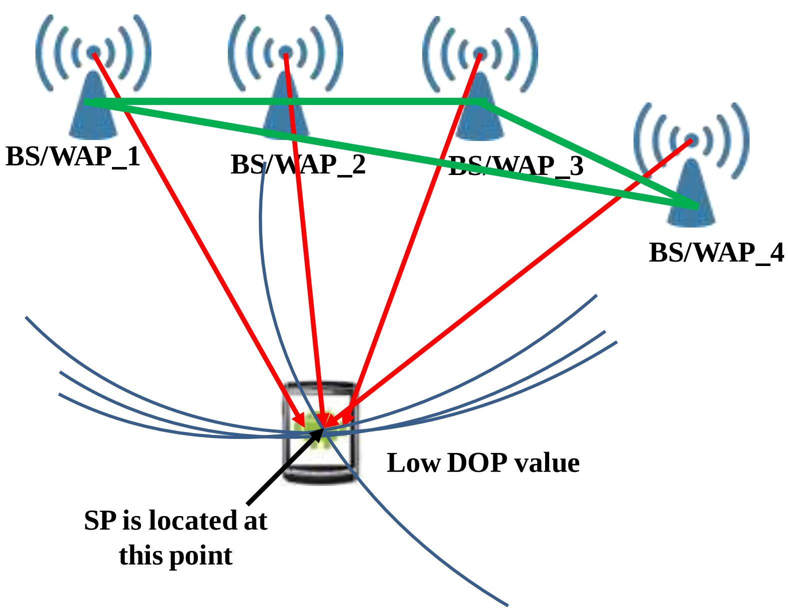


Figure 2-2: Improved geometry-shape of reference positions.

2.1.3 Performance Metrics

In this section, the question of "What are the metrics used to evaluate the performance of SPs localisation solution?" will be answered. Despite the obtained SP position and the error sources, there are some metrics to evaluate the performance of SP localisation solutions. One of the main metrics is the accuracy of the obtained SP position. Due to the existing uncertainties of measurements in the localisation solutions, such as timing error and measurements noise, the estimated SP's location will be characterised by errors. The location estimation error at a specific instance t is given by the average euclidean error between the estimated location and the true location [21], as expressed in equation (2.1):

$$Le = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(\widehat{x_i} - x_i)^2} \tag{2.1}$$

Where Le is the estimated average euclidean error of the location, N is the number of measurements, for example 3 for XYZ coordinates in 3D or 2 for XY coordinates, $\widehat{x_i}$ is the estimated location coordinates and x_i is thr true location coordinates. From the errors, one can form different statistics that can be used as accuracy measuring of a given method. For example, one may use the RMSE between the estimated location and the true location [21]. RMSE is expressed as in equation (2.2):

$$rmse(L) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}[EC_k - TC_k]^2} \tag{2.2}$$

Where: N is the number of measurements, EC and TC are estimated and true location coordinates respectively, and k is the index of the coordinate measurements. The RMSE may not completely serve to describe the accuracy an associated method. For example, one may use the percentage of confidence (precision) in the estimate [9]. Precision considers how reliably the localisation solution works, i.e., it is a measure of the strength of the localisation solution as it detects the variation in its execution over numerous iterations. For example, if solution 'A' has a location precision of 68% within 5 meters & 96% within 7 meters; and solution 'B' has a precision of 50% within 5 meters & 96%

within 7.6 meters. In this way, the assessment could pick the solution 'A' as a result of its higher precision. Figure 2-3 illustrates the distinction between the various estimated locations' accuracy & precision in comparison with the true location.

Applicability and cost are also used to evaluate localisation solutions [22]. These metrics measure the physical restrictions and necessities associated with the implementation and use of certain technology regarding of technical issues. Despite great benefits in performance, a localisation solution may be inapplicable on SPs if its installation & operational expense is high in comparison with some other low-cost solutions. Furthermore, a low cost SP-localisation solution is to utilise an existing infrastructure without any dedicated hardware. For example, WiFi positioning system on SP may be considered to have no hardware cost, since all the necessary units of that system including: WAPs and WiFi-transceiver onboard SPs have already been installed/equipped for Internet services purposes.

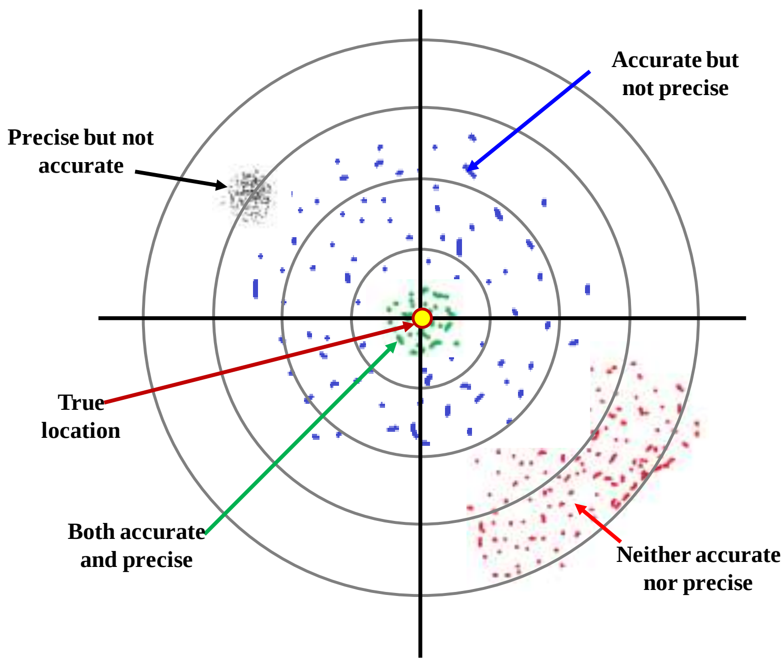


Figure 2-3: Estimated location in various accuracy and precision.

In another vain, specifically for SP positioning, battery lifetime & TTFF are other possible metrics to show the localisation-solution performance [23]. Battery-power consumption measures an amount of consumed energy during any locating SP trials. Time-to-fix records the time elapsed when the wireless/sensor device is power-on until the output of the localisation solution within particular accuracy. The high demand for low-power consumption and short TTFF is acknowledged in the emergency cases as well as from the LBS-users' perspective. Yet, both the elapsed time and the battery-power consumption during positioning have tracked little attention, especially in the field of SPs localisation.

To evaluate the effectiveness of a localisation solution, the problem of the robustness and determining the network coverage for a designated area are also important [24]. Coverage metric indicates the area where SPs can be located by the method within a specific accuracy, while the robustness-metric is the resistance of the method to some weaknesses including lack of radio visibility, self-measurement noises, and access-point/anchors failures.

A positioning solution with high robustness could work regularly even when some observations are not available, for example LOS signals from reference positions such as WAPs signal are completely blocked. Or when some of the observations are obtained which are never seen before, i.e. they are out of function and damaged in a harsh environment such as high vibration heading estimation using magnetometer and accelerometer when indoors caused by interferences. In these cases, robust positioning solutions have to use this incomplete information to compute the location in combined with other measurement units within a particular localisation technique.

2.1.4 LBS Application on SPs

This subsection addresses the question of "What are the main demands for LBS-SPs users?" LBS is an information or service, able to access with onboard devices/sensors SPs through the mobile network and utilising the ability to make use of the geographical position of the SPs [25].

LBS include services to identify a location of a person or object. LBS applications are becoming a crucial aspect of mobile computing. The most recent LBS applications which are interest to by SPs-users within different scenarios and categories are showed in Table 2-1.

Table 2-1: LBS applications on SPs in different categories **[26]**, **[27]**, **[28]**.

LBS Categories	**Scenarios**	**Applications on Actual-SP**
Marketing	Shopping centres advertise for their items using location information of LBS-users	ShopKick
Emergency	LBS-users call the emergency response agency in fire, stolen and abnormal situations	911 in US & 112 in EU via nearest PSAP
Geotagging	Finding location of touristic services using geotagged images	GeoRSS
Tracking	LBS-users can track on SPs exact location of the bus to be sure about the path and the schedule	PDX Bus
Navigation	LBS-users can use location information and Map information to navigate through the path of the trip to a specific destination	Google Places
Mobile Location-Based Gaming	Treasure hunts (e.g. GeoSocial & Geocaching)	SCVNGR
Location Based Social Media	LBS-users use their location information to keep their relation via Facebook and/or Twitter	Gowalla, Loopt, Facebook Place, Foursquare
Sports	Real-time route of outdoor sport activity via SP using Google Maps and sharing that data with a social networks	Nike+, Run Keeper, Endomondo
Billing	Using location information to charge LBS-users, when they access a particular services	On-Board Units
POI	Discovering nearest cafes, restaurants, petrol stations as well as real-time traffic information	OpenTable, Fandango, Vouchercloud, NearbyFeed

LBS applications on SPs are arising as current and/or next-generation 'killer apps' [29]. However, high performances such as accuracy & reliability for particular applications, determination of physical location in different environments (i.e. seamless positioning from outdoors into indoors) and low cost adoption are some challenges to meet the LBS-users demand [11]. In another vain, the usability of LBS applications will be grown and emerged when these limitation are solved. In the next sections, the available localisation techniques and technologies within several commercial and research oriented solutions for these LBS applications are presented in detail.

2.2 Localisation Techniques

This section addresses the first research question (**see section 1.4**): what are the abilities and limitations of SP localisation techniques which are associated with current available solutions? As well as, in the next subsections the question of "How can the techniques be performed to define SP's location?" will be answered.

Location information provides an important role in most current SPs' services including traffic information for navigation and POI information for routing/planning and emergency calls. Localisation techniques such as AOA, RSS, Cell-ID/Proximity, time-based, MM technique and DR have been developed to achieve these services [30]. Mainly, such services need high quality of performance from the localisation solutions.

Combination of different location techniques is possible to make a powerful localisation solution including reasonable accuracy, short time to define SPs' location and low battery power consumption.

Figure 2-4 displays the taxonomy of such techniques as well as shows new combined localisation techniques including: TOA & DR (which is the proposed in this thesis, see **section 5.5.1.3**), RSS-Fingerprinting & TOA [31], MM & DR [32] and Proximity & RSS-radio propagation model [33]. The details of these techniques-implementation in the next subsections with their requirements/limitations and their ability are presented.

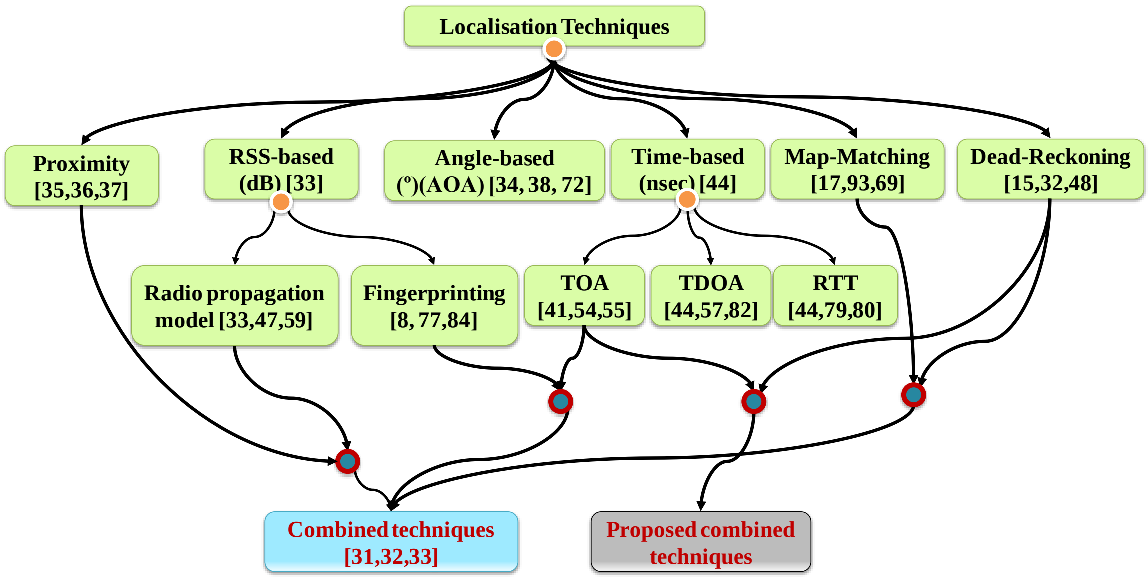


Figure 2-4: Taxonomy of SP-localisation techniques, including references.

2.2.1 Cell-ID (Proximity) Technique

Cell-ID or proximity technique is a simple localisation technique. It refers to define location of an SP as being within radio coverage of a reference position such as WAPs, cellular base-stations and BT-anchors. Thus, the SP is known to be within the area around that reference position. Therefore, the issue here is that the accuracy of the defined SP location is based on the radio coverage, i.e. cell size, of the reference position. For example, in cellular networks the cell size lies between 2 Km to 20 Km [34]. Certainly now in urban area the cell size is reduced to only tens of meter.

Additionally, this technique has been used in WiFi networks, since the cell size of these networks is much smaller than the one in cellular network. However, the accuracy of this technique in WiFi networks depends on the effective signal propagation pseudoranges as well as the density & distribution of WAPs in the area [35]. Several solutions have been proposed for this technique to improve location accuracy, especially for cellular technology, as illustrated in Figure 2-5, including providing cell sector, providing cell ID with time advance Cell-ID +TA and providing Cell-ID with max RSS value [36] [37].

In cell sector: the cell is divided into sectors, such as by using directional base station antennas with 120' beam width antenna. In such cases, the obtained location accuracy of SPs can be more narrowed by taking only the coverage of the received-signal sector. Also, further improved accuracy can be achieved by reading the signal-received measurements either based on timing, for example measuring RTT of the received signal or based on RSS measurements.

Practically, this technique is the easiest technique to implement on SPs as well as it takes short time and consumes low power to locate SPs. However, the accuracy of this technique is not enough for most of SP LBS applications, especially when the SPs are indoors, i.e. implementing this technique as a standalone or hybrid with other technique needs further investigations.

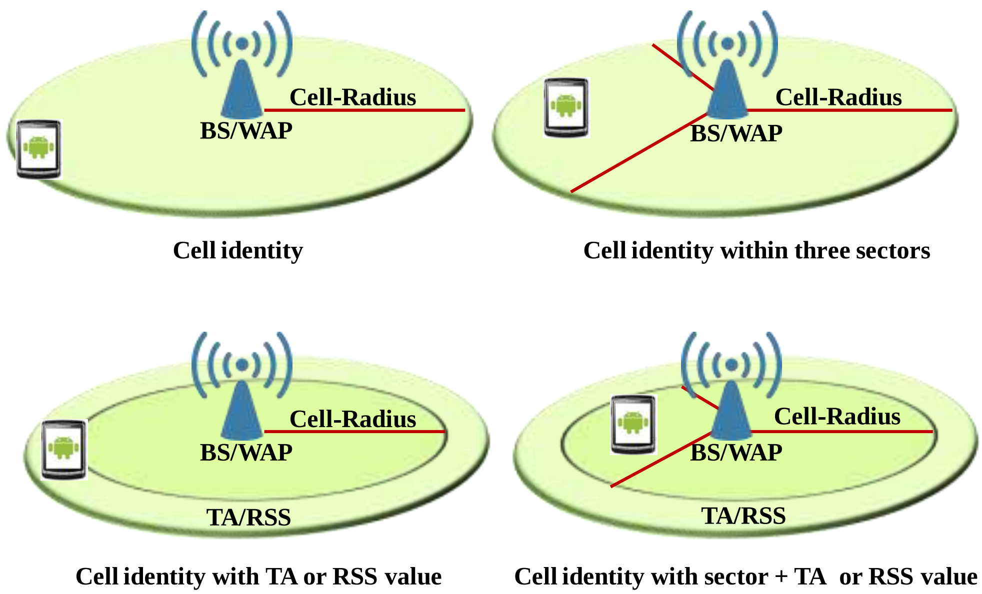


Figure 2-5: Proximity technique.

2.2.2 Angle-Of-Arrival Technique

In AOA technique, pseudoranges and location are found by performing triangulation process [38]. Using triangulation, the location of an SP can be defined when the angles of the received signals from the SP by two or more BSs/WAPs and the position the

BSs/WAPs are known. Such position definition for 2D coordinates is expressed in equation (2.3):

$$(x_i - x_{sp})\sin(\theta_i) = (y_i - y_{sp})\cos(\theta_i) \qquad (2.3)$$

Where x_i *and* y_i are XY coordinate values of BS/WAPs positions, $\theta_i$is the arrival angles for the received WAPs signals and x_{sp} & y_{sp} are XY coordinate values of the smartphone location.

Figure 2-6 shows the geometrical view of triangulation, in multilateral system, to calculate XY coordinates of an SP's location. Note: in multilateral, angle measurements and the calculation of an SP position are taken by BSs/WAPs, i.e. the SP will be a transmitter and BSs/WAPs will be receivers.

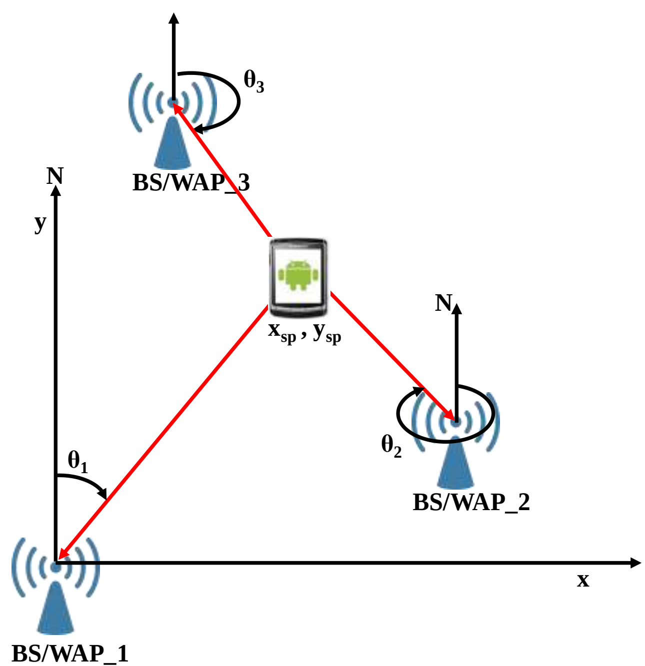


Figure 2-6: AOA technique with three angle measurements.

To be more specific, SP location determination from numerous distance measurements is known as lateration, while angulations allude to the use of angle or heading measurements respect to reference points of known position. However, the main factors that produce angles-measuring errors are:

1. Varying of SNR,
2. Modulation technique of the received signals,
3. The SP movement,
4. Reflecting surfaces near the LOS path of the received signals.

Also, technically, AOA technique needs to deploy array-antennas to find out the angles of the received signals. Practically, due to requiring these special antennas and then incurs large cost, this technique is rarely applicable to locate SPs and it remains a considerable challenge.

2.2.3 Time-Based Technique

Time-based localisation techniques measure signal's propagation time to estimate pseudoranges between an SP and multiple BSs/WAPs. These estimated pseudoranges together with known BSs/WAPs position then can be used to define the SP location. TOA, TDOA, and RTT are the common techniques for pseudoranges estimation [39]. However, several factors are existing which extremely influence on these techniques and then affect on localisation accuracy. These factors include:

1. NLOS and multipath issue [40],
2. Inaccuracy of existing chipset-clocks on BSs/WAPs [41],
3. Obtaining position information of BSs/WAPs [41],
4. Radio-signal coverage of BSs/WAPs [42],

5. Time-source functions for timestamping [43],

6. Taking time measurements at different network stack layers and OS interrupt handling time delay (this will produce instability measurements) [44].

To mitigate the impact of these factors, practically, statistical/filter processes or some calibration/compensate algorithms are needed to estimate accurate pseudoranges between SP and BSs/WAPs and then to define SP location.

2.2.3.1 Time-Of-Arrival Technique

In TOA, an SP estimates pseudoranges from multiple BSs'/WAPs' signals, first, and then it employs Trilateration process to define SP using the estimated pseudoranges and known BSs'/WAPs' locations. BSs/WAPs share their locations (assumed to be pre-defined positions) with the SP. Note: using Trilateration, a process of geometrically defining the location of an SP, in a manner similar to the triangulation except the angle measurements. For unilateral systems, an SP can estimate the pseudorange by calculating the amount of time that it took for the signal from each BSs/WAPs to arrive at the SP and multiplying this measured time-value by the speed of light (c), as express in equation (2.4).

$$TOF_i = T_i - t_i$$

$$p_i = TOF_i * c \qquad (2.4)$$

Where p_i is the estimated pseudoranges between SPs and BSs/WAPs, TOF_i is the calculated propagated time of the received BSs/WAPs signals, T_i & t_i are the received and transmitted time of the signals, and c is again speed of light.

In unilateral systems, an SP will be a receiver while BSs/WPAPs will be transmitters and the whole process of position calculation will be done on the SP, as illustrated in Figure 2-7.

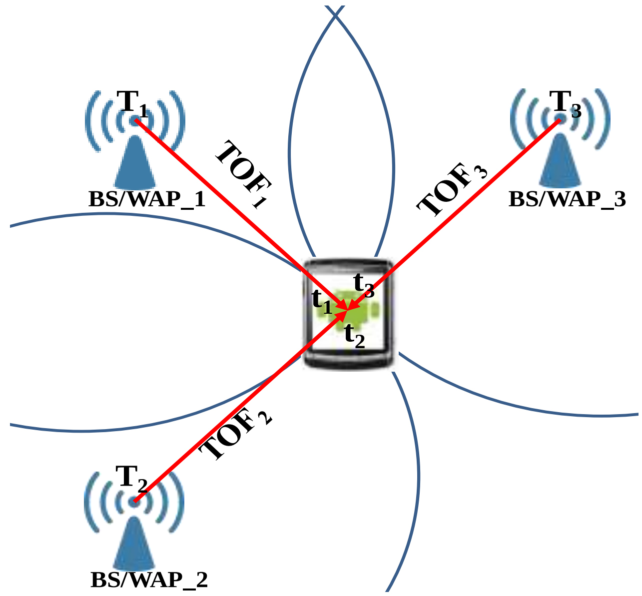


Figure 2-7: TOA technique for SP's location determinations.

Geometrically, through the process of Trilateration, the SP knows its locations at the intersection of a set of circles, when the estimated pseudoranges represent the radius of the circles as well as the position of BSs/WAPs are used as the centre of the circles. For more detail, the location of SP is known to be within the area around a circle. Two circles intersect to produce two points (locations), i.e. now the SP knows its location at the two intersection points (two locations). In order to avoid this ambiguity, i.e. which location is the SP's location, these circles should intersect with another circle (i.e. one extra BS/AP). Or, for example in GNSS systems, one of the locations can most often be ignored as not feasible (because it is in space or in the middle of the Earth).

However, the estimated pseudoranges between the SP and the BSs/WAPs or satellite have an unknown bias error, because the SP clock usually is different from the BS's/WAP's clocks or satellites' clocks. Therefore, a clock-time synchronisation algorithm is required between the SP and BSs/WAPs. Note: this issue has been investigated and solved in **section 3.3** when GNSS time on outdoor-SPs is used to synch WAPs. In order to resolve this bias error one more reference point is required. So generally, three reference points for 2D and four reference points for 3D are needed when the TOA applied to define SPs' positions.

The basic equations for TOA to define the SP position based on Trilateration will be expressed by equations (2.5) and (2.6), for 2D and 3D respectively.

$$p_i = \sqrt{(x_i - x_{sp})^2 + (y_i - y_{sp})^2} + b_{sp} \tag{2.5}$$

$$p_i = \sqrt{(x_i - x_{sp})^2 + (y_i - y_{sp})^2 + (z_i - z_{sp})^2} + b_{sp} \tag{2.6}$$

Where p_i is the estimated pseudoranges, x_{sp}, y_{sp}, z_{sp} are unknown XYZ coordinates of SP position, x_i, y_i, z_i are known XYZ coordinates of BSs/WAPs positions, b_{sp} is the clock bias error between the SP & BSs/WAPs and *i* is number of reference positions (BSs/WAPs positions).

However, practically it is difficult to implement Trilateration process on SPs, this is because: 1) the equations are nonlinear, 2) and SP location accuracy can be further improved by using a larger number of BSs/WAPs than the minimum required. To solve this difficulty, the equations can be linearised by using an iterative method such as LLS [45]. For example, mathematically to linearise equation (2.6) when *i* is more than 4, the result will be expressed by equation (2.7):

$$\begin{bmatrix} \delta p_1 \\ \delta p_2 \\ \delta p_3 \\ \delta p_4 \\ \vdots \\ \delta p_i \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & 1 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 1 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 1 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{i1} & \alpha_{i2} & \alpha_{i3} & 1 \end{bmatrix} \begin{bmatrix} \delta x_{sp} \\ \delta y_{sp} \\ \delta z_{sp} \\ \delta b_{sp} \end{bmatrix} \tag{2.7}$$

$$\text{Where } \alpha_{i1} = \frac{x_i - x_{sp}}{p_i - b_{sp}} \qquad \alpha_{i2} = \frac{y_i - y_{sp}}{p_i - b_{sp}} \qquad \alpha_{i3} = \frac{z_i - z_{sp}}{p_{i-b_{sp}}} \tag{2.8}$$

Equation (2.7) can be re-written in a simplified form as [45]:

$$\delta p = \alpha \, \delta x \tag{2.9}$$

Where δx and δp are the vectors and α is the matrix. Since α is not a square matrix as well as it cannot be inverted directly. Equation (2.9) is still a linear equation. If there are more equations than unknowns in a set of linear equations, the least-squares approach can be used to find the solutions. The pseudo-inverse of the α can be used to obtain the solution. The solution can be expressed as equation (2.10) [45]:

$$\delta x = [\alpha^T \alpha]^{-1} \alpha^T \delta p \qquad (2.10)$$

Where T is representing the transpose of the matrix and $[\,]^{-1}$ represents the inverse matrix. By implementing equation (2.10), the values of δx_{sp}, δy_{sp}, δz_{sp}, and δb_{sp}can be found. In general, the LLS method gives a better solution than the position when obtained by only data from four BSs/WAPs, since more measurements are used.

2.2.3.2 Time Difference of Arrival Technique

TDOA calculates SP location from differences of the measured arrival times on pairs of received BSs'/WAPs' signals, as expressed in equation (2.11):

$$\begin{aligned} \Delta t_{12} &= t_1 - t_2 \\ \Delta t_{13} &= t_1 - t_3 \\ \Delta t_{23} &= t_2 - t_3 \\ \Delta p_{ij} &= \Delta t_{ij} * c \qquad (2.11) \end{aligned}$$

Where t_i is the time measured of the received BSs/WAPs signals, Δt_{ij} is the differences of the two received of BSs/WAPs signals, Δp_{ij} is the estimated difference of the pseudoranges and c is the speed of light.

TDOA calculations then employs a hyperbola process as the possible SP position, as shown in Figure 2-8.

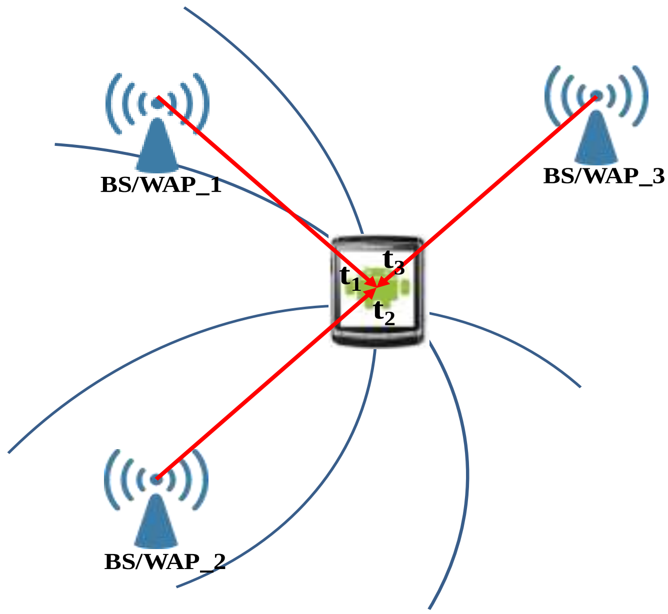


Figure 2-8: TDOA technique for SP's location determinations.

At least, one extra BS/WAP is required for TDOA per dimension compared to TOA. Similar to TOA, the basic equations to solve TDOA calculation is expressed in (2.12):

$$\Delta p_{ij} = \sqrt{(x_i - x_{sp})^2 + (y_i - y_{sp})^2 + (z_i - z_{sp})^2} - \sqrt{(x_j - x_{sp})^2 + (y_j - y_{sp})^2 + (z_j - z_{sp})^2} \qquad (2.12)$$

Where: Δp_{ij} is the estimated difference of pseudoranges from any pair of BSs/WAPs. The same aforementioned process of linearisation for equation (2.10) could be performed to obtain further SP position accuracy, when there are more than four BSs/WAPs are available.

For TDOA, like TOA, the clock synchronisation algorithm is required, but only between BSs'/WAPs' clocks. A practical way to do this clock synchronisation is to use signal transmission between SPs and BSs/WAPs. Beacon signals is a proper one, since it is a continuous or periodic transmission that facilitates timing synchronisation or position measurements between the SPs and BSs/WAPs. However, for localisation purpose, wireless devices' clocks such as WAPs and cellular BSs clocks are cheap & inaccurate

[44]. Note: one *μsec* in time error is equivalent to 300 meter in position error. Therefore, the relevant research community needs further investigations and it has been concluded that high quality reference-time in nanosecond resolution is needed to synchronise such clocks.

2.2.3.3 Round Trip Time Technique

RTT works similar to TOA in terms of SP position calculation, except that this technique measures the TOFs of signals travelling from an SP to the BSs/WAPs and back to the SP [44] as expressed in equation (2.13).

$$RTT_i = (t_{i2} - t_{i1}) + (t_{i4} - t_{i3})$$

$$TOF_i = (RTT_i/2) - \Delta t_i$$

$$p_i = TOF_i * c \qquad (2.13)$$

Where RTT_i is the estimated round trip time for each received BSs/WAPs signals, t_{i1}& t_{i4} are measured time of the transmitted and received signals (via the SP) respectively, while t_{i2}& t_{i3} are measured time of the received and transmitted signals via BSs/WAPs respectively, Δt_i is the delay time of the packets/signals processing through receiving and transmitting signals, and *c* is the speed of light. To calculate SP location, RTT techniques employs Trilateration process, as shown in Figure 2-9.

In TOA, calculating the delay is by using both SPs and BSs/WAPs clocks, while in RTT, it uses only the clock of the SP to record the signal transmitting and arrival times. Because of this advantage, this technology solves the problem of synchronisation to some extent. RTT, like TOA, applies Trilateration to calculate the SP. However, one of the drawbacks of RTT is to estimate the pseudorange from multiple BSs/WAPs that need to be carried out consecutively which may cause huge latencies for applications where the SPs move quickly. In addition, this technique makes huge traffic-load on the network due to exchange large number of frames between the mobile device and the BSs/WAPs.

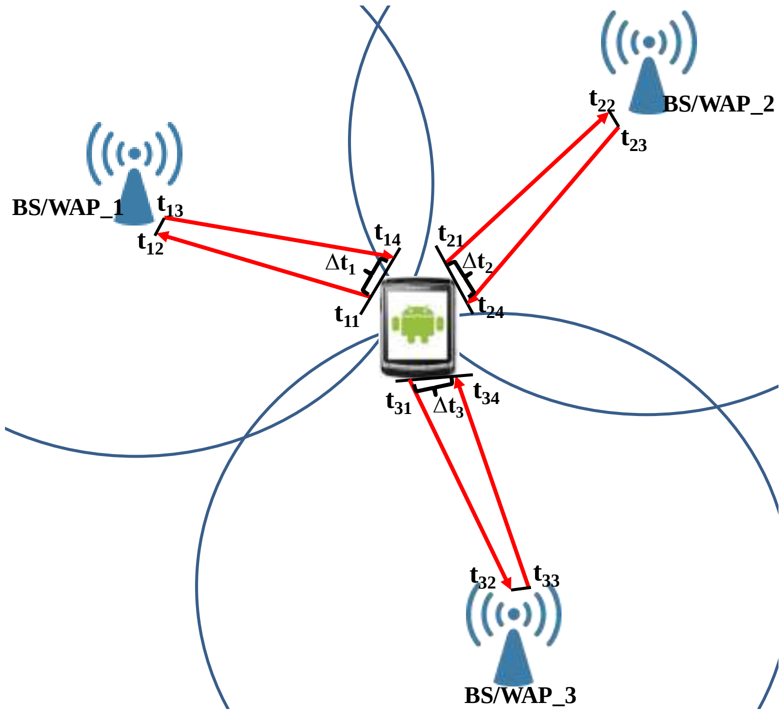


Figure 2-9: RTT for SP's location determinations.

2.2.4 Received Singal Strength-Based Techniques

To estimate SP's location based on RSS measurements, two techniques have been used:

1. Pseudoranges-based technique [33]: This technique is based on known radio propagation analytic relationship. It also employs Trilateration process to find SP location from the estimated pseudoranges (based on computed RSS values) between an SP and multiple BSs/WAPs, as it can be observed in Figure 2-10.

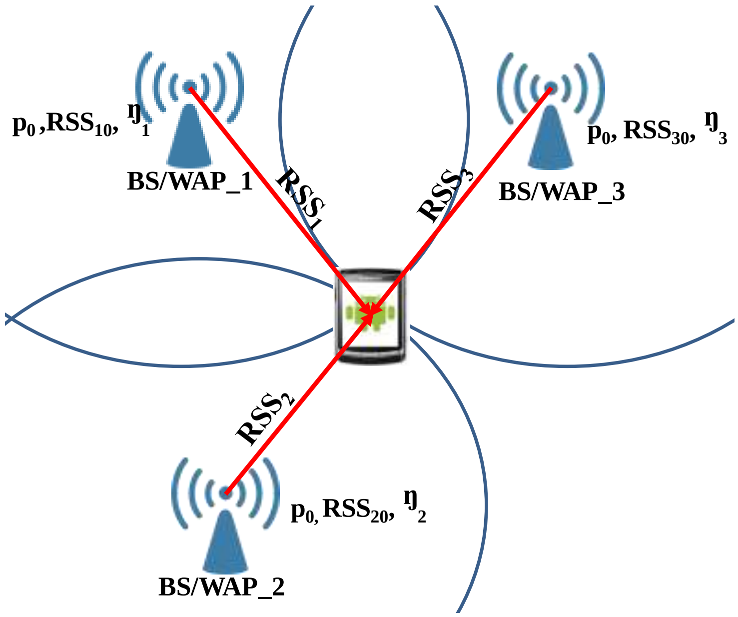


Figure 2-10: RSS-Radio propagation technique.

To estimate the pseudorange between the SPs and BSs/WAPs, equation (2.14) should be utilised.

$$p_i = p_0 * 10^{\left(\frac{RSS_{i0} - RSS_i}{10 * \eta_i}\right)} \qquad (2.14)$$

Where p_i is the pseudorange between smartphones and BSs/WAPs, p_0 is the estimated calibrated pseudorange at zero distance, RSS_{i0} measured signal stength value for the p_0 , RSS_i is the measured signal strength for the received BSs/WAPs signals, and η_i is the calculated/calibrated path loss exponent for the received BSs/WAPs signals.

Practically estimating the pre-defined path loss exponent, signal propagation parameters, and environmental conditional are the main challenges to measure the pseudoranges between the SP and the BSs. In addition, inaccuracy of measuring RSS values in a localisation solution is due to: HW implementation approximately ±4dBm varies, mathematical methods to calculate the RSS values, other working systems in the same band (interference issue), moving objects (human moving) in buildings and fixed and/or movable obstacles. Certainly, many dynamic models have been proposed to mitigate these inaccuracies, such as in [46], which are more accurate than the traditional static model. But still these models are not appropriate for most LBS applications due to obtained low position accuracy, for example within ±5 meters accuracy in a small coverage [47].

2. The second technique, RSS-fingerprinting, is based on searching for pre-stored RSS values of BSs/WAPs in a database. The location estimation process in fingerprinting technique consists of two stages: offline stage and online stage. These stages with their localisation process are displayed in Figure 2-11.

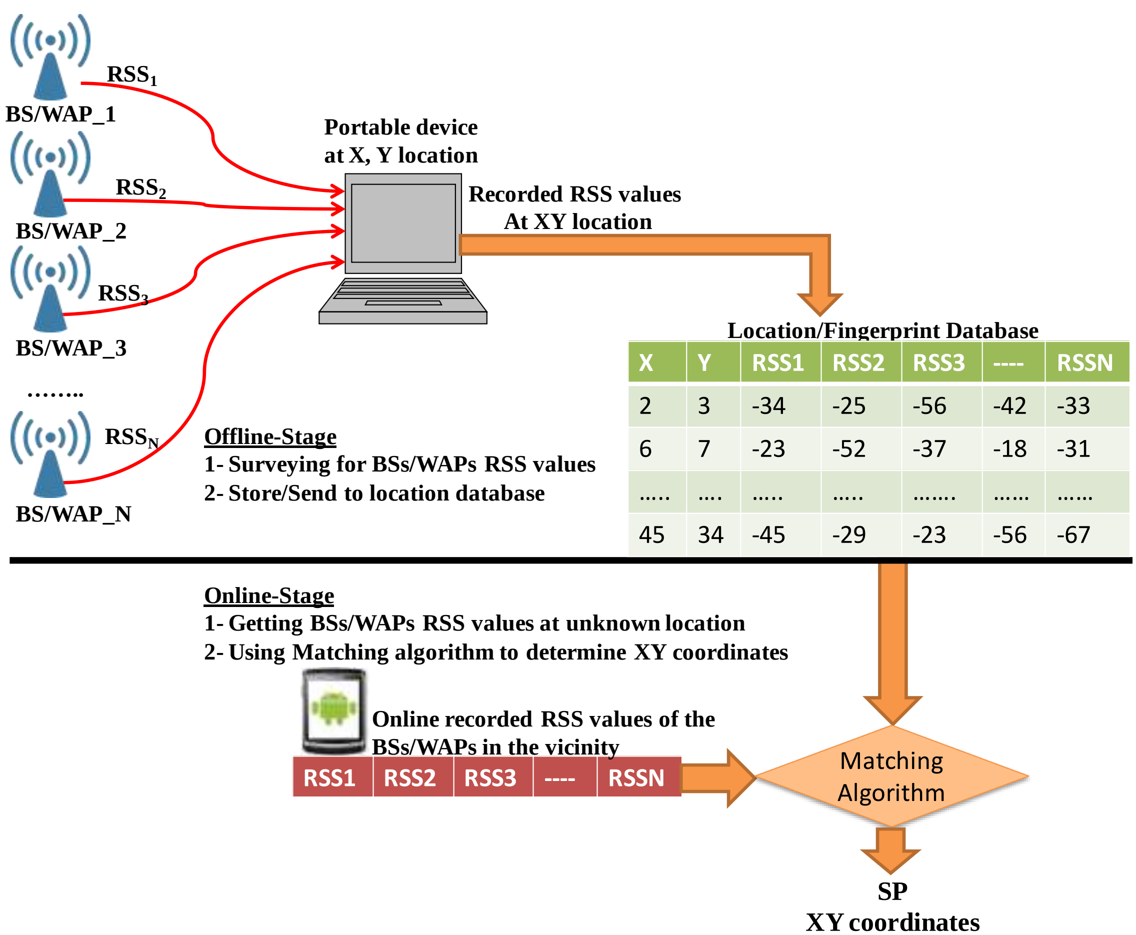


Figure 2-11: SP's location determinations via RSS-Fingerprinting technique.

In the offline stage, a radio map or a database for signals' strength of the major BSs/WAPs in different reference points, i.e. surveyed points, around an area should be recorded. Therefore, the accuracy of this technique depends on the number of reference points, for example large number of reference points provides better location accuracy or vice-versa.

Then in online stage, a matching process between real-time RSS and the recorded of the pre-defined radio map is involved to estimate SP's location.

The main advantage of this technique is that it is based on actual path loss at points near the SP. Thus, unknown factors of multipath and shadowing are bypass and affect only minimally on the SP location estimation. However, practically, this technique has many challenges such as: this technique is for a specific building or site-dependent, takes a long time due to connect with the Internet and searching in the location database/server and then sends back the result of location calculation for the users, this technique deduces huge cost to make the radio map and maintaining the radio-map for dynamic structure and movable-objects is also needed. Therefore, using and improving this technique for SP localisation remains as an open research area.

2.2.5 Map-Matching Technique

This technique is based on the theory of machine learning algorithms, pattern recognition/matching, which combines map with the measured SP's location observations to obtain the real position of SPs in 2D or 3D coordinates. The use of maps is an efficient alternative to the installation of extra HW.

Many solutions on the SPs are available to utilise this technique such as GNSS, SLAM solution, and WiFi-SLAM solution. This technique could be combined with other localisation techniques including dead reckoning, WiFi RSS-based, and time-based technique. Actually, this technique is mostly used in order to increase the accuracy of the localisation solutions [17]. However, this technique needs smart-buildings. That is, this technique maintains huge knowledge of the buildings' layout as well as takes a huge amount of memory & incurs high battery-power consuming when a complex map-computation algorithm is running on the SPs.

2.2.6 Dead Reckoning Technique

This localisation technique is based on utilising onboard SPs inertial sensors including gyroscope, accelerometer, and magnetometer sensors. DR estimates distance-displacement (i.e. step length) and heading to compute current SP position based on a previously known position, as it can be observed in Figure 2-12. That is, at the beginning it needs an initial reference position. The DR technique uses: gyroscope for estimating the heading and accelerometer sensor for distance estimation.

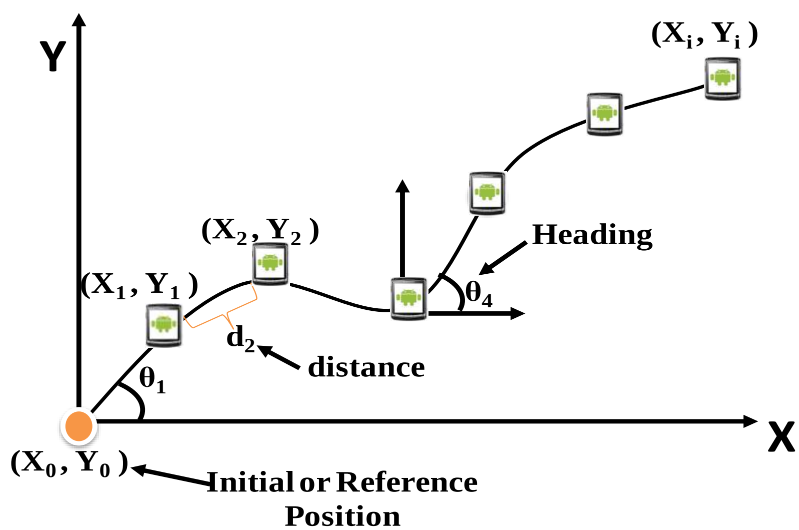


Figure 2-12: SP location calculation based on DR technique.

To calculate smartphone position based on DR technique, equation (2.15) should be utilised.

$$X_i = X_{i-1} + d_i * cos(\theta_i)$$

$$Y_i = Y_{i-1} + d_i * sin(\theta_i) \qquad (2.15)$$

Where X_i & Y_i the estimated XY are coordinate values of SP's position, X_{i-1} & Y_{i-1} are the previous determined XY coordinate position, d_i is the calculated distance using the accelerometer measurements and θ_i is the estimated heading of the SP via gyroscope measurements.

DR technique is highly smooth and stable, but its performance degrades quickly over few seconds due to the accumulated measurement noise of sensors causing cumulative positioning error [32]. Therefore, this technique to define SPs as well as due to sensors noise and drift, it needs to calibrate the sensors periodically based on some known location information such as GNSS heading, position and velocity. Figure 2-13 shows a typical SP's DR prototype model to compensate and to reduce both drift and sensor noise using dedicated filtering algorithms, for example Kalman Filter [48]. The figure also shows how the model utilises inertial sensors to estimate both distance and heading and then how use these measurement to calculate SP's location.

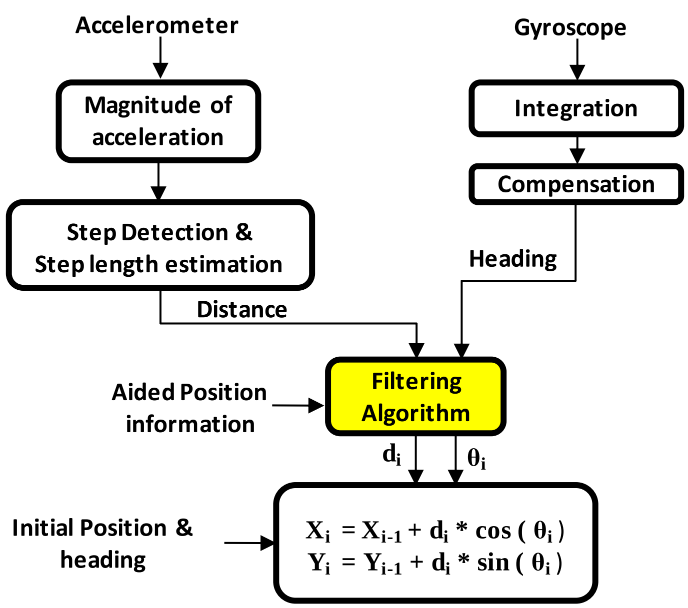


Figure 2-13: A DR-prototype model for SP localisation solutions.

2.2.7 Combined-Localisation Techniques

In order to improve the location accuracy, to reduce measurement records, short time to locate SPs, and consequently to reduce battery-power consumption, a combined localisation techniques is needed [49].

The combination technique is not only to make powerful localisation solution, but it is also to reduce the number of reference positions to involve the SP location estimation. For example, combining TOA and DR techniques has been used to hybrid the distance-displacement and the heading of the SP only with a single WAP [50]. In this hybrid approach TOA is used to measure the pseudorange between the WAP and the SP in two different locations and it uses DR to estimate the heading & the distance-displacement between the two different locations. The combined technique, as it is shown in Figure 2-14, has the following achievements:

1. Obtains better accuracy than DR, when it is used as a standalone technique

2. Needs only a single WAP to contribute SP location calculation in comparison with TOA technique alone that needs three WAPs as reference positions.

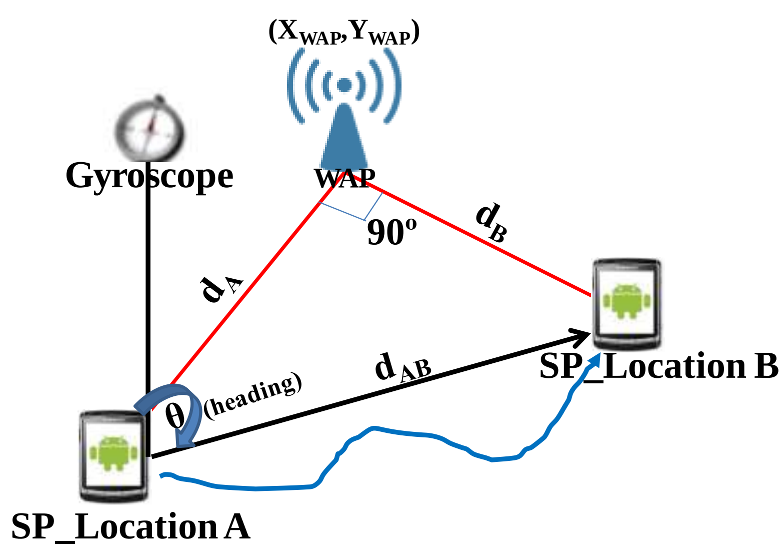


Figure 2-14: The right-triangle formed between a WAP and an SP from location A to B.

2.3 Onboard SPs Technologies for Localisation

This section addresses the second research question (**as mentioned in section 1.4**): what are the main error sources of onboard SP wireless/sensor technologies for localisation? Also in the below subsections, the question of "What is the ability of each technology?" will be addressed. The increasing technologies such as: GNSS receivers as well as LTE, NFC, WiFi, BT transceivers and inertial sensors on SPs makes possible to more powerful positioning with SPs in different circumstances. Figure 2-15 illustrates these technologies in standalone and in combined hybrid mode. In fact, some of these technologies are not originally intended for positioning functionality such as: Cellular, WiFi, and BT. But, the reading from of transmitted/received radio signals of these technologies can be utilised for localisation purposes. In addition, each individual technology has its own advantages and limitations in terms of availability and robustness. Therefore, this section demonstrates the ability and the imperfections of the localisation measurements via these technologies.

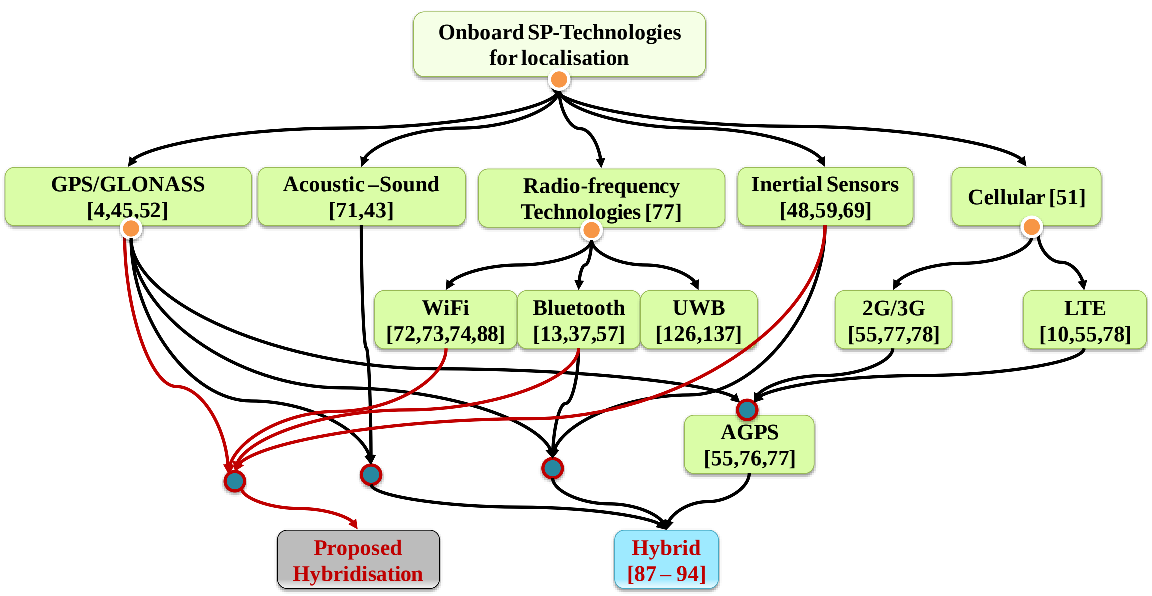


Figure 2-15: Taxonomy of technologies for SP localisation, including refernces.

2.3.1 Cellular Technologies

Cellular technologies rely on a group of BSs, with the radio coverage from a few meters to about tens of kilometres. Historically, the localisation solutions of Mobile handsets were focused on achieving the requirements of the FCC-E911 mandates, and they were cellular-network based. Cell-ID, AOA, TOA, TDOA, and E-OTD were some of the localisation techniques deployed by some of the cellular networks at the time [51]. Nowadays, with the emergence of the LTE technology, since release 9, a new protocol, known as SUPL has been included to offer secure SP positioning.

However, the obtained accuracy by cellular networks using above techniques is often low, in the range of 20–200 m. The accuracy depends on the cell size and pseudorange measurements between the SPs and the BSs. Generally, the accuracy is higher in densely covered areas, for example urban places, and much lower in rural environments. For indoors, SP positioning based on cellular technologies is conceivable if the buildings is covered by several BSs or a single BS with strong signal received by SPs.

2.3.2 GNSS Technology

The GNSS receivers, such as GPS and GLONASS which are integrated on SPs, are extensively used to obtain the SP position, when outdoors. These systems provide accurate, continuous and world-wide, three dimension position, and velocity information to users with appropriate receiving equipments.

Taking GPS as an example, the GPS satellite constellation nominally maintains of at least 24 satellites, 95% of the time, arranged in 6 orbital planes with 4 satellites per plane. The satellites broadcast ranges codes and navigation data (ephemeris and Almanac data) on two frequencies using a technique called CDMA. The two frequencies are L1 (1,575.42 MHz) and L2 (1,227.6 MHz). GPS uses the concept of TOA pseudoranging and Trilateration to determine SP position [45].

Pseudoranging code enables the SP's receiver to determine transit time (propagation time) of the signal thereby determines the satellite-to-SP pseudorange. Navigation data provides the means for the receiver to determine the location of the satellite at the time of signal transmission. GPS receiver in a 3D mode three satellites and three distances are needed. The equal-distance trace to a fixed point is a sphere in a 3D case. Two spheres intersect to make a circle. This circle intersects another sphere to produce two points. In order to determine which point is the user position, one of the points is close to the earth's surface and the other one is in space. Since the user position is usually close to the surface of the earth, it can be uniquely determined.

However, the distance measured between the receiver and the satellite has a constant unknown bias, because the SP clock usually is different from the satellites' clocks. In order to resolve this bias error one more satellite is required. Therefore, in order to find the SP position four satellites are needed. Despite the position error due to the clock time error, there are several other error sources which are affected on location accuracy such as: selective availability, DOP issue, ionospheric delays, tropospheric delays, multipath and receiver noise [52].

GNSS receivers on modern SPs have been developed with improving performance and enhancing its accuracy as well as they are able to run positioning services in much more severe signal-degraded area than before [53]. Note, to show the accuracy of the estimated SP position via GNSS technology, set of trial experiments have been performed during research study (**as it is included in section 5.4**). Following these improvements of GNSS-based positioning services for outdoors application, however, the demand and challenges now have shifted to the provision of such services for the dense urban and/or indoors (warehouses and buildings).

Several attempts to enhance this technology have been succeeded by adjusting new infrastructures including Pseudolite [54], Locata [55] and IMES [56] for indoors applications. However, GNSS ability to locate SPs indoors remains a substantial challenge, forming the major challenges to prevent accurate positioning seamlessly from outdoors to indoors due to cost of deployed extra hardware for these solutions. The detailed explanation of these solutions is included in **section 2.4.2**.

2.3.3 WiFi Technology

The WiFi transceivers integrated on SPs are not only for data communication, but they can also be used to estimate SPs position. Mainly, the SP LBS applications use this technology to estimate the position of the user inside buildings, where the WAPs signals are available. For example, an SP can calculate the TOF of beacon signals coming from each WAPs distinguished through its MAC address, and assuming these WAPs' position are previously known. Based on these observations, the SP can perform a localisation technique dynamically to report an estimation of the SP position. Specifically for WiFi time-based localisation solution, however, due to existing inaccuracy clock (clock drift and clock offset) for timing/TOF measurements on WAPs and onboard SP WiFi transceivers, pseudorange estimation will not be accurate [50]. Note: trial experiments to demonstrate WAP-SP clock accuracy using beacon signals via SPs are included in **section 3.2**.

Regardless of this inaccuracy clocks on WAPs and WiFi transceiver, accurate SPs-WAPs clock synchronisation within few nanoseconds, small coverage of WAPs in the building, obtaining WAPs positions information and no WAPs localisation algorithm are some other challenges to design SP-positioning solution with reliable accuracy.

2.3.4 Bluetooth Technology

BT has developed as a practical choice of indoors-SP localisation solutions and several indoor positioning systems relying on this technology [57]. This is mainly because it has emerged as a low cost, low power consumption and bigger coverage range than traditional/classical BT classes.

The recent developed localisation solution based on BT is BLE-iBeaconing. With BLE, all its needed is to drop a set of BT-anchors around the area and then SPs based on RSS measurements can detect these anchors. In this way, a localisation solution using these measurements can successfully track SPs location. The main feature of BLE is that permits us to supply just enough contexts, while still being able to move quickly and easily. This peer-to-peer messaging opens up numerous potential outcomes, extending from LBS applications in shopping centres to emergency reaction circumstances. However, this research work found out that RSS measurement have signal propagation issues which causes huge location error [58]. Therefore any pseudorange measurements and/or location estimation based on RSS-techniques will not be accurate & reliable [59].

2.3.5 Inertial Sensors

Embedded inertial sensors on SP only give a relative location estimate with accuracy degrading over short run; therefore, they could be utilised together with other technologies including GNSS, WiFi, and BT to estimate absolute location and to get better accuracy. Basically, an SP can read measurements from these sensors to locate users by performing DR technique. Accelerometer sensor to measure change of velocity (acceleration force), magnetometer sensor to measure magnetic field, and gyroscope to measure change of angles are the main inertial sensors that can used for SP positioning. However, accelerometer and magnetometer measurements are affected by sensors noises and interference issue (especially for indoors), while gyroscope measurements suffer from huge drift over few seconds to estimate the heading [60].Actual gyroscope drift and accelerometer noise measurements on Android-based SPs during SP position estimation based on several trials are showed in **section 5.5**.

2.4 SPs Localisation Solutions

This section addresses the third research question, (see chapter 1, **section 1.4** and it is mentioned in the beginning of this chapter): do the existing SP localisation solutions based on particular localisation technique offer seamless outdoors-indoors positioning service? LBSs on SPs adopt several solutions to ensure that location is achieved accurately and continuously. We adopted the following criteria to classify such solutions:

1. Environments (outdoors and indoors)
2. Standalone and hybrid solutions
3. Satellite and terrestrial
4. Unilateral and multilateral

In this section, we attempt to classify the current trials & the improved localisation solutions into: outdoors, indoors, and seamless outdoors-indoors. Additionally, practical challenges for implementing of the solutions and for new available commercial solutions are discussed.

2.4.1 Outdoors Localisation Solutions

Cellular networks and GNSS technologies are candidate solutions for SP outdoors localisation [34]. The GNSS receiver onboard the SP can define its location within few meters. However, GNSS receiver consumes more power, provides inaccurate location, and takes long time to fix the SP, when indoors or in high-dense urban areas, due to the availability of the GNSS weak signal and multipath issue [61].

Another factor of GNSS inaccuracy or losing GNSS signal tracking is due to GNSS jamming/interference [62]. Vulnerable of GNSS signals from interference sources is due to received low GNSS signal strength. The interference sources do not necessarily need to be centred at the same frequency as the GNSS signals.

The promise of alternative solution for such cases is to use cellular network signals for positioning, as a GNSS backup solution or aid GNSS such as AGPS [63]. An example of AGPS architecture is shown in Figure 2-16.

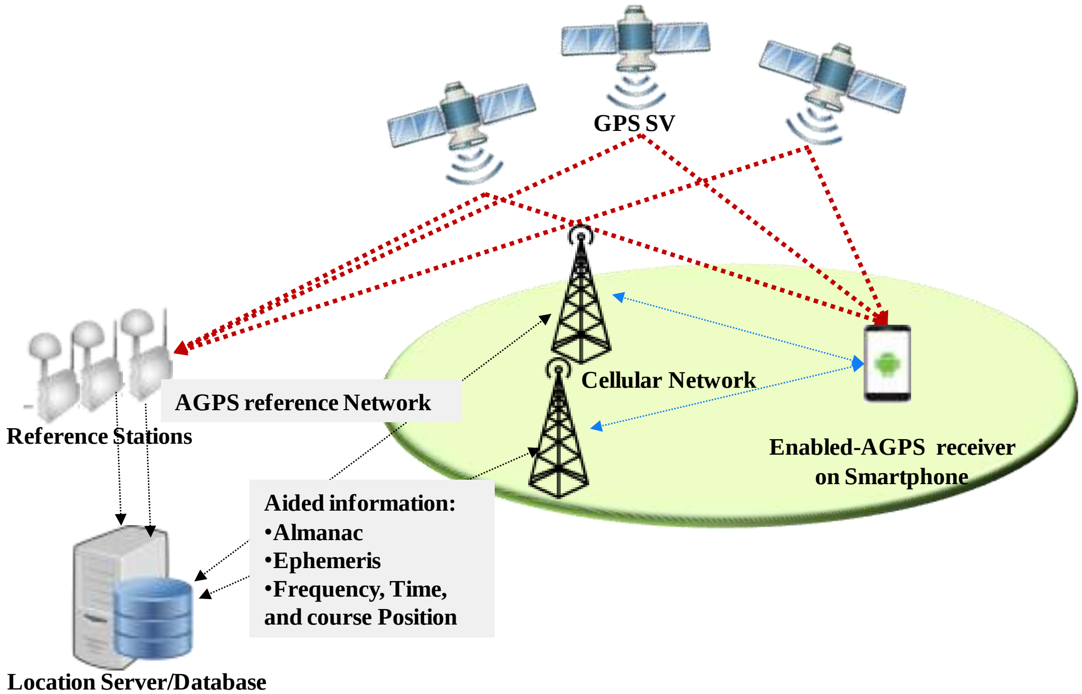


Figure 2-16: AGPS overview-system representation.

In addition, several solutions have been proposed to locate the SPs through using only cellular signals based on different techniques. For example: Cell-ID, RSS-based, AOA, TDOA, OTDOA, U-TDOA, and E-OTD [34]. Furthermore, these solutions could be classified into two major types of localisation solutions: network-based solutions and handset-based solutions.

Both localisation solutions have different capabilities in terms of privacy, software/hardware upgrading, accuracy, and battery-power consumption. These capabilities and performance parameters are evaluated and explained in Table 2-2.

In addition, these solutions are utilised as indoors or urbane localisation solutions. However, since most of these solutions' accuracy is within tens of meters, as well as they are customised with special hardware and take a huge cost [64]. Therefore, they are not suitable for most current SPs LBS applications [65].

Table 2-2: Handset-based and network-based localisation solutions comparison **[66]**.

Handset-based (SP) location solution	**Network-based location solution**
It is more secure	It is less secure than Handset-Based
It doesn't affect the network capacity	It uses facilities and resources of the network
It is more accurate for location; it is not limited by the network to the number of measurements	It depends on the requiring measurements to be improved for location accuracy
It needs special SW and HW that must be incorporated together	It doesn't require upgrading SW for the SPs' devices. Most legacy phone-handset can receive services
It consumes the SP's battery power to carry out the positioning task	It frees the SP of the power battery
It participates in the positioning task, or the calculation is done by itself	Network performs the positioning task without intervention by the SP
It is known as self-positioning solution.	It is known as remote positioning solution.

There is a trade-off performance evaluation for these solutions for example: if one solution has a good accuracy then it will take long time to fix and consequently consume more battery power such as with the GPS and AGPS. In comparison, Cell-ID and Cell-ID + TA take short time to fix and low power consumption but have low accuracy. Additionally, some of these solutions could not be applicable transparently, due to having huge costs and function limitations such as U-TDOA and AOA, respectively.

Since the deployments of WLAN infrastructures based on WiFi technology standards are widely adopted in urban area, WiFi technology has been employed for such area as an alternative localisation solution [67]. Especially, when the cellular solutions are not accurate enough, or they are not applicable [68]. However, in these situations WiFi-based solutions do not perform well due to having multipath and NLOS signals which affect SPs' location accuracy.

2.4.2 Indoors Localisation Solutions

There is a huge demand on making reliable indoors positioning solutions, since people spend 80-90% of their time, and 70% of people calls & 80% of their data exchanging are occur when indoors [69]. Recent commercial indoors localisation solutions based on different technologies and techniques with their accuracy are listed in Table 2-3. However, neither high performance nor wide-spread indoors localisation solution is obtainable yet [70]. This is due to wireless technologies limitations and the complexity indoors structure. Although some of these solutions, for example WiFi-SLAM, Skyhook and Ekahau can achieve a reasonable accuracy [71]. But they need to deploy new/additional hardware; or they are using Internet to connect with reference-location database/server in order to calibrate the interest area and then to locate the SPs. Furthermore, some of these solutions are implemented on the SPs, like Sensewhere and Navizon, while some others are in process, i.e., they need more researching & solving practical issue such as PlaceLab, ArrayTrack, and PinPoint.

For indoors-SPs, most of the researches focused on a technology to locate the SPs as they have been located when outdoors and enable navigation, local search, sharing location and other LBS. To achieve these services, several solutions and researches have been proposed. For example, Pseudolite is as an alternative solution for GPS and used as an indoors solution to find the location of SPs in sub meter accuracy [54]. However, it requires deploying ground-based transceivers which incurs huge cost. High quality of time synchronisation, near-far problem, and multipath are the main challenges of the solution to locate the SPs.

Table 2-3: Indoors localisation solutions.

Solution name	Accuracy	Wireless/sensors Technology	Localisation Technique	Overhead	Comments
ArrayTrack [72]	Up to1m	WiFi	AOA	Needs to deploy a new WiFi directional antennas	It is good for LOS signals and for a small coverage
Ekahau [73]	5m-15m	WiFi	RSS	Clients need to calibrate and make the radio map for a specific area	An Internet connection is needed to reference the location-database
Skyhook [73]	10m-20m	WiFi	RSS	Solutions need to calibrate and make a radio map for a specific area	
Navizon [74]	20m-40m	WiFi	RSS		
Place Lab [75]	20m-40m	WiFi	Proximity and RSS		
Sensewhere [76]	Up to10m	WiFi and AGPS	Proximity and RSS	No WAP surveying nor associated database	An Internet connection is needed to reference the location-database
Polaries [77]	100-500m	RF technology	RSS-Fingerprinting	Survey RSS-values for a specific geographical area	
Qualcomm [77]	50-400m	GPS and Cellular	AGPS/AFLT method as a hybridization solution	It doesn't need any additional or tailored hardware during localisation	It depends on the visibility GPS satellite vehicles in sky and cellular network conditions
NextNav [77]	2-4m vertically and 50-150m horizontally	RF-technology	TOA	Needs to deploy a special infrastructure in a geographical area	Special receiver should be connected with the SPs
iBeaconing BLE [37]	1-2 m	BT	Proximity & RSS	Needs to deploy large number of BT-anchors or sensors	The obtained position accuracy depends on number of deployed sensors
U-TDOA (Trueposition) [78]	Up to 50m	Cellular	TDOA	Needs to install LMU on the cell towers	SNR, number of cell towers, timestamp, and transmitter/receiver geometry condition are the main factors on the solution’s accuracy
PinPoint [79]	Up to 7m	WiFi	TOA (two-way measurements) and TDOA	Needs constant number of message exchanges between SP and WAPs or any other nodes	The accuracy is based on the accuracy of the clock rates (e.g. WiFi clock off-the-shelf is 40 MHz is ~ 25 ns). And the coverage in tens of meter.
Goodtry [80]	Up to 4m	WiFi	TOA (four way measurements)		
WiFi-SLAM [81]	3m-5m	WiFi and Map	Mapping and RSS	Needs to upload the map of the buildings/area and calibrates WAPs’ signals parameters	They need an internet connection to communicate with the system’s location servers
GraphSLAM [81]	4m-7m	MAP and Sensors	Mapping and dead reckoning	Needs to upload the map of the buildings/area and calibrates sensors	
Pseudolite [54]	Sub-meter	Terrestrial replica of GNSS	TOA	They need to deploy new transmitters for (IMS) and transceivers for (Locata and Pseudolite)	They are GPS-like technology.
IMES [56]	Up to 10m		Proximity		
Locata [55]	Sub-meter		TOA		
Proposed solution	Up to 2m or 3m	GNSS, BT, WiFi and inertial sensors	Combined TOA (using GPS time to synch SP-WAPs clocks) & DR measurements	No need to: deploy new hardware, war-driving and radio mapping.	It is not site dependent, does not need Internet, and it is on-the-go, anywhere, anytime.

In a variety of localisation solutions, an IMES and GPS receiver to provide indoors positioning solution has been proposed [56]. The architecture of the IMES can be seen in Figure 2-17. SPs consume low battery power when IMES is used. However, obtained SPs location performance by using this solution doesn't meet the LBS user's requirement, since IMES is based on proximity technique and it offers limited SP location accuracy. In addition for that, practically, a GPS receiver firmware modification is needed to implement IMES on the SPs.

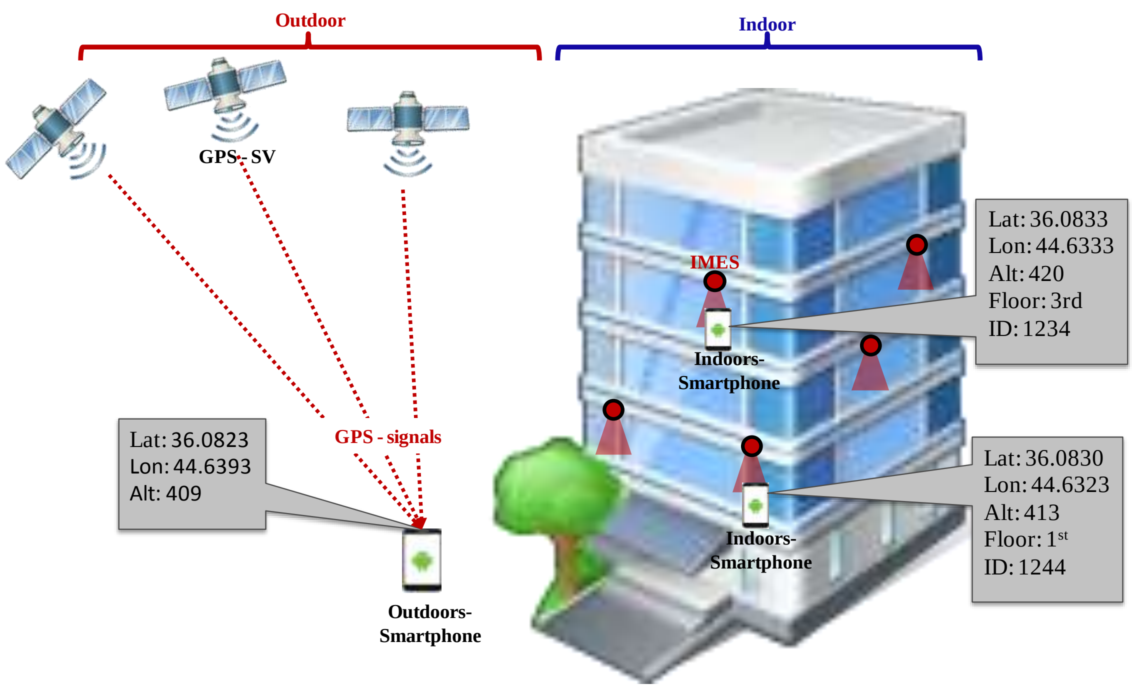


Figure 2-17: Outdoors and indoors positioning using IMES and GPS.

Locata system is another indoors solution [55]; it is able to replicate GPS/GNSS performance indoors, as it shown in Figure 2-18.

Locata is GPS-like solution; it needs four transmitted signals to locate the SP as well as needs high quality clock synchronisation to calculate accurate pseudoranges between the SPs and Locata-transmitters. All requirements and capabilities for IMES and Locata solutions are listed in Table 2-4.

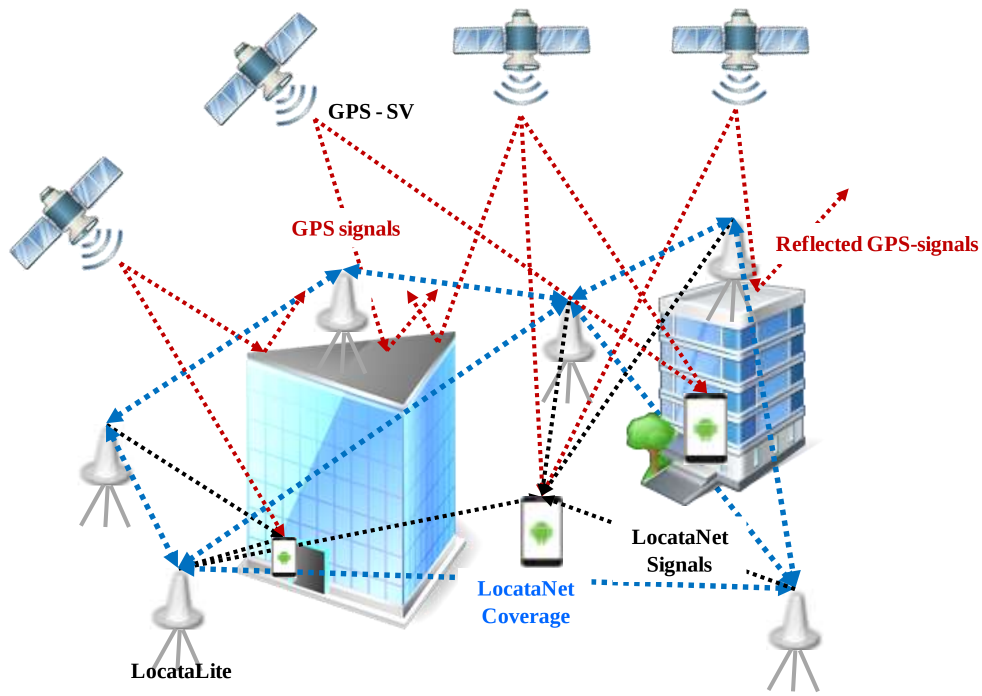


Figure 2-18: Locata positioning concept.

Table 2-4: Comparison between Locata and IMES solutions.

IMES	Locata Solution
It does not need any synchronisation.	Strong time synchronised ranging signals are needed.
It operates in GPS L1 band (with offset 8.2 kHz).	It operates in the ISM band 2.4–2.4835 GHz.
It does not need any HW modification.	Needs equipped Locata receiver in SPs.
The accuracy is up to 10 meters.	Position accuracy is in cm-level.
Application in deep indoors shopping builds and underground.	Applications are in open-cut mines, urban and even indoors locations.
Need a single transmitted signal & message to locate the SPs.	Needs four transmitted signals to locate the SP.

The main unique drawback of Pseudolite, Locata, and IMES is to establish new infrastructure to cover indoors-SPs for LBS application which is incur huge cost.

Although WiFi technology is not planned/deployed for the purpose of localisation, but measuring WAPs signal-parameters provide the possibility of locating indoors-SPs [82]. WiFi technology based on some calibration conditions & predefined WAPs positions information shows better SP positioning accuracy when other localisation technologies onboard SPs cannot be utilised. Many localisation techniques such as: RSS-based, proximity, and time-based localisation are likely being used to locate SPs based on WiFi technology. However, due to having a lot of big obstacles indoors, most of the time the WAPs signals cannot penetrate the obstacles, i.e. multipath issue [83]. Thus, such signals may reach SPs by bypass deviation, i.e. NLOS, and then introduce large error on estimating pseudoranges and on location estimation.

Several researchers have been involved to mitigate this inaccuracy in measurements by using different probabilities/statistical and/or mathematical models. For example, due to the fluctuation of WAPs signals, calibrating some parameters for these signals are examined in [33] including attenuation factor of the WAP signal and offset parameter of the RSS. The calibration algorithm has been proposed to improve the accuracy of pseudoranges measuring. However, the accuracy of measured pseudoranges is not adequate as well as the algorithm takes huge processing and then consequently more power consumption. In other study, RSS-fingerprinting technique has achieved better accuracy in [84], but the database generation and maintenance requirements are the main disadvantages of the method.

The other major approach in WiFi positioning solutions is to use time-based approach such as TOA, TDOA and RTT, which are more accurate than RSS technique, as it has been proved in [85]. However, all WiFi time-based localisation solutions suffer from timestamps generation of the received and transmitted signals by using inaccurate local clocks, instability and limited of WAPs coverage [24]. Mainly, the current solutions based on time measuring ignore the use of accurate reference time for clock synchronisation.

In order to improve the localisation accuracy, a combination of RSS and TOA localisation techniques based on WiFi technology is proposed in [31]. The combined approach has achieved higher location accuracy than the RSS technique and TOA technique. However, in most cases, statistical processing or calibration algorithms, again, are needed as a pre-processing step.

Another indoors localisation solution such as iBeaconing based on BT technology is released on Apple-iPhones and Android-based SPs. This solution offers good SP-position accuracy based on the combined version of proximity & RSS techniques [37]. The position accuracy will be varied (up to 2 meters) based on the number of deployed BT-anchors in the vicinity. The main LBS-application based on this solution is starting from shopping to patient monitoring in hospitals. However, the incurring cost to install this solution on SPs and deploying large number of the BT-sensors are the main limitations for indoors-SPs solutions.

SLAM solutions using WiFi, inertial sensors and Map building information based on various localisation techniques, such as TOA, RSS, and DR, are reliable indoors localisation solutions, when Internet connection with SPs is available to connect with the pre-defined radio-map/database of reference locations. GraphSLAM and WiFi-SLAM software are examples of such indoors-SP positioning solutions. Taking GraphSLAM as an example [81], it fuses map buildings-information and inertial sensors readings to define indoors-SPs position by performing statistical/mathematical filtering. The well-known example these filters are particle filter and Kalman filter. However, the achieved SPs position accuracy within 4m – 7m is not dependable for most indoors LBSs.

On the whole, because the indoors environment are complex areas and the need of high location accuracy in SP LBS applications, current indoors localisation solutions based on Cellular, WiFi, BT and inertial sensors technologies do not satisfy LBS users' requirements. Therefore, more researches and further work are needed to mitigate and to overcome these limitations.

2.4.3 Seamless Indoors-Outdoors Localisation Solutions

Outdoors to indoors seamless localisation is one of the main requirements for current SPs LBS applications. However, wireless technologies available on SPs do not provide continuous positioning due to their environmental limitations and their own low performance. The performance of current localisation implementations and their constraints on SPs are shown in Table 2-5. To avoid technologies' environmental limitations and/or to provide outdoors-indoors seamless positioning, combining these technologies should be utilised into a single positioning solution [31].

Table 2-5: Performance of current localisation implementations on SPs.

Technology	Time to fix	Accuracy	Coverage	Environments
GPS	Quick fix, when outdoors	Up to 3m in clear sky.	World wide	Outdoors
WiFi	Quick fix, when Internet and location database are available.	Better accuracy when GNSS is worst, between 5m-15m.	Build up area	Urban/Indoors
BT	Quick fix, when large number of BT-anchors are available	Between 2m-3m.	Build up area (smaller coverage than WiFi)	Indoors
Cellular	Quick fix, when communication with basestations is available.	25m-100m, wherever that there is cellular coverage.	Build up area	Urban/Indoors
Embedded inertial sensors	Possible fixing, when the other technologies are not available.	Up to 10m, for short time since the last calibration (drift issue).	Build up area	Indoors

The combination usually could be based on taking the advantages/capabilities of the technologies and mitigating their limitations. Actually, such combination is not only to offer seamless positioning, but it can also provide other performance improvements including reduce SPs' battery power consumption, reduce time to fix, reducing cost, provide on-the-go anywhere anytime, maximise localisation coverage, and improve location accuracy [86]. However, all these performance improvements are not supplied in a single localisation solution so far, as well as current localisation solutions are normally customised with extra hardware and they incur large cost. Note: to assure the cost level for each localisation solutions, this research study produces Table 2-6. As shown in the table, the cost level is presented in different type of costs including: SW, HW, HR and/or DB/server.

Table 2-6: Cost level for current localisation solution.

Level	**SW**	**HW**		**HR**	**DB/Server**
		Cheap	**Expensive**		
Very-Low (VL)	√	X	X	X	X
Low	√	√	X	X	X
Medium	√	√	√	X	X
High	√	√	√	√	X
Very-High (VH)	√	√	√	√	√

Furthermore, for HW cost either installing expensive basestations such as in SUPL and using vehicles in Skyhook solution or deploying cheap sensors such for iBeaconing solution. In addition, current localisation solutions might use a dedicated DB/server and Internet connection to report the SP location information or sometime the solutions need HR to do the survey or calibrating the installed localisation-infrastructures such in Ekahau. In research community, many trials to provide seamless positioning have been conducted over few years ago. Table 2-7 shows capability of recent SP localisation solutions in comparison with our proposed solution. For example, specifically to hybrid GNSS with WiFi technology: combining GPS technology with WiFi technology based on directional approach of WiFi RSS-Fingerprinting with GPS parameters has been proposed in [87]. GPS parameters include HDOP, Code to Noise Ratio, and the number of satellite signals acquired. The combination scheme provides large reduction in computational burden, different coordinate systems to be used in different situations (LLA for outdoors and XYZ for indoors), also provides intended blocks of RSS-location information to be selected from the database when necessary.

Table 2-7 Seamless outdoors-indoors positioning solutions

Solution	Accuracy	Hybridising onboard technologies	Combined/standalone Localisation Technique	Cooperative	Cost (according to Table 2-6)	Pre-knowledge/ Pre-calibration
Proposed Solution	2 – 3 m	GNSS with WiFi, BT and inertial-sensors	TOA & DR	Yes	V-Low	No
WGCP [88]	Up to 19 m	GNSS with WiFi	Standalone TOA	None	V-Low	No
WGIM [35]	Cell-ID size (e.g. 20 m)	GNSS with WiFi	Cell-ID with RSSI	None	V-Low	No
DREAR [89]	5 – 10 m	Inertial-sensors with Internet	DR & Activity-style	Yes	Low	Yes
Infra-free [90]	Up to 5 meter	GNSS with WiFi and INS	TOA & DR	Yes	Low	No
ADPS [91]	Up to 7 meters	GNSS and INS	TOA & DR	None	Medium	Yes
CPSM [92]	Up to 4 m	WiFi with BT	Distance-based & RSSI-Fingerprinting	Yes	V-High	Yes
HCLSN [93]	Up to 5 meter	GNSS with WiFi	TOA & RSSI-Fingerprinting	Yes	V-High	Yes
IGSC [94]	Up to 4 meters	GNSS and acoustic	Standalone TOA	Yes	V-High	Yes
SUPL [63]	Up to 3 meter	GNSS with Cellular	Cell-ID, TDOA and TOA	None	V-High	No
WiFi-GPS [87]	2-10 m	GNSS with WiFi	TOA & RSSI-Fingerprinting	None	V-High	Yes

Simulation experiments of GPS-WiFi combined positioning algorithm based on Trilateration technique is conducted in [88]. The simulated scheme constitutes a further step to better position anywhere and to ensure continuity of a positioning service. The aim of this scheme is to locate of GPS-enabled device if the number of visible satellites is less than four satellites. Mathematically, the scheme is to complete the set of the equations (which is less than four equations) with equations obtained from the Wi-Fi positioning system. Also, a hybrid urban public WiFi with GPS positioning method to improve the availability and accuracy of positional information in a KBLS has been proposed in [35]. The integrated solution provides full use of the already available public WiFi signals to support correct position of the mobile devices in real time when insufficient GPS data are available for correct and reliable position fixing.

The aforementioned Wi-Fi-aided-based solutions are to improve the location accuracy and to offer seamless positioning when GPS signals are weak or numbers of visible GPS signals are not enough for localisation. However, these solutions are tailored or customised solutions for some specialised scenarios. As well as accurate GPS parameters (e.g. time) for WiFi transceiver clock synchronisation, traffic burden on WiFi networks, establishing special hardware to survey the localisation area, and cost for deploying the localisation solutions have not been considered.

Furthermore, integrating cellular networks with GPS technology could be applied to offer seamless positioning service. Using cellular signals, aiding information (like position information, time and frequency information) could be received from a server in the cellular network to enhance SPs' GPS receivers, when in harsh areas. For example, an enhanced method by hybrid of Cell-ID, OTDOA and AGPS using LTE is analysed in [63] to improve position accuracy up to 3 meters seamlessly from outdoors into indoors by using cellular signal timing and consequently to reduce TTFF and battery power consumption.

The other possible way to achieve seamless SP localisation service is to use inertial sensors to aid GNSS technologies. These sensors are available on the SPs, and using these sensors with GNSS technology for localisation can offer seamless localisation [91] and SPs' battery power saving [95]. However, using such sensors need calibration algorithms that are accurate for up to only few seconds.

Cooperation between SPs, currently, is a new solution to improve location accuracy as well as to offer continuous & reliable localisation solutions. A GNSS based cooperative location optimisation scheme has been developed using a host server to fuse location coordinates supplied from onboard GNSS of any group of cooperative-SPs to improve location accuracy [94]. Then, pseudorange estimation between the group SPs is calculated based on TOA technique using acoustic signal. The server then, as a final stage, receives these pseudoranges and uses a complex optimisation model to obtain further location accuracy improvement, within 1.2 – 4 meters. Obviously this scheme has two main drawbacks, such as:

1. It needs to access a dedicated database/server to improve and share the location information among all these SPs which acceptable as a small overhead.

2. Porting the task of the server into the SPs will eliminate the overhead of this server and its associated wireless connectivity, but the optimisation algorithm will take considerable resource and time that will drain the SPs batteries.

WPS-Skyhook enabled SPs can obtain WAPs location in any vicinity, as mentioned before. A group of such SPs can then use these WAPs as reference point to locate themselves within a claimed 10-20 meters when indoors. An improved location can be achieved if a GNSS position from an outdoors SP is shared with this group of SPs via WiFi connectivity. This can be achieved by applying "conditional prior probability" to improve the indoors-SP location via probability distribution of the set of shared information (WAPs pseudorange, GNSS location of the reference SPs outdoors). For example, the "cooperative SPs localisation" algorithm in [93] is based on four probabilistic methods namely: 1) centroid method, 2) nearest neighbour method, 3) Kernel method and 4) WAPs density method. Both empirical and simulation results claims that the WAPs density method provided more accurate results than the others, since WAPs density provides a function to distinguish the overlapped or the common shared WAPs information between the outdoors SPs and the indoors-SPs. However, this location enhancement has resulted in 5-meter accuracy.

Also, an infrastructure independent cooperative indoor localisation (i.e. on-the-go) using sensors onboard SPs GNSS, inertial sensors such as accelerometer & magnetometer, and WiFi has been implemented to locate indoors-SPs to within 5 meters [90]. In this solution, a group of WiFi networked SPs, when outdoors, start a calibration process where estimated heading error is calibrated by GNSS heading estimation, and where pseudoranges error between these SPs is mitigated by detecting pedestrian-step trajectory using the onboard accelerometer. When indoor SPs join this network, shared location information will help establish initial position and the heading calibration process of these indoor SPs. Experimental results show that this cooperative solution can achieve location accuracy up to 5 meter, if number of SPs is exceeds 40.

In another vain, to avoid the use of aided reference-positions and/or fixed devices such as WAPs and beacons, when indoors, DREAR [89] proposes a new solution for indoors-SPs localisation using onboard sensors based on user-activities recognition. I.e. the solution is completely independent of using any infrastructures and offers low cost solution. DREAR uses DR techniques to locate any indoors-SPs based on some pre-defined constraints such as user's motion-style, taking escalators and climbing stairs. This is important to mitigate the accumulated positioning error that caused by inertial sensors such as gyroscope. The solution is also follows to a client-server concept in which the coarse position based on DR is processed on client-side, while the refinement of the obtained position is performed on server-side using the defined constraints. The obtained results from a set of trials show that the achieved SP-position accuracy is within 5-10 meters.

Another collaborative indoors-SP-based solution using BT-RSS measurements between SPs has been proposed in [92] to improve indoors-SPs location. In this solution, the indoors-SPs, first, use the measured WAPs-RSS values to define their location via existing WiFi-Fingerprinting technique. Then in next step, the solution estimates pseudorange measurements between SPs by using BT-RSS measurements values to narrow the accuracy of the achieved SPs location. The process of location improvement is based on using force-directed-graph concept such as spring model. Different experiments in various indoors situations have been conducted to validate this solution. The high position accuracy that has been achieved is near to 4 meter.

These seamless localisation solutions need further investigations to offer a robust, applicable and reliable solution. Furthermore, locating SPs via these solutions are based on the estimation process, i.e. real complexity of indoors; obtaining high location accuracy, cost and traffic of the wireless networks are not considered.

2.5 Summary

Outdoor localisation solutions have become a huge success mainly because it provides an ideal environment for innovation. Currently it has the advantage of reliable positioning via GNSS and a defined business model for the delivery of content to the LBS users. However, the same cannot be said for indoor LBS. Probably because both pre-knowledge of location-reference database and infrastructure have to be provided by the owner of the building, indoor positioning and context-sensitive services have mostly stayed away from each other. Thus, this means that current solutions still suffer from the lack continuity due to loss of GNSS signals when SPs go indoors, unless pre-installed localisation sensor networks, such as those offered by BLE-iBeaconing and WiFi fingerprinting, are specifically implemented to provide the SPs location while indoors in that particular vicinity. GNSS signals suffer from strong attenuation caused by building materials when the SPs enter indoors. I.e. the provision of semi-accurate indoors-SPs location, on-the-go anywhere anytime, has proven somewhat problematic to deliver thus far.

After having reviewed the available localisation techniques, technologies and solutions on SPs, we have established the following investigation:

1. Achieving accuracy of SPs location in localisation solutions is varying according to: environmental complexity, using localisation techniques as a standalone or as a combined approach, limitations/capabilities onboard technologies, hardware or

software designed of the solutions, and mathematical/statistical calculation 'SPs position method.

2. Existing localisation techniques, such as RSS-based techniques, do provide good performance (despite signal propagation problem) but at the expense of pre-installing dedicated infrastructure and therefore limited in LBS applications. Other distance-based techniques are sufficient on its own, but it suffers from jitters, instability, coverage and dilution of precision issues. Finally, DR technique, especially when using low-cost inertial sensors such as accelerometer and gyroscope onboard SPs, are highly smooth and stable, but their performance degrades quickly over time due to the accumulated measurement noise of sensors causing cumulative positioning error. Therefore, a combination of these techniques is necessary to exploit the advantages of each of these techniques while compensating for their limitations.

3. Cellular, WiFi, BT or inertial-sensors based positioning solutions on SPs can be used to define SPs location as an alternative solution in environments where GNSS-signals are not available. However, limited coverage of WAPs/BT-anchors, no information of BSs/WAPs/BT-anchors physical positions within a building, no access to API functions of important device data onboard SPs, no localisation protocol extensions, no synchronisation between SPs and reference stations are some of the main challenges to design a spontaneous autonomous positioning solution with reliable accuracy at reasonable cost.

Due to these limitations, currently technologies and/or localisation solutions on SPs do not provide seamless positioning solution from outdoors to indoors, accurately, on-the-go, anywhere, anytime. To achieve this, this research study developed a hybridisation solution of GNSS with other onboard wireless and sensors technologies. The developed solution is a good candidate for seamless LBS localisation applications on SPs. This is because:

1. The developed solution, to calculate accurate indoors-SPs position (e.g. within 2 or 3 meters accuracy), uses BT signals between SPs to construct an aiding network and then to: a) synchronise all reachable WAPs with GNSS time from outdoors SPs (**see section 3.4**), b) define synched WAPs position (**see section 4.4**), c) exchange and establish SP location and time-offsets based on available/reliable GNSS location from outdoors SPs (**see section 5.4**), d) Hop-counting synchronisation to estimate pseudoranges between connected SPs (see **section 5.4.1.1**), e) SMSR to enhance the estimated pseudoranges (see **section 5.4.1.2**), f) PRP to mitigate DOP issue (see section **5.4.1.3**), g) and to then calculate location of indoors-SPs based on the proposed SILS (see **section 5.4.1**).

2. The developed solution is also fusing inertial sensors measurements with the measurements from SILS so to improve localisation accuracy when deep indoors, as we called UNILS, (see **section 5.5.1**). This means that, in deep indoors, UNILS utilises only available devices/sensors on SPs, when communication with WAPs or BT-anchors is deemed unreliable or unavailable. In another vain, this fusion of measurements provides a combination of localisation techniques, (see **section 5.5.1.3**). For example, when the UNILS introduces pseudorange-measurements (via TOA technique) into the DR technique, it leads to mitigate or even to avoid the accumulated drift issue. Also, using enhanced DR measurements at the same time is to keep stability of pseudorange estimation and reducing the number of reference positions which are constrained in TOA technique.

3. The developed solution doesn't need any dedicated hardware as all the SPs are equipped with a Wi-Fi transceiver as well as there is no need to deploy an extra dedicated network as radio signals from at least a few WAPs can be detected in the majority of areas of interest, due to the proliferation of WLAN. Furthermore, the developed solution doesn't require radio mapping and calibration which are typically associated with current indoor solutions including Skyhook and Ekahau. Also, the solution doesn't need to use Internet service to connect with host servers or location-databases which is associated with the current solutions (WiFi-SLAM

and Sensewhere). Therefore, the solution shall reduce the required memory and traffic on SPs thus saving battery consumption, connection/interaction traffic and reduce processing time.

4. The developed hybridisation solution is compatible with all SPs interfaces. Because the solution will work regardless of how the user holds the SP, it is not restricted to a direction to the WAPs or other SPs in the vicinity such as in the case of RSS-Fingerprinting technique.

To the best of my knowledge, my proposed hybridisation technologies and combined techniques solution to offer seamless, accurate, low-cost and on-the-go localisation solution for LBS application on SPs has not been attempted in other work, and thus represents a novel direction of research.

CHAPTER3. UTILISING OBTAINED GNSS TIME ONBOARD SMARTPHONES TO SYNCHRONISE WAPS CLOCK

3.1 Introduction

The literature review on existing localisation techniques concluded that WiFi time-based techniques suffer from (as it is included in **section 2.2.3**): inaccuracy of available local clocks, low quality of clock synchronisation between WAPs and/or mobile-devices (e.g. SPs), and no WAPs localisation protocol extensions. These limitations produce huge SP positioning errors and they create new challenges to define SPs location.

In this research study, a WAPs-SPs clock synchronisation based on a developed DWC model is implemented for localisation schemes that accurately utilises GNSS time (obtained from SPs) to synchronise WAP-clocks and then to aid SPs localisation. By using GNSS time, obtained from when the SP(s) was/were insight of GNSS signals, the SPs can measure clock time delays/offset and virtually synchronise in time with any available WAPs. This time delay/offset knowledge will then be used to determine the SP's position accurately when indoors.

This chapter is organised as the following; **section 3.2** introduces the main problem of SP location estimation via WiFi technology using both RSS-based and time-based techniques as well as it shows the accuracy of the available time-source functions on SPs to generate timestamp measurements. **Section 3.3** gives a background on WAPs clock accuracy and architecture as well as explains the limitation and challenges for synchronising WAPs clock with SPs clocks. In **sections 3.4**, possible WAP-SPs clock synchronisation protocols/algorithms which can be implements on SPs and the proposed WAPs-SPs clock synchronisation algorithm for localisation purposes are examined in detail. Then, **section 3.5** shows the test, simulations, results and evaluation of the proposed algorithm in comparison to the reference/existing algorithm. Finally, **section 3.6** summarise the proposed novel algorithm in terms of accuracy as well as lists the achievements of the proposed algorithm for SP-localisation purposes.

3.2 Problem Statement

Wireless communication protocols and pseudoranges-based localisation technique (using time measurements) depend on the time synchronisation of all involved devices in such network/solution. However, these devices include inaccurate clocks and low-quality of clock synchronisation, which are considered as the main inaccuracy factors in time-based localisation measurements, since a one microsecond clock time error shall cause a 300 meters in position error (based on the speed of light).

Therefore, to ensure the accuracy of pseudoranges between SPs and WAPs and then the SP location estimation via WAPs signals using time-based localisation technique, the development of an efficient/accurate clock synchronisation algorithm between WAPs and SPs is required. This section, first, overviews the main issues of SP location estimates via WAPs signals, then it demonstrates the timestamp accuracy of time-source functions on SPs.

3.2.1 Overview

SPs contain both Wi-Fi and GNSS technologies. When outdoors, the GNSS receiver onboard any SP can define its location and provides accurate time. This capability is lost when the SP is indoors [96]. In another vain, WAPs can be used to locate indoors-SPs, using Trilateration. However, WAPs do not have accurate, GPS like, timing and are not synchronised in time together [97]. Putting aside indoors multipath and obstacles issues, the following are example of two localisation techniques that can locate SPs based on WAPs signals in the vicinity:

1. The first technique is RSS-fingerprinting; the detailed explanation of this technique is described in **section 2.2.4**. Several implementations of this technique have been concluded, however, extra/special hardware (like host/server) and radio-mapping are the main challenges to build and to maintain the database of the WAPs and areas within their coverage. Obviously this has big overhead in infrastructure as well as maintenance processing.

2. The second technique is pseudorange-based or distance-based where an SP can measure pseudoranges from multiple WAPs, if WAPs are synched in time and if their location is known, based on RSS or TOF observation. Then by applying Trilateration, the position of the SP can be defined. The literature survey has concluded that (see **section 2.2.4**) WAPs-RSS measurements have signal propagation issues and so any pseudorange estimation based on RSS would not be accurate. Also, onboard-SPs WiFi-transceivers and WAPs include a low grade clock to measure TOF that would produce large errors (see **section 2.2.3)**.

Pseudorange-based technique using TOF measurements is proven to offer accurate estimation than when it is based on RSS measurements, when a calibration or conditioned algorithm such as calibrated RTT measurements is used [98]. Although, RTT-based solution avoids the issue of SPs-WAPs clock synchronisation, however, it will affect on the network traffic load as well as drain the battery-power on SPs hugely due to exchange large number of frames between SPs and WAPs such as: RTS)-CTS and ACK–DATA [99]. Hence, any localisation solution based on this technique will not be applicable on SPs in real time and cannot be adopted with on-the-go solutions.

Therefore, to define an SP location, specifically for WiFi time-based localisation solutions, in real time to within few meters, the SP needs accurate timestamps at nanosecond resolution as well as synchronisation algorithm to measure its pseudoranges from multiple WAPs. Note, WAPs and WiFi-transceivers clocks on SPs are already synchronised via broadcasted WAPs-beacon signals during the network processing, but it is not sufficient for localisation purposes, since the quality of the clock-synchronisation is within few microseconds.

To achieve these timestamp-resolution and accurate clock-synchronisation, this research study implements a WAPs-SPs clock synchronisation algorithm by using an accurate reference time (like GNSS time) instead of using any other reference-time or timing-function sources such as: SPs-CPU clock, SPs-Kernel timer and the Network Time. The SPs-WAPs clock synchronisation algorithm includes the following distinct features:

1. Utilising the obtained accurate GNSS time on SPs within few nanoseconds (e.g. ± 6 nanoseconds), when outdoors, as a reference time to synchronise SPs-WAPs clocks,

2. Developing dynamic WAPs clock model which includes clock offset and all noise sources which exists in actual WAPs clocks such as: white noise on frequency, flicker noise on frequency, white noise on phase, flicker noise on phase and random walk on frequency,

3. Utilising broadcasted WAPs' beacon frames/signals to achieve the SPs-WAPs clock synchronisation. This mechanism is preferable, because when these beacons are used via SPs passively, the algorithm will not affect on the network traffic, since WAPs periodically broadcast these beacon signals to advertise their availability.

The SPs-WAPs clock synchronisation is evaluated in various localisation experiments/scenarios in more detail, including the OPNET-simulation implementation and results, in **section 3.5**.

3.2.2 Timing-Function Sources

This section addresses the fourth research question (as it is mentioned in **section 1.4**): what is the accuracy of the available timestamp-function sources on SPs for estimating TOF? Specifically, for SP localisation based on WiFi-time localisation solutions; actual SP's Kernel, for example Android-based SPs, supports several time-source functions to generate timestamps and then to estimate TOF including the Kernel timers, CPU clock-counter and WiFi-MAC clock. However, these timing-source functions on SPs to generate the timestamps, based on trial experiments, are not adequate to meet high quality of pseudorange measurements as well as they are not able to achieve high location accuracy. This is because:

1. The Kernel timers have huge time degradation, i.e. the time drift/delay, since they are interrupted by "software/hardware interrupts". To prove this, a set of trial experiments, during this research study, have been conducted on actual-SPs. Figure 3-1 shows the result of one of the experiments which includes two trials one for HTC phone and the second one is for Samsung-Galaxy II phone.

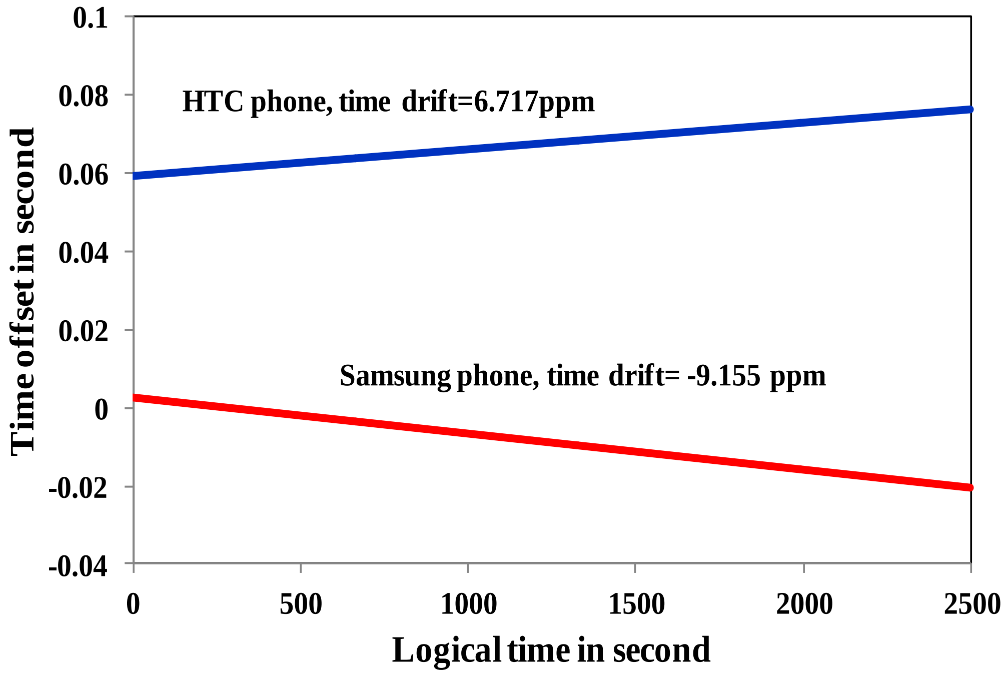


Figure 3-1: Timer-drift estimation example of Android-based SPs.

The experiment starts by reading the timer at kernel level every second, for 2500 seconds. After having a set of record values for each phone, the linear regression (based on linear-least-square) using the equation 3.1 is applied on the set readings to determine the slope (i.e. time drift) of the set timer-readings [100]. We need add the text to explain the estimated time drift

$$TimeOffset_i = a * LogicalTime_i + b \qquad (3.1)$$

Where TimeOffset is the difference time between logical time and timer-reading is, a is the estimated slop (i.e. the estimated time drift) and b is the estimated intercept between TimeOffset and LogicalTime values. The estimated time drift by the regression method is 6.71 microseconds (forward) for HTC phone and -9.155 microseconds (backward) for Samsung phone. As it is noticed, in both case the time error is within ± microseconds, which are inducing huge positioning error.

2. The CPU clock is accurate than the Kernel timer due to being handled by "hardware interrupts" only as well as the resolution is within nanosecond resolution. To show the accuracy of CPU clock on Android-based SPs, a set of trial experiments are conducted using the cycle-counter register that has, until recently, been a low-overhead and an excellent high-resolution way of getting CPU timing information [101] on SPs. Taking the result of a single experiment, a histogram of the difference measured CPU clock cycles between any two readings is illustrated in Figure 3-2.

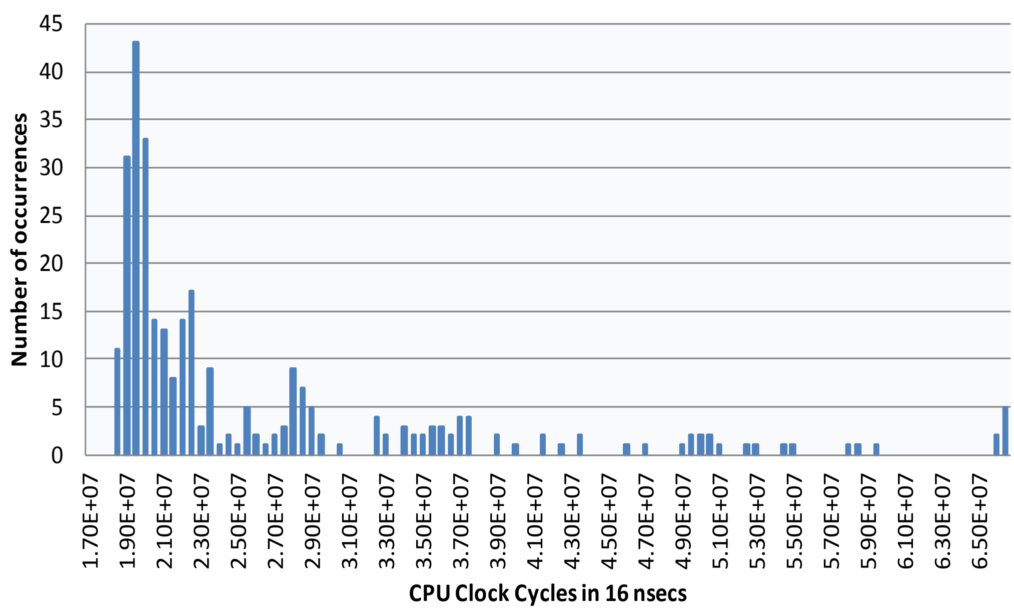


Figure 3-2: Occurrences of CPU clock cycles' histogram, every 0.4 second.

The parameters of the experiment are configured as follows: a) the time interval between any two TSC register readings is 0.4 second, as shown in Figure 3-3, b) the selected length of the CPU clock cycle is 16 nanoseconds, c) and the number of the records, i.e. the total number of the resister readings, is 300 records. However, as the reading of CPU clock cycles, there is unstable behaviour of measuring number of cycles. This is due to interrupt the CPU clock counter by different embedded HW on the SPs and the quality of the CPU clock material.

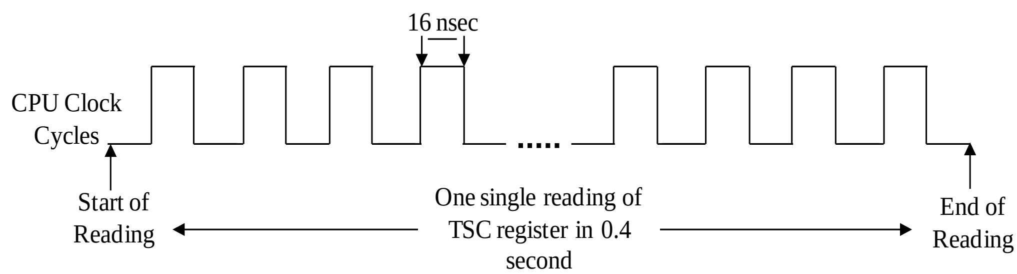


Figure 3-3: The CPU clock timing diagram.

3. When WiFi-MAC clock on SPs is synchronised with the reference clocks in a network, i.e. with WAPs clocks, the accuracy or the clock offset is in the order of ppm. In addition, the resolution available WiFi MAC clock is ~22.5 nanosecond, which is equivalent to approximately 6.75 meter of positioning error, since the maximum rate of the MAC clock could be 40 MHz or 44 MHz [97]. To demonstrate the clock drift and clock offset of a WAP in relative to a WiFi transceiver onboard SP, this research study conducted few trial experiments on actual SPs. Figure 3-4 shows the results of the one trial experiment. During the experiment an HTC Android-based SP is used to collect the clocks measurements. The SP's WiFi transceiver is worked in monitor mode to receive WAPs beacons frames passively. The SP calculates the clock offset by timestamps for the received beacons and retrieves timestamp-function values from the beacon frames. Also, to calculate the relative clock drift, the linear regression method has been applied (based on the equation 3.1). As it is observed, that the clock error (clock offset + clock drift) without any calibration or compensating algorithms is within microsecond's level which is produce huge positioning error (one microsecond error in clock measurements is equivalent to 300 meter in positioning error) .

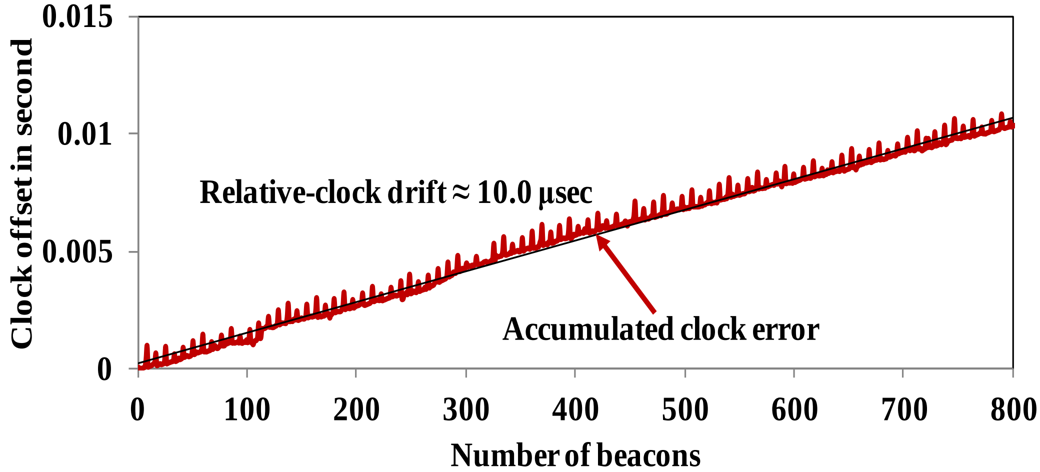


Figure 3-4: Clock time different between an onboard SP WiFi transceiver and a WAP.

Thus, the accuracy of these timing-source functions on actual Android-based SPs are within microsecond level which are not accurate for synchronising SPs-WAPs clocks and they are not reliable for TOF estimation between SPs & WAPs and then to SP location calculation when the pseudorange-based technique is used.

3.2.3 GNSS-Timing Accuracy

GNSS-enabled SP can obtain accurate time at nanosecond resolution from GNSS receiver when the SP is insight of GNSS signals [102]. This accurate time can be used to generate timestamps and it also can be used as a reference time to synchronise WAPs clock delay/offset accurately. This delay/offset clock knowledge can then be used to estimate pseudoranges between SPs and WAPs more accurately when compared to measurements based on non-GNSS time references. However, most commercially available GNSS receivers onboard SPs provide the GNSS time to the phone-application via the receiver's API as NMEA messages every second with a resolution in millisecond. To achieve a time resolution in nanosecond and so enable using it as an accurate reference time for synchronisation which is necessary for SP positioning, it is necessary to modify the SPs GNSS receiver firmware to output nanosecond resolution time via preferable a dedicated time-function.

Anywise, to show the accuracy of GNSS clock receiver, in this research study, a single trial experiment is conducted using Garmin GPS receiver (low-cost GPS receivers like embedded GPS-receivers on SPs) [103]. The experiment is done when the GPS receiver is fixed all the time, i.e. when the GPS receiver acquired enough SV-GPS signals. Figure 3-5 shows the result of the first trial when the time drift error is within few nanoseconds. As it is demonstrated, the GPS clock is re-synchronised at every second. The time error of the GPS all time during the trial experiment is calculated by using equation (3.2):

$$t_e = abs\big((t2_{gps} - t1_{gps}\big) - (T2_{logical} - T1_{logical})) \qquad (3.2)$$

Where t_e is the obtained time error of the GPS receiver clock, $t2_{gps}$ & $t1_{gps}$ are any two consequent GPS time reading from the clock receiver and $T2_{logical}$ & $T1_{logical}$ are any two consequent logical time.

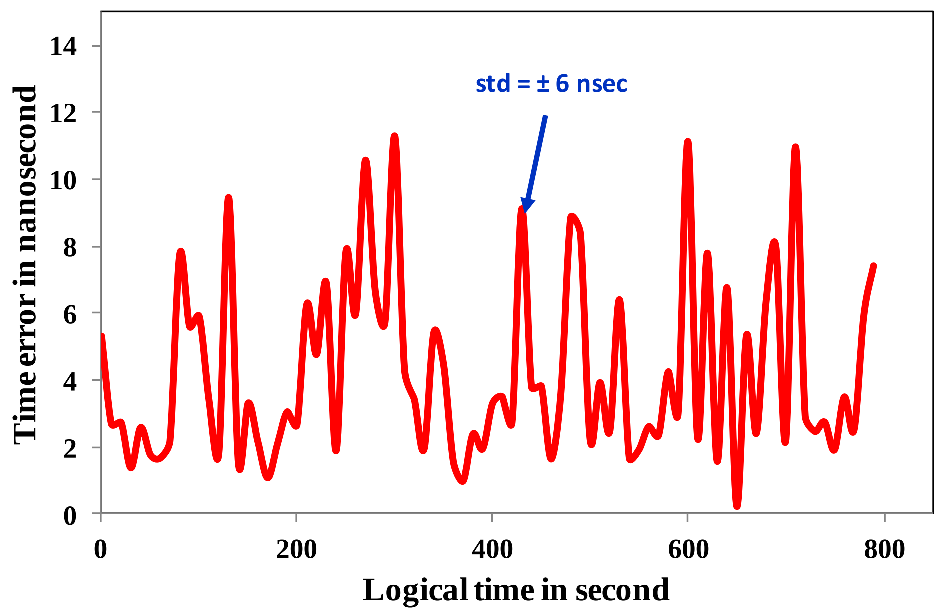


Figure 3-5: GPS-receiver clock error, when the enabled-GPS receiver is fixed.

Thus, it can be concluded that if the SPs are fixed, or when outdoors, they can obtained accurate time within few nanosecond such as ±6 nanoseconds.

3.3 SPs-WAPs Clocks Synchronisation

This section addresses the fifth research question (as it is mentioned in **section 1.4**): how obtained GNSS time on SPs is reliably used to synchronise WAPs in the vicinity? To estimate pseudoranges between SPs and WAPs and then to calculate the target location accurately using time-based technique, an accurate reference time is needed to synchronise the SPs and WAPs clock. In addition, as it is proved practically, in previous section, the only accurate timing-source function available on SPs is the obtained GNSS time, when SPs are outdoors. To do this, technical issues and requirements are demonstrated in the followed subsections.

3.3.1 Limitation and Challenges

This research work found out that the implementation of using the obtained GNSS time to synchronise WAPs on actual Android-based SPs and using onboard SP WiFi transceivers and WAPs has many challenges including:

1. Modify the SP's GNSS-receiver-firmware to output nanosecond resolution time available from the GNSS signal. Since, most embedded GNSS receivers (as low-cost receivers) onboard SPs provide the GNSS time to the phone-application via the receiver's API as NMEA messages every second with a resolution in millisecond [104].

2. Access to the onboard SP WiFi transceiver function API to generate timestamps at low-level layers (MAC layer), this is for avoiding instability delay at higher layers, and needs to add monitor mode (i.e. passive scanning function) for scanning broadcasted WAPs beacon, which is not activated on SPs WiFi transceiver.

Due to these limitations, therefore, this research study has focused on proving the novel idea based on OPNET simulation. Also, to achieve accurate simulation results, a WAP clock model is required. Since, the operation of OPNET clock model for wireless nodes is represented as a timeline sequence of events.

Therefore, a realistic WAP clock oscillator model for such system should include all clock noise sources which are exist on real WAP clocks.

To simulate and implement wireless device clock model in any simulator, we need to: understand what the behaviour of the clock oscillator is? And then what are the main noise sources which are contributed into the real clock-time measurements? The discussion on these questions is shown in the **subsection 3.3.2** and **section 3.3.3**.

3.3.2 WAP Clock Behaviour

Hardware implementation of wireless devices, in general view "clock/time" as consisting of an oscillator that determines the length of the clock time interval and a counter that keeps track of the number of the oscillator ticks. Additionally, the imperfection of the mentioned oscillating operations produces clock inaccuracy. However, as there is no memory in the clocks' implementation, any clock error/inaccuracy is uncorrelated with any previous/next clock cycle errors [105].

Therefore, the clock time errors are accumulated during triggering the clock's counter, which now becomes an implicit storage for the clock's delay. This clock issue is encountered during clock/time measurements experiments and therefore needs to be compensated [105].

The main noise sources of clock oscillator have been identified and classified as: White Noise Family, Flicker Noise Family, and Random Walk Noise [106]. These noise sources of clock oscillator are contributed to introduce the clock time errors, i.e. they are introducing clock offset and clock drift. Also, existing these noise sources is based on material type of clock oscillator, for example clock crystal include all the noise sources while the quartz clock type includes all except white noise on the frequency [107].

For cost-saving, low-cost crystal clock oscillators are used in most 802.11 Wi-Fi transceivers instead of a quartz crystal one. These low-cost clocks are good enough to achieve the strict oscillator performance requirement in WAPs [97]. The irony is, from localisation applications view, clock/time synchronisation is essential to establish a common time reference among the various wireless network nodes and to overcome the inaccuracy of their clocks, since clocks are not stable during the network processing, which results in expensive solutions. Otherwise, the localisation solution needs to implement compensation or synchronisation algorithm to reduce unstable clock behaviour. Such synchronisation is consuming (in terms of cost, time, complexity & power) for most wireless communication applications requiring accurate localisation [108].

3.3.3 WAP Clock Modelling

An accurate clock model should include all clock behaviour issues. These are the clock offset (deterministic time error), as well as precise clock noise/drift (non-deterministic time error) [109]. To do this, this research study developed a DWC model that can be used in localisation applications. This model accurately generating WAP clock delay error dynamically within each clock cycle time. The choice of the dynamic clock delay error was necessary to accurately produce the actual/real-time behaviour of the WAP clock based on the implemented/actual device clock noise.

For implementing of DWC model, the model is designed in a way to ensure that simulations do not run for long time or use large RAM on the host environment. This is important to allow several scenarios run at the same time.

The developed DWC model is shown in equation (3.3). The model consists of three components: 1) time offset according to the reference time, 2) clock offset, and 3) clock drift (the research contribution) based on the real clock noise error sources.

$$C_{time} = T + t * C_{offset} + C_{drift} \qquad (3.3)$$

Where C_{time} represents the local time of the WAP clock, T is the time offset, t is a logical time (counter time increased in 1ms, since the selected beacon frame type is generated in milliseconds-level), C_{offset} is a clock offset in ppm, and C_{drift} represents the clock drift in nanoseconds.

The first component "T" in equation (3.3) is based on how the reference time is accurate (using GNSS time is much more accurate than WLAN time ‘network time’). The second component "t*C_{offset}" contains a specific offset of WAP clock according to the reference clock time in ppm, as a deterministic clock error. This model reference time is continuously being refreshed as and when the reference time is available. Also, this clock model generates a local time within the dynamic time delay error of the clock cycle, based on actual state of the noise delay of the device system; it is not based on static noise delay as can be seen in the comparable model [109]. This is to enhance the localisation accuracy, because every ‘1 nanosecond’ of clock drift “clock error delay” will induce approximately ‘0.3 meter’ in horizontal positioning accuracy (based on the speed of light) [110]. The general structure of the clock model is shown in Figure 3-6.

The component "C_{drift}" (in equation 3.3) represents the dynamic time delay error of the clock, which is represented by the clock drift. The WAPs clock drift consists of five types of clock noise sources (since crystal clock is embedded on WAPs and 802.11 Wi-Fi transceivers):

1. White noise on the frequency (S_WF),

2. Flicker noise on the frequency (S_FF),

3. White noise on the phase (S_WP),

4. Flicker noise on the phase (S_FP),

5. And random walk noise on the frequency (S_RWF).

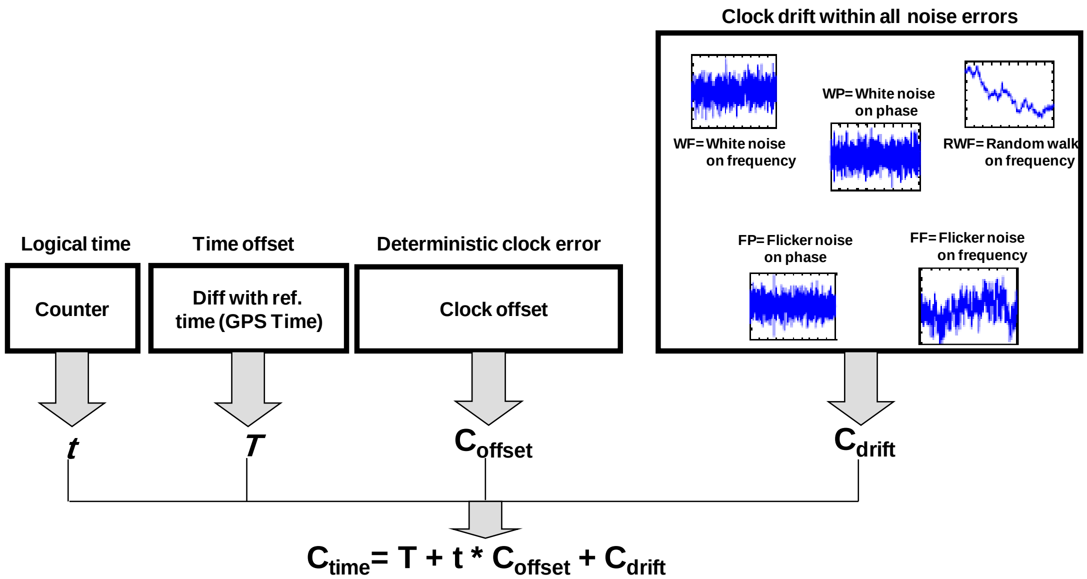


Figure 3-6: General structure of the developed clock model.

Note that these noise sources are changed according to the oscillator type and/or the defining application/environment. Any noise value is based on measuring real clock behaviour and actual parameters. A well-known method for measuring clock behaviour is "Allan deviation" [107]. Table 3-1 shows the Allan variance and scaling variable h-parameter of the typical clock noise error sources, which are used into the model.

Table 3-1: Allan variance and scaling variables.

Clock Noise	Allan Variance	Scaling Variable h-parameter	
S_WP	$(3*h_2*Fh)/(4*pi^2*T_t^2)$	h_2	$3*10^{-14}$
S_FP	$h_1*((1.038+3*log(2*pi*Fh*T_t))/(4*pi^2*T_t^2))$	h_1	$3*10^{-12}$
S_WF	$h_0/(2 * Tt)$	h_0	$1*10^{-12}$
S_FF	$2*h_{-1}*ln(2)$	h_{-1}	$5*10^{-11}$
S_RWF	$h_{-2}*((2*pi^2* T_t)/3)$	h_{-2}	$7*10^{-14}$

Furthermore, the developed applicability range of the WAP clock drift, which is based on each noise sources for localisation application, can be computed using equation (3.4). This clock drift, together with the clock offset, are producing the clock delay error in DWC model.

$$C_{drift} = \begin{bmatrix} S_{WP} * U(0,1)10 * \pi & if\ mod(t, 1ms) \\ -S_{FP} * U(0,1)10 * \pi & if\ mod(t, 10ms) \\ S_{WF} * U(0,1)10 * \pi & if\ mod(t, 50ms) \\ S_{FF} * U(0,1)10 * \pi & if\ mod(t, 100ms) \\ S_{RWF} * U(0,1) * 10 * \pi & if\ mod(t, 1s) \end{bmatrix} \tag{3.4}$$

Where C_{drift} is the generated clock drift, U(0,1) is zero mean and unit variance, and S_{WP}, S_{FP}, S_{WF}, S_{FF}, and S_{RWF} are the scale factors of WP, FP, WF, FF, and RWF clock noise error sources, respectively.

For WiFi time-based localisation purposes and scenarios, this model can be used as a realistic WAP clock behaviour, especially when the WAPs-SPs clock synchronisation and Trilateration (using TOA technique) are implemented (see **section 4.4.2**).

3.4 Clock Synchronisation Algorithms

Achieving high precision clock synchronisation is one of the basic requirements in distributed real-time systems during implementing applications such as localisation applications [111]. Since, practically no two clocks will have exactly the same frequency, and clock frequencies may change depending on oscillator stability and factors such as temperature. Significant frequency changes can occur over time periods as short as few seconds. The local devices' clock drift is inescapable, and so, establishing a reference time common to all devices in the network is needed. To avoid these limitations/complexities of multiple independent clock-sources, a synchronisation mechanism is needed.

3.4.1 Related Work

The existing clock synchronisation approaches including Cristian [112], SNTP [113], IEEE 1588 synchronisation protocol [114], and RBS [115] which are used for mobile-devices (like SPs), do not cover high quality of synchronisation (for localisation purposes) and make huge traffic load on network. This is because:

1. In Cristian's algorithm [112], for example, to synchronise an enabled-WiFi SP transceiver with the clock of a WAP, the SP must send a request to the WAP and records time (t^a). Then, the WAP receives the message and the WAP must immediately send a response with it's ($t^b = t_b^a$) to the SP, as shown in Figure 3-7. Finally, SP receives this at time (t_a^b), and the SP sets its clock to time: $t^b + \frac{1}{2}(t_a^b - t^a)$. However, many samples of time propagation should be measured to do the clock synchronisation as well as when the network traffic is high the response may be delay in an unpredicted manner.

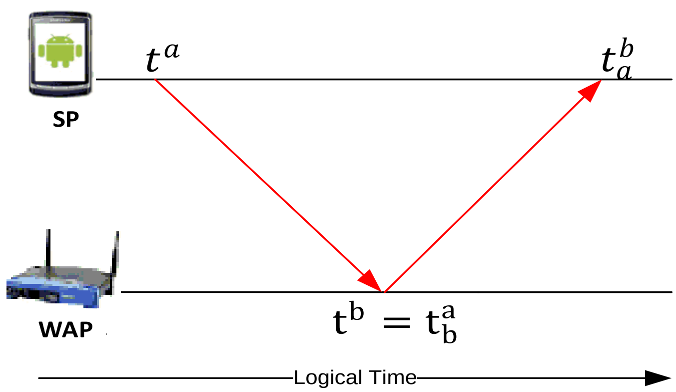


Figure 3-7: Message exchanging in Cristian's Algorithm.

2. SNTP [113] is the same as Cristian's algorithm except that each message has a send and a receive timestamp, thus for total, there are four timestamps. That is, the SP can sets its clock according to clock time of the WAP by using equation 3.5:

$$t^a = t^a + offset_{a \to b} \qquad (3.5)$$

Where $$offset_{a\to b} = \frac{1}{2}(t^b - t_a^b + t_b^a - t^a) \quad (3.6)$$

The main difference between Cristian's algorithm and SNTP is that in Cristian algorithm the WAP doesn't need respond immediately. But, in SNTP, the WAP must have timestamps when the message are incoming and outgoing, as it can be seen in Figure 3-8.

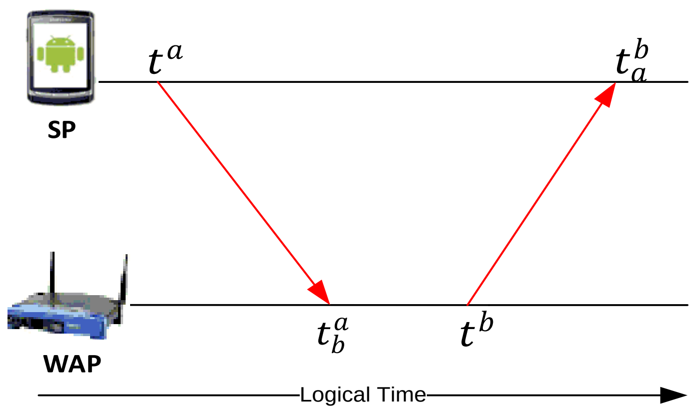


Figure 3-8: Message exchanging in SNTP.

Additionally, the accuracy of SNTP depends on network conditions and the frequency that the messages are sent. However, in all these conditions, SNTP makes huge overhead on the network. Also, the accuracy of clock synchronisation will be in few microseconds, which is introducing big error in localisation solutions.

3. IEEE 1588 synchronisation protocol [114] gives sub-microsecond accuracy at physical layer of the IEEE 802.11b, and few microseconds at MAC layer (software support). This protocol is the same as SNTP, but outgoing message sent times are included in follow-up messages, as it is demonstrated in Figure 3-9. This means, that this protocol makes more burdens on the network traffic than of using SNTP. Furthermore, in comparison with the proposed WAPs-SPs clock synchronisation algorithm, this traffic overhead on WAPs is avoided. This is because the proposed algorithm is based on utilising the WAPs beacon frames

(which are broadcasted periodically, by default) and SPs do not need to connect and exchange frames with the WAPs.

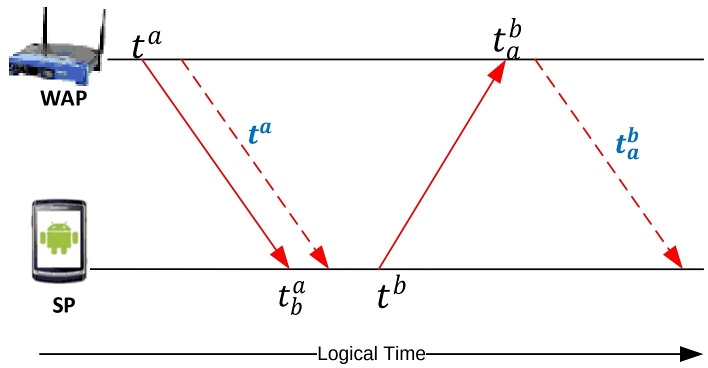


Figure 3-9: Message exchanging in IEEE 1588 synchronisation protocol

4. In RBS algorithm [115], a second SP (as a third-party) is used to synchronise an SP and the WAP. I.e. the WAP works as a broadcast station, while the two SPs are working as receivers, as shown in Figure 3-10. Also, the time transmissions of the messages are not needed. Additionally, a linear regression method should be conducted to calculate the relative clock-drift and clock-offset of the measured timestamps by the two SPs during receiving messages [116]. However, the accuracy or the synchronisation tolerance of this algorithm is in milliseconds.

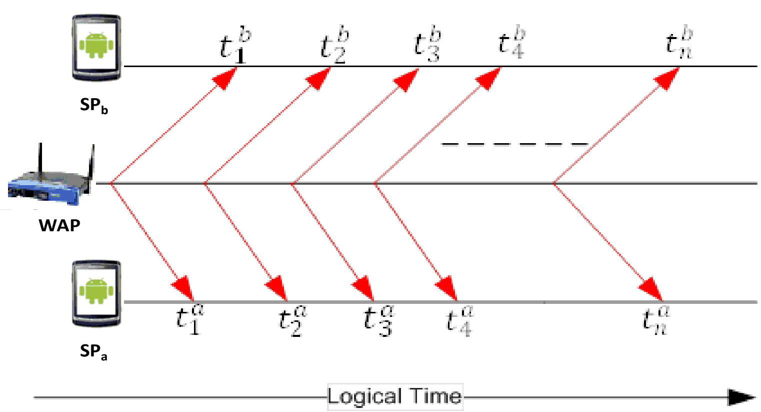


Figure 3-10: Message exchanging in RBS algorithm

All these studied synchronisation algorithms are not adequate for time-based localisation techniques due to their accuracy of clock synchronisation are not on the order of few microseconds as well as they make traffic-overhead on networks .

3.4.2 Proposed Algorithm

A new clock synchronisation strategy/algorithm between SPs and WAPs is needed that will not affects on the network traffic and yet provides high quality of synchronisation. To achieve this, this research study developed a new SPs-WAPs clock synchronisation algorithm that:

1. Uses GNSS-time obtained by nearby SPs when outdoors to synchronise WAPs clock, instead of using any other reference time (like Network Time). The quality of the clock synchronisation algorithm will be in few nanoseconds accuracy (accuracy will be within ± 6-10 nanosecond accuracy when fixing GNSS-enabled SPs within 2-3 meter accuracy).

2. Implements new clock synchronisation mechanism based on exploiting several send and receive timestamps (t_{wm}^{i}, T_{wm}^{i}) from several received WAPs' beacon signals, as shown in Figure 3-11. The algorithm based on this mechanism will not affect on the network traffic, since WAPs periodically broadcast beacon signals to advertise their availability.

The algorithm can be implemented on any SP via a time-based localisation scheme and it starts when an SP (or more) is outdoors, which has already acquired a GNSS fix and has obtained accurate time. As this SP approaches buildings, it shall receive WAPs beacon signals/frames, and will calculate WAP-SP clock offset and drift. These offset & drift clock knowledge will then be used to synchronise the SP-WAPs accurately. Furthermore, direct or indirect SP-WAP clocks synchronisation methods are possible: direct and indirect.

1. Direct clock synchronisation is achieved when any WAP within the vicinity of the SP will sync its clock directly with the SP's GNSS receiver clock. WAPs' beacon signals are used to achieve the synchronisation with SP. Since the clocks of the WAPs are continuous on their skew, see Figure 3-12.a, a virtual WAPs clock time synchronisation can be done by calculating WAPs clocks' skew and storing them

in the SP's memory. This knowledge then could be used to compensate the WAPs clock error during the localisation.

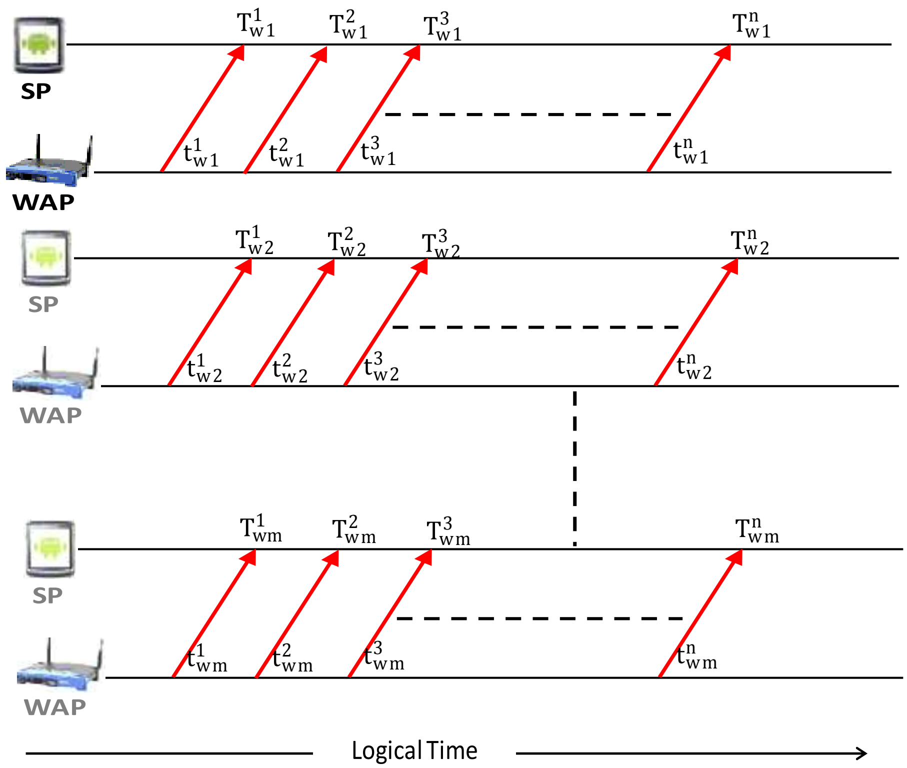


Figure 3-11: Timestamps measurements in WAP synchronisation algorithm.

2. Indirect clock synchronisation is achieved when a nearby WAP sync its clock with the SP's GNSS receiver clock, and then pass-on this time to other WAPs on its wired network, as shown in Figure 3-12.b. This method is useful when the SP can only receive a single WAP beacon signal, at the location of the synchronisation.

For both methods, the WiFi transceiver firmware needs to be modified on the SPs, while the indirect method also necessitates a firmware modification to all WAPs on the network.

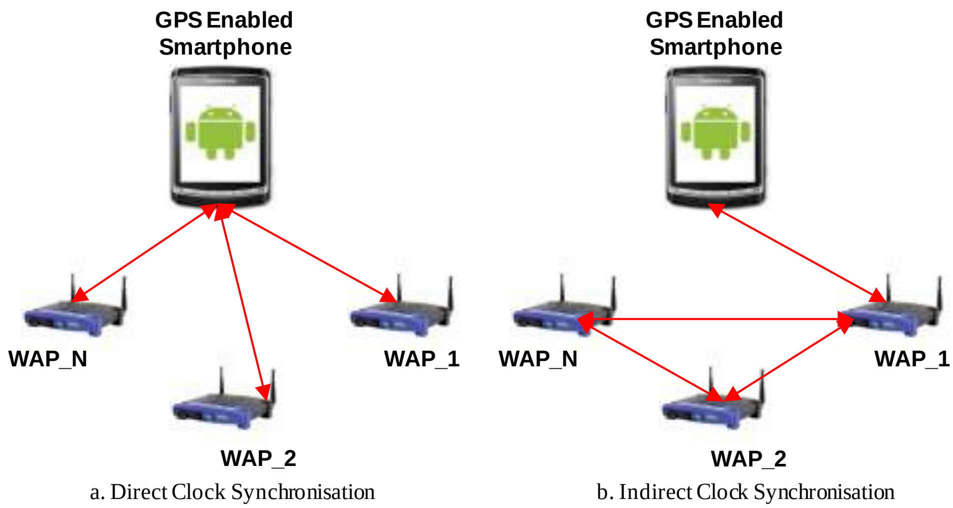


Figure 3-12: Clock synchronisation between SP and WAPs.

To implement this algorithm in OPNET simulation based on the DWC model, the DWC model is customised for generating WAP-local time with the clock delay error at 1milliscond sampling time. This is because the selected beacon signals are transmitted by WAPs at milliseconds rate [100]. The first component "T" in Equation (3.3), in this case, would be based on the obtained GNSS time. Therefore, this model reference time is continuously being refreshed as and when the GNSS time is available. This ensures that the WAPs beacon frame delay is accurate for every transmission when DWC model is used. Also, the effect of temperature on the clock delay in the DWC model has been assumed constant, because its effect is negligible in milliseconds variation.

A flow-chart of the WAP local time generation is illustrated in Figure 3-13. As it can be seen, the model calculates clock drift based on the real clock noise sources, differently, at each cycle. That is, the clock delay error is generated dynamically.

3.5 Tests, Simulation Results, and Performance Evaluation

To prove and evaluate the novelty of using GNSS time to synchronise WAPs based on the DWC model, OPNET network modeller 14.5 has been used to test different scenarios. The implemented DWC model is used for generating timestamps during sends & receives beacon frames as well as for estimating the WAP clock error for clock synchronisation.

Simulation results show that the clock model achieves higher accuracy/quality of synchronisation than the current clock models.

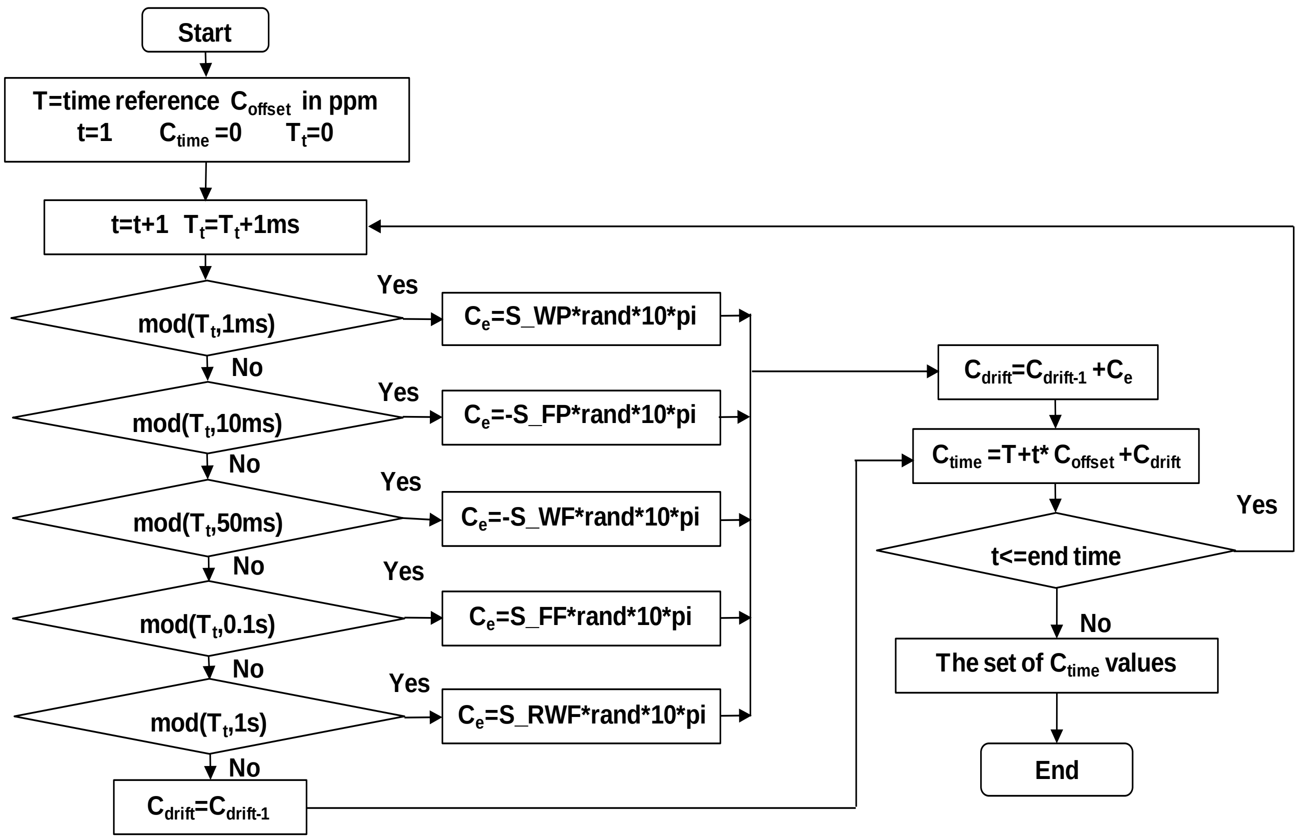


Figure 3-13: Flow Chart of WAP clock time generation.

3.5.1 Test Scenarios and Results

A 1 minute run and a 1 hour run scenarios are tested to prove the model in comparison with the static clock model that has been used as a reference model [109], as shown in 'Figure 3-14 and Figure 3-15' respectively. This simulation assumes that the clock offset in both models has a static value in ppm, and that the reference time for both models is GNSS time. For each of these two scenarios, the developed clock model generates two clock delay errors. The first is when DWC model includes all clock noise sources for a crystal clock (used for WAPs clock). The second is when it includes all clock noise sources, except WF clock noise error because it does not exist in quartz clocks (used for GNSS clock on SPs).

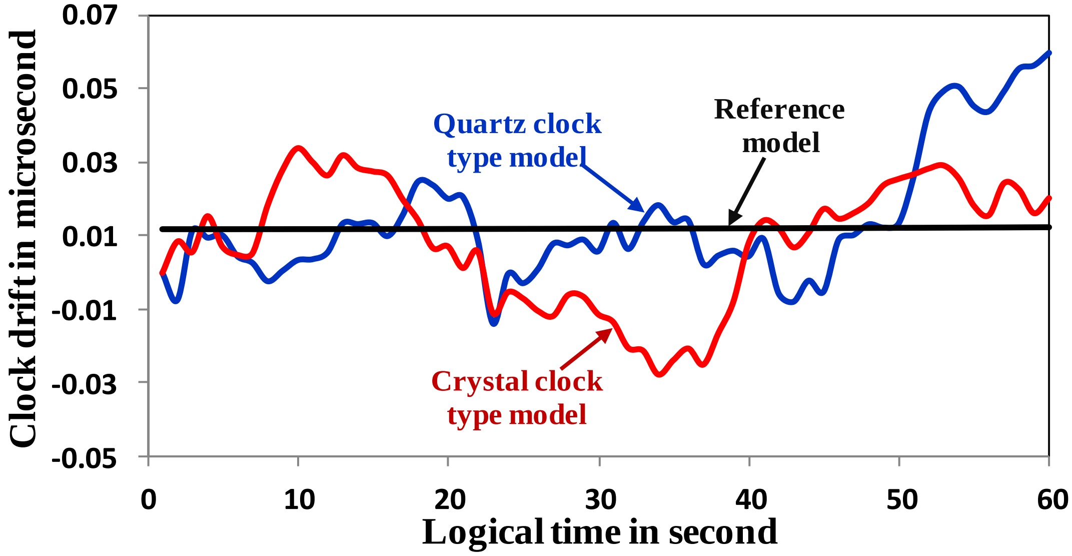


Figure 3-14: Generating clock delay error for developed model and simple model for 1 minute.

In Figure 3-14 and Figure 3-15, the black line represents the clock drift error (in nanoseconds) generated by the reference model, while the red and the blue lines represents the developed crystal clock model and quartz clock model respectively. In both scenarios, DWC model exhibits more realistic behaviour, i.e., the reference model generates static clock drift error while DWC model produces dynamic clock drift error for every clock cycle.

To test the clock synchronisation capability of DWC model, the performances of both models are evaluated in terms of synchronisation quality (time delay error since the start synched time) versus synchronisation interval. This simulation experiment is done using OPNET setup for five different synchronisation intervals (1 ms, 10 ms, 100 ms, 1 sec, and 10 sec), as shown in Figure 3-16. Note: the reason of choosing milliseconds intervals is due to the time-generation of beacon frames broadcasting via WAPs. It shows that DWC model is more practical/dynamic than the reference model.

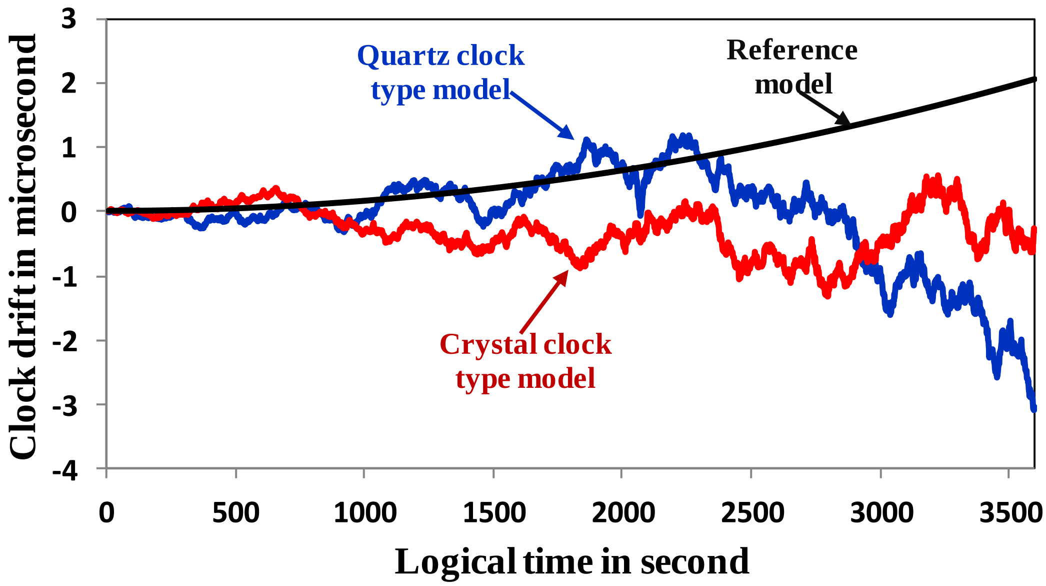


Figure 3-15: Generating clock drift error for developed and simple model for 1 hour.

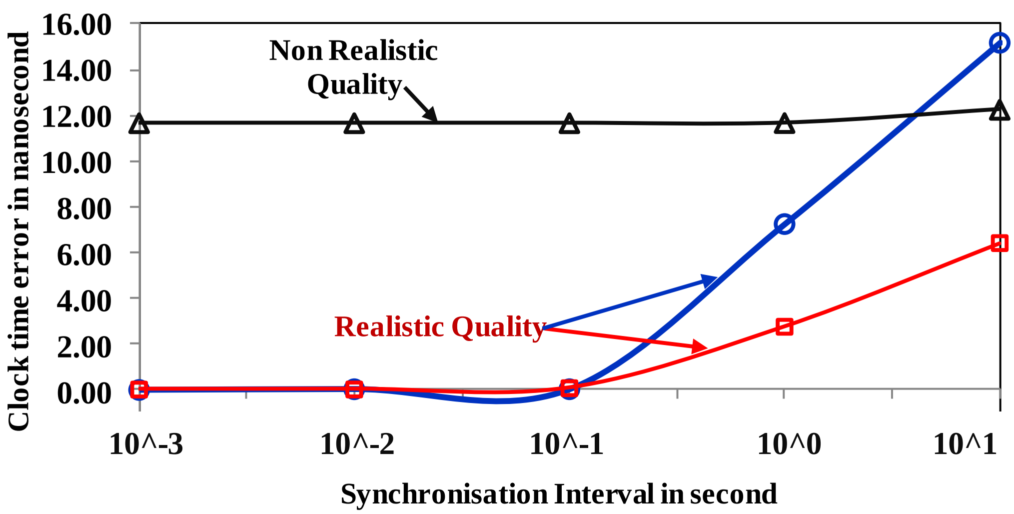


Figure 3-16: Synchronisation qualities vs. Synchronisation interval.

The time errors of the reference model are static as expected [106]. On the other hand, DWC model exhibits time errors in different levels (realistic progression). For example, the time errors at 100 ms, 1 sec, and 10 sec intervals are ~ 12 ns, if the reference model is used for generating the clock time, meaning that, time errors will not change at all (static progression error); while when DWC model is used, including all clock noise types, then the time errors at 100 ms, 1 sec, and 10 sec intervals are 0.2 ns, 2.3 ns, and 6.1 ns, demonstrating different quality (real progression error [106]).

3.5.2 Performance Evaluation

To prove DWC model regarding of timestamp resolution and accuracy for locating an SP based on various WAPs availability, an experiment is simulated in OPNET modeller network simulator as shown in Figure 3-17. In this experiment, DWC model and the reference model are used to generate the "local times" for four WAPs setup for different parameters.

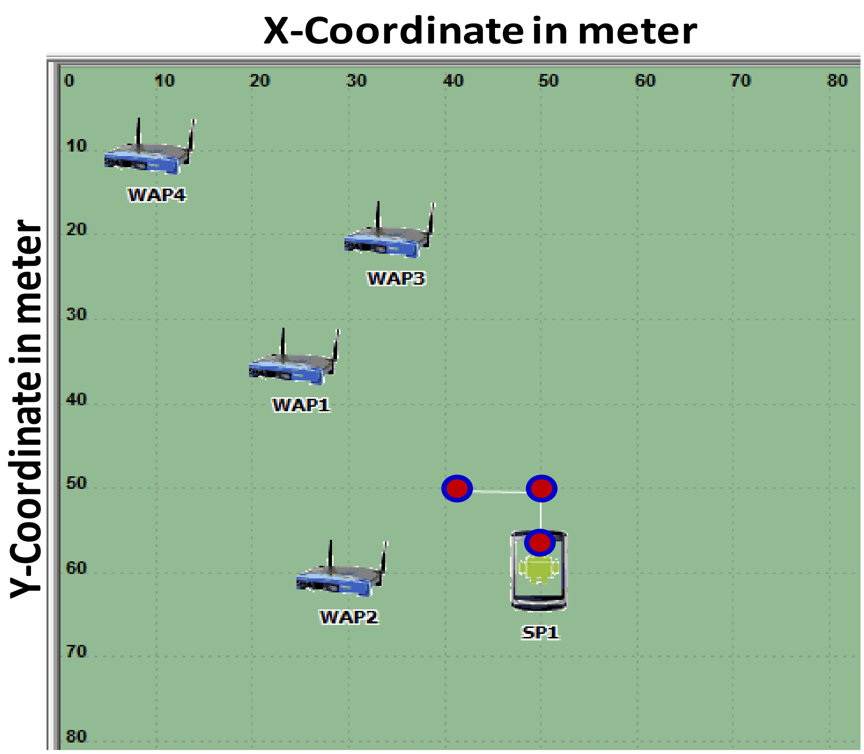
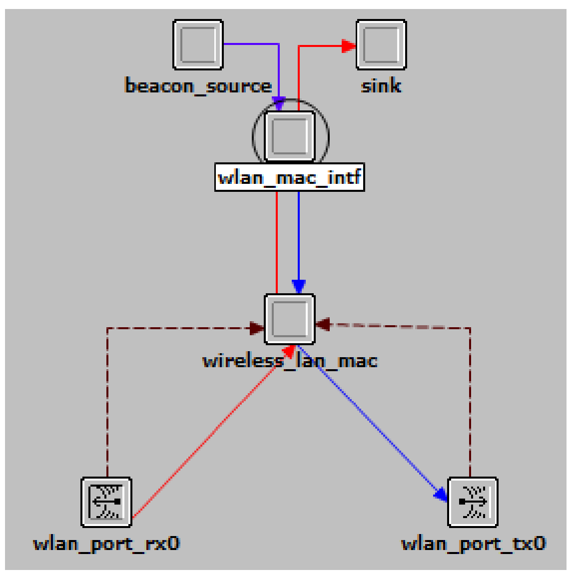


Figure 3-17: Locating an SP (labelled SP1) scenario using 4 deployed WAP signals.

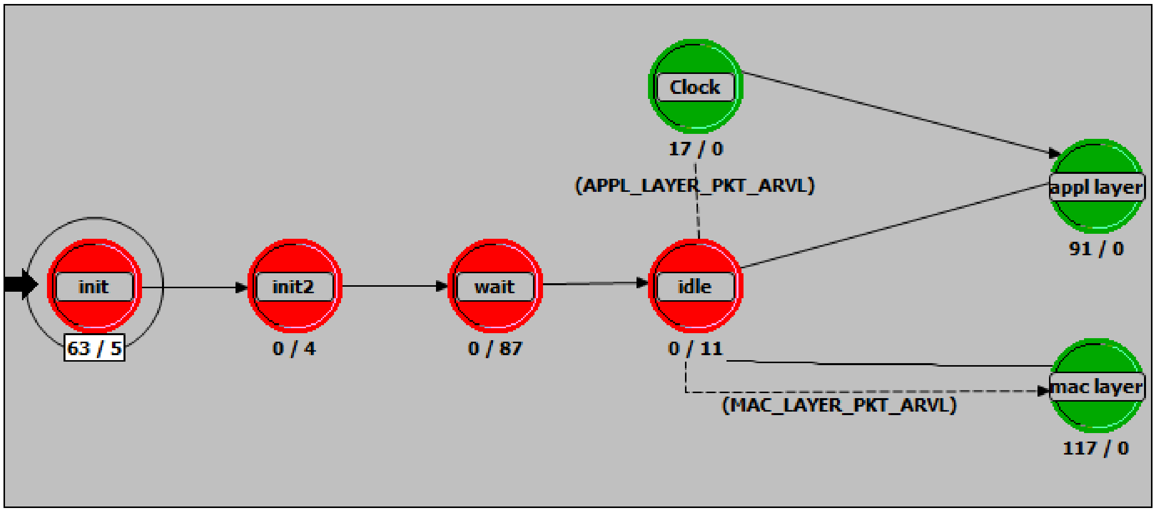
In this simulation experiment different parameters are configured, including:

1. **For WAPs:** The following functions and parameters are added and/or configured:
   a. **MAC Layer:** The MAC layer for the SP localisation scenario was configured based on IEEE Standard 802.11b, where the beacon frame is transmitted at

every 100 millisecond. A single modified WAP-model within implemented DWC model is shown in Figure 3-18. Clock offset and clock drift parameters are initialised at this Layer for all WAP local-times in different values. For example, clock offset is (3, 6, 9 and 12 ppm for WAP1, WAP2, WAP3, and WAP4 respectively) while clock drift for all WAPs is based on the DWC model in ppb. These "local times" are used to timestamp the generated WAPs beacon frames during transmission, labelled “WT”. The timestamps of the received beacon frames at the SP are generated by the logical time, labelled “ST”. The difference between WT and ST are used for measuring the distance between each WAP and the SP. But, to test the single scenario, two different generating local-times are used including WAPs “local time” is synchronised based on Network time (i.e. based reference model) and GNSS time separately.



a- Node-model level



b- Process-model level for WAPs (modified)

Figure 3-18: Modified OPNET WAP process model.

b. Clock Model: This new model represents as local time to make timestamp for each generated beacon frame by WAPs, near to the realistic behaviour. Since OPNET modeller supports time-event based to generate timestamps for beacons or for any events. The new timestamp values are based on the developed WAP clock model. The clock model produces the timestamps at every millisecond, continuously, during the simulated experiment. This assumption is referred to the fact that beacon frame generation interval is within 100 milliseconds. The c-language code of this model in the OPNET modeller is listed in **Appendix A.1**.

2. **For SPs**: The same WiFi-node functionalities have been used for: receiving beacon frame, generating timestamps for the received beacons based on GNSS time. However for localisation purposes SP_Move, GNSS_clock and Locate_WAPs models are added into the SP process model as shown in Figure 3-19, as follows:

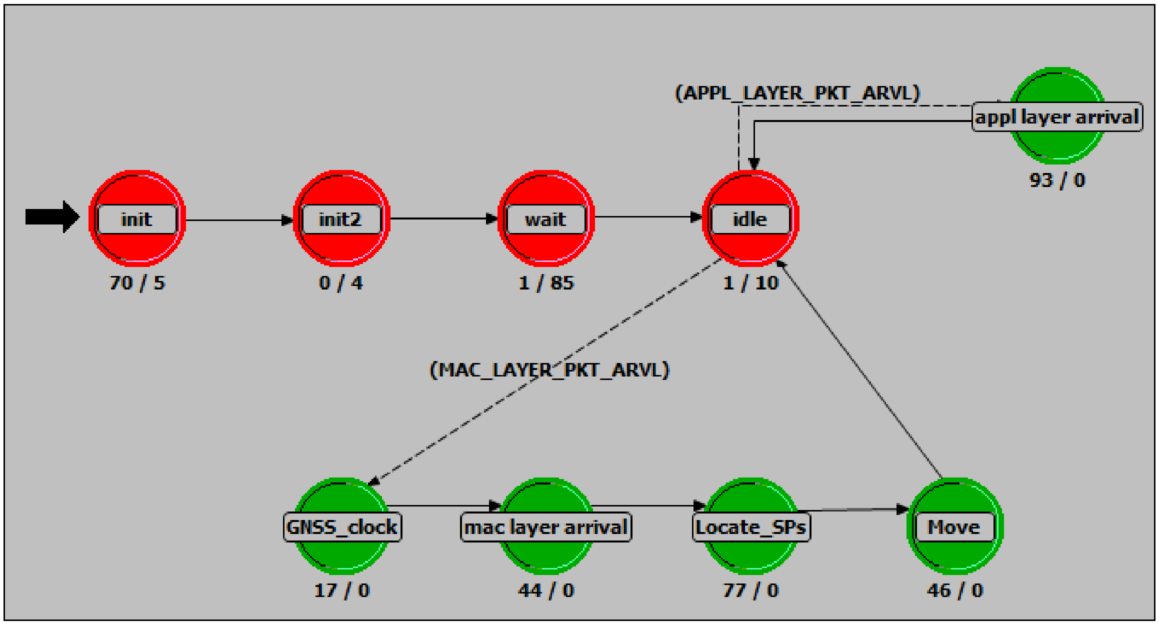


Figure 3-19: Modified OPNET SP process model.

a. **GNSS_Clck:** The SP uses GNSS_Clck to generate timestamp at nanosecond resolution during receiving and transmitting frames. The c-language code of this model in the OPNET modeller is presented in **Appendix A.2**.

b. **Locate SP:** SP uses Locate_SP model to implement the procedure of Trilateration in an iterative LLS and then to calculate SPs location. As well as, this model has already implemented DOP calculation algorithm to indicate the HDOP value during location calculation. The c-language code of this model with the invoked functions for the OPNET modeller is presented in **Appendix A.3**.

c. **Move model:** A new trajectory mobility model was implemented at MAC layer, as shown in Figure 3-19, and it is used to model the SP moving around the WAPs. The speed for the SPs' moving was configured to 1 meter/second.

Since, OPNET trajectory models either based on some pre-defined random waypoints or based on fixed/variable waypoints over the simulation time. Thus, for test localisation algorithms moving node from one position to the next position will affected on location estimation accuracy if these models are being used. The new implemented trajectory mobility model avoids all these issue. Thus, the trajectory mobility is accurate and controllable (in terms of localisation) than the predefined OPNET trajectory models, since at every change of location the new model will precisely gives location information and putted into the localisation algorithms. The c-language code of this model in the OPNET modeller is listed in **Appendix A.4.**

The performance metric considered to evaluate the DWC model and compare it with the reference clock model in the localisation scheme was the location accuracy. In one scenario, the SP is located in three consecutive positions by calculating its distance from the four WAPs with known WAPs location.

The results show that the standard deviation of the accuracy in the SP location is 17.6 meters when using the reference model. When the "local times" generated by DWC model, the standard deviation of the accuracy of this SP location is reduced to 5.2 meters. Therefore, the developed model is more accurate when compared to the reference model. This is because when using the reference model, the accuracy of the measurement is un-stable, which is solved by using DWC model.

3.6 Summary

Available timing-function sources on SP cannot be used accurately to measure TOF measurements and then to estimate pseudorange between SPs and WAPs, since their accuracy and resolutions are within microseconds. This inaccuracy timing will produce large positioning error.

The novel SPs-WAPs clock synchronisation algorithm based on the DWC model, which can be used for accurate study localisation solution, in different simulated scenarios is developed. The accuracy of the clock model which represents the wireless device in the network is important to achieve dependable localisation. This is achieved because the accuracy of the developed DWC model is based on using accurate reference time, and including the clock offset and the clock drift in the model. The DWC uses GNSS time as a reference time and also models the clock offset & clock drift in a precise calculation based on actual clock noise sources. This is necessary to make the simulation model of the clock as accurate as the actual hardware clock inside WAPs.

It also observed that an efficient way to evaluate the performance of the enhanced clock model should be based on the quality of the clock synchronisation. Furthermore, precise clock synchronisation is a requirement for accurate localisation-algorithms, especially for algorithms which are based on the TOA technique.

OPNET simulation experiments prove that the quality/accuracy of DWC model is close to that of actual hardware clock synchronisation. The result of the single scenario to locate SP via WAPs signals using Trilateration, shows that our model achieves higher location accuracy of up to 41.33 nanosecond than existing approaches; resulting in average of 12.4 meter position error improvements. Note: the SP is located using the beacon frames from the four WAPs with known WAPs location to. I.e. through running the experiment, the SP uses pre-defined WAPs position. In order to overcome this limitation, a new scheme to locate WAPs, on-the-go via a single SP or multi-SPs, based on the WAPs-SPs cloak synchronisation algorithm is proposed in as well as the evaluation of the scheme is presented in next chapter.

CHAPTER4. WAPS LOCALISATION SCHEME FOR SEAMLESS SMARTPHONES POSITIONING

4.1 Introduction

Providing the position of WAPs is one of the most viable requirements in WiFi-based localisation solutions, through which indoors-SPs can calculate their location based on this WAPs-positions information. Time based localisation technique could be utilised for locating SPs when the clock of WAPs is synched accurately and WAPs positions are defined. These WAPs requirements are necessary for SPs in order to be able to use these WAPs as reference positions to define their locations, while the SPs are moving indoors.

In the previous chapter, it has proved that WAPs clocks can be synched accurately with SPs clocks within few nanoseconds (i.e. 6 to 10 nanoseconds). This process has been conducted through using the obtained GNSS time on outdoors-SPs. However, on-the-go solution to define WAPs position information is remained unsolved to design a seamless outdoors-indoors SP positioning solution.

This chapter presents a proposed WAPs localisation scheme via any outdoor-available GNSS-enabled SPs in order to tackle the issue of on-the-go solutions to locate WAPs. The scheme starts when an SP is outdoors and has already maintained GNSS position. As this SP approaches a building, it might receive WAPs signals from inside the building. Through the received signals, all the WAPs pseudoranges could be measured. Then the SP can compute these WAPs locations through using Trilateration based on having the SP's instantaneous geographical locations and the pseudoranges with WAPs. Note: the accuracy of pseudoranges measurement depends on the SPs-WAPs clocks synchronisation via using the obtained GNSS time on outdoors-SPs (as proved in **section 3.5.2**). After that, the located and synchronised WAPs can act as reference points for calculating the indoors-SPs location, particularly when GNSS signals are not available. As a result, such scheme can offer seamless outdoors-indoors localisation, when it is installed on SPs at large scale via several SPs and WAPs. In addition, the fresh calculated WAP location can be transported into a central-server-database on the fly from any participating SPs. This transported WAP location could be shared as a beneficial aspect for saving cost and time-involved of using purpose-built-vehicles to survey WAPs location [117].

This chapter contains five main sections as follows: **section 4.2**, the main problem of WAPs location estimation through using various localisation techniques. **Section 4.3**, “WiFi-Based Localisation Solutions” which describes how can Wi-Fi help indoors-SP localisations? And the sixth research question (as it is pointed out in **section 1.4**): “how can synchronised-WAPs be accurately located?” is addressed in. **Section 4.4** includes related work and the design/prototype implementation details of the proposed WAPs localisation scheme. **Section 4.5** demonstrates test and scenarios of the proposed scheme and the obtained results from OPNET. **Section 4.6** summarises the achievements and advantages of the proposed WAPs localisation scheme that can be implemented on SPs.

4.2 Problem Statement

WAPs are usually available everywhere for accessing the Internet services. It means WAPs originally are not deployed to be used as a reference positions for SP localisation. So, if the WAPs are not designed for localisation applications [118] (as using dedicated HW [50] and WAPs position accurately calibrated [117]), they cannot offer accurate position calculation and do not know their geographic position.

As it is concluded in **section 2.4.2**, most of the current indoors-SP localisation solutions to provide SPs position (based on WiFi technology) use pre-defined WAPs positions or obtained WAPs position information via costly offline training processes. WPS-Skyhook and RTLS-Ekahau are examples of indoors-SP localisation solutions [117]. However, the two assumptions are not applicable practically in many situations because there is no provision of automatic WAP position estimation.

4.3 WiFi-Based Localisation Solutions

Mainly, the LBS-SPs applications use on-board WiFi technology to estimate the position of the user inside buildings where the WAPs signals are available. For example, an SP can estimate the pseudoranges (through using TOF or RSS measurements) via beacon signals coming from each WAPs distinguished through its MAC address, and assuming these WAPs' position are previously known. According to these observations, the SP can perform Trilateration technique dynamically to report an estimation of the SP position. Moreover, the essential reasons of using WiFi technology as an alternative technology are:

1. The LBS users don't need any specialised hardware as all the SPs are equipped with a Wi-Fi transceiver.

2. There is no need to deploy an extra dedicated network as WAPs radio signals can be detected in the majority of areas of interest, due to rapid-increase of using WLAN.

3. Received WAPs-signals' measurements can be utilised to define SP position based on some constraints and calibration algorithms.

However, in order to provide accurate SPs position with using WiFi-time based localisation solution, SPs need to:

1. Develop a synchronisation algorithm which can synch WAPs-SPs clocks (i.e. calculate clock offset and clock drift) with one another within few nanoseconds accuracy.

2. Estimate pseudoranges with WAPs based on the clock error knowledge.

3. Develop an algorithm to obtain location information of the WAPs (which is regularly unavailable).

4. Apply Trilateration to calculate their positions when WAPs location information and pseudoranges are obtained.

In **section 3.3**, it has been illustrated how obtained GNSS time on SP can be used to avoid the necessity of SPs-WAPs clock synchronisation. Then how SPs measure accurate their pseudoranges with WAPs accurately. However, finding these synched WAPs location is remained as considerable challenge. Therefore, in the incoming section, the most recent solutions/schemes for WAPs localisation in comparison with the proposed scheme are examined.

4.4 WAPs Localisation Solution

WiFi technology can be used as an alternative way for defining SPs location, especially in environments where GNSS signals do not exist such as indoors and/or urban areas [96]. To achieve this aim, accurate WAPs location information must be available on the go (without prior special arrangements), as it is required from any WAP in any vicinity. For example, this WAP location information is necessity in existing WiFi positioning techniques such as pseudorange-based techniques (using RSS measurements) [119].

4.4.1 Related Work

Two localisation schemes have been developed by Skyhook [120] and Ekahau [73] to offer WAP based localisation solutions. These schemes use the Fingerprint-technique. In addition, purpose-built-vehicles are deployed to conduct a full WAPs' coordinate survey in built-up areas. When a user wants to locate his/her SP (via using Skyhook), the localisation application will scan nearby WAPs for MAC address, and then sends the collected addresses to a Skyhook's server via an Internet connection. After that, the server calculates the coarse position based on the nearest WAP location stored on its database and sends back the location of the SP to the application. This solution incurs large cost for surveying the WAPs in that location.

A laptop hosted system is proposed to localise WAPs. This proposed system includes a GPS receiver which is used to define reference points, and an external motorised-WiFi antenna that is utilised to detect the direction of the WAP's signal at two different points. In addition, this system works based on the AOA technique as well as calculating the maximum strength value of the received signals. After obtaining WAPs location information, this system displays the location coordinates on a Google Earth application in the host laptop [121]. However, the position measurements have large errors due to fluctuating WiFi signal strength (this fluctuation is varied based on the path of the received packet all the time). To implement this system, SPs have to include extra hardware such as the motorised antenna to detect the WiFi signal direction which is not practical.

Another RSS-based measurement technique to calculate the WAP pseudoranges from a GPS enabled device has been published recently [118]. This is similar to our proposal in treating with the SP, located by the onboard GPS receiver, as a reference position to locate WAPs. However, this technique measures the pseudoranges based on heuristic algorithms to assure the WAP position. Similarly, the strength of any WAP signal is not stable and its fluctuation is depended on the used WiFi protocol. The published simulation results showed that, when the SPs are outdoors, the WAPs are located within 5 meter error with probability of about 80%. That is, the accuracy of WAPs' position will be worse for indoors SPs' localisation. In addition, carrying out such heuristic calculations on a SP will drain too much battery-power. An enhanced attempt was proposed through establishing a relationship between RSS and pseudoranges of WAPs from SPs. To assure the accuracy of the pseudoranges, a linear regression is applied to estimate linear coefficients which are used to perform Trilateration [122]. This solution works even if there is no available information on the WAPs' signals such as path loss exponent and transmission power. However, when the geometric relationship between the WAP and SPs is poor (worst case), the estimated coefficients would be variable and the position error becomes relatively large.

This research has concluded that WAPs signals strength has a signal propagation issue, therefore any pseudorange estimation based on RSS would not be accurate. Likewise, reflection and multipath affect RSS based localisation accuracy. Thus, to mitigate such errors, some computational mechanisms are needed such as the local signal strength gradient algorithm [123]. This algorithm claims to improve RSS-positioning technique accuracy. The idea of the gradient algorithm is to find the direction of the received signal by pointing arrows towards the WAPs. However, this algorithm works based on using heuristic or estimated values (such as estimating a window size for any measurement) which are varied in different locations. Furthermore, if the window size is too small, the result of the direction estimation is likely to have a huge error.

The proposed scheme, in this research, does away with RSS by locating WAPs via SPs, when outdoors, using the TOA technique which has been proven to offer more accurate location [31]. In addition, GNSS time is used to assure the accuracy of TOA through having both the WAP and SP sync for the same time. Finally, the scheme does not require any extra hardware for performing this localisation. As a by-product, a central server can be deployed, similar to Skyhook server, to store the location information of WAPs in a dynamic way. This means, WAPs' location information gets updated by nearby SPs, instead of using dedicated signal-survey vehicles.

4.4.2 Proposed WAPs Localisation Scheme

To achieve WAP positioning in an on-the-go solution, this research proposes a WAPs localisation scheme via SPs' GNSS information. The scheme works based on the following steps:

1. When a GNSS-enabled SP (or more) moves outdoors in the vicinity of any WAP and acquires a GNSS fix ($x_{i,}y_{i,}$ and $z_{i,}$), it can receive the WAP's beacon signals.

2. This SP can measure TOF and then its pseudoranges with the WAP through using equation (4.1), by obtaining the timestamp of the recieved (T^{i}_{wm}) and transmitted (t^{i}_{wm}) beacon signals. The pseudorange is measured based on the clock synchronisation algorithm that has been already implemented based on GNSS-time in chapter 3. Thus, estimated pseudoranges will be precise due to having accurate WAPs clock offset and clock drift (b_w).

$$p_i = TOF_i * c \qquad (4.1)$$

Note: the TOF_i is equal to the $(\mathrm{T}_{\mathrm{wm}}^{\mathrm{i}} - \mathrm{t}_{\mathrm{wm}}^{\mathrm{i}})$, c is the speed of light, i is the index of the transmitted/received beacon signal and w is the index of the WAPs.

3. Then his SP can be considered as a reference position at that instance in time, and together with other SPs (multi-SPs) in the vicinity, the WAP position ($\mathrm{x_w}, \mathrm{y_w}$ and $\mathrm{z_w}$) can be calculated based on the TOA. The basic equations for TOA to define the WAPs position based on Trilateration will be expressed by equations (4.2) and (4.3), for 2D and 3D respectively [45]. Note: 2D system equations are selected for the implemented WAPs localisation scheme.

$$p_i = \sqrt{(\mathrm{x_i} - \mathrm{x_w})^2 + (\mathrm{y_i} - \mathrm{y_w})^2} + b_w \tag{4.2}$$

$$p_i = \sqrt{(x_i - x_w)^2 + (y_i - y_w)^2 + (z_i - z_w)^2} + b_w \tag{4.3}$$

Practically, it is difficult to implement Trilateration process because of: 1) the set of equations is nonlinear, 2) WAPs location accuracy can be further improved by contributing a larger number of SPs rather than the minimum required of SPs. To solve this difficulty, the equations should be linearised. The detailed of this linearisation is explained in **section 2.2.3.1**. Note: **Appendix A.3** shows the c-language code of the Trilateration code to define WAPs position.

Furthermore, if there are no other SPs available in the vicinity at that time, the movement of the single SP will be used to provide several reference positions to calculate coarse WAP position.

Figure 4-1 illustrates these aforementioned steps when the number of reference positions of the SP movement can be predefined as a threshold. Note: the threshold should be greater than 2 for 2D system and greater than 3 for 3D system. In addition, the SP(s) compares defined WAPs position all the time to ascertain an accurate position for these WAPs.

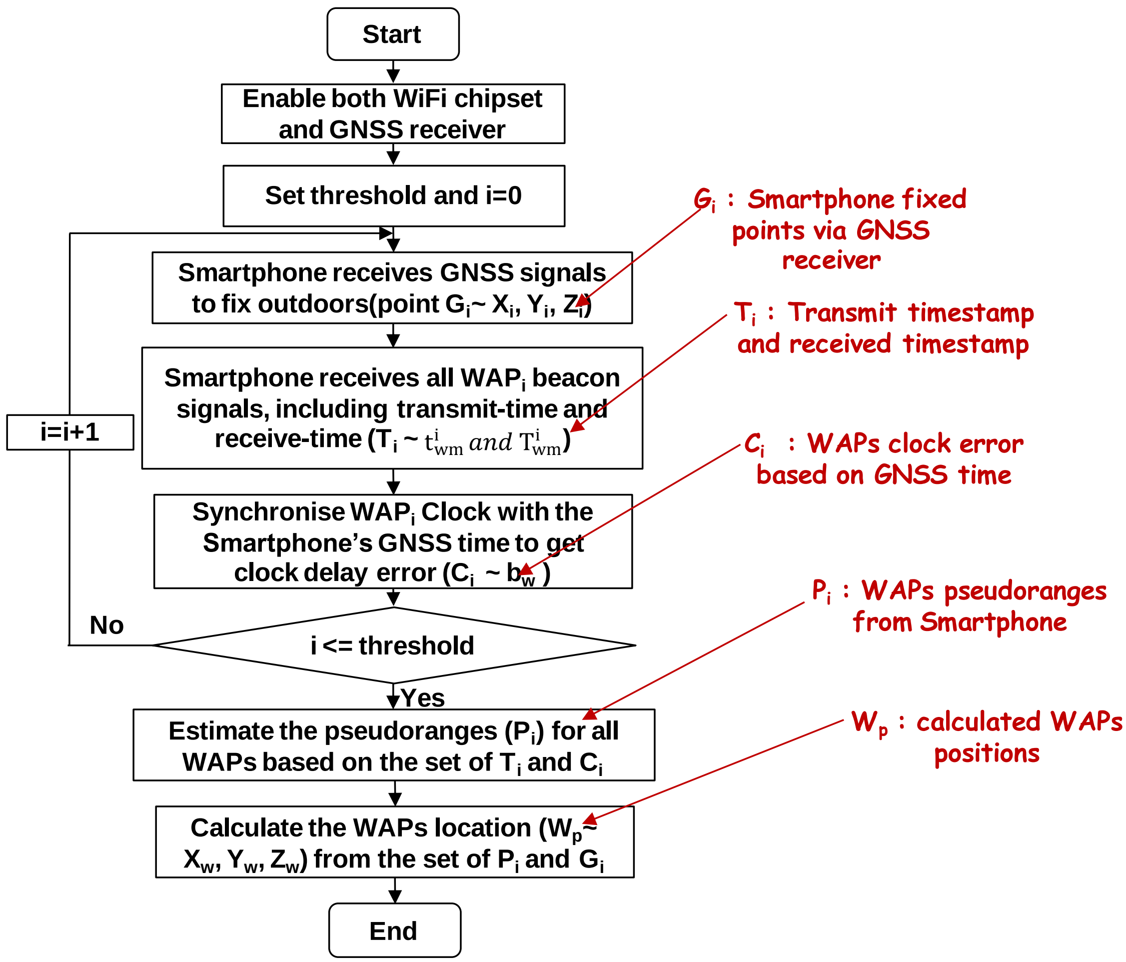


Figure 4-1: Flow chart of WAPs synchronisation/localisation scheme via a SP.

In multi-SPs scenario, a group of SPs contributes to define WAPs' position with a dynamic-replacement of SPs position algorithm which can also mitigate DOP issue. Geometrical condition of the SPs positions should be always considered due to its effectiveness on location accuracy. Especially, when the pseudorange-based techniques is being used. Particularly, inaccuracy can happen in the system due to a bad geometry of SPs positions. In order to show this geometric impact, a metric which is called DOP can be used. This parameter defines a quality of the geometric reference position configuration through using measurement units. For example, a poor geometry condition between a WAP and a group of SPs in the vicinity leads to high DOP value, or vice versa. To mitigate this DOP issue, the scheme implements a dynamic reference positions re-placement algorithm during WAPs location calculation. The algorithm indicates the geographical condition level (by using HDOP parameter, since the scheme is for 2D) when SPs are moved around the coverage area of the WAPs. Then it compensates this effectiveness by using conditioned number of SPs positions.

Then these localised and time-synchronised WAPs can be used as reference positions to aid localisation of the indoors-SPs. Thus, the scheme offers seamless outdoors-indoors SPs localisation on-the-go, anywhere and anytime.

As a result, the proposed scheme does not require dedicated hardware such as host server, sensors, and calibration. These dedicated hardware/sensors are typically associated with indoors solutions such as Fingerprinting [119]. Therefore, the scheme reduces the required memory and traffic on SPs so as to saving battery consumption, connection/interaction traffic and processing time. In addition, as a by-product, a main server is capable of being deployed to save the information of WAPs' location as they get update through nearby SPs.

4.5 Test Scenarios, Simulation, and Results

To prove the validity of the scheme, OPNET simulation is used because of availability of all WAPs functionalites at MAC layer. Several tests and experiments have been conducted as they are illustarted in the following subsections.

4.5.1 Test Scenarios

A single-SP and 4-SPs scenarios are selected, as shown in Figure 4-2 and Figure 4-3 respectively. Single-SP scenario has high DOP issue because the reference points are near each other. Therefore, multi-SPs positioning will be much more accurate due to a lower DOP value (DOP value in TOA-technique depends on geometric condition of the reference position and the target position).

In order to test both scenarios in different geographical anticipation, four tests have been conducted as follows:

1. **Test1:** in the first test, it is assumed that an SP is located in three continuous different positions (as shown in Figure 4-2), and at the same time the SP can synchronise the WAP clock and measures its pseudoranges. Thus, the SP can calculate the WAP's position in a single iteration.

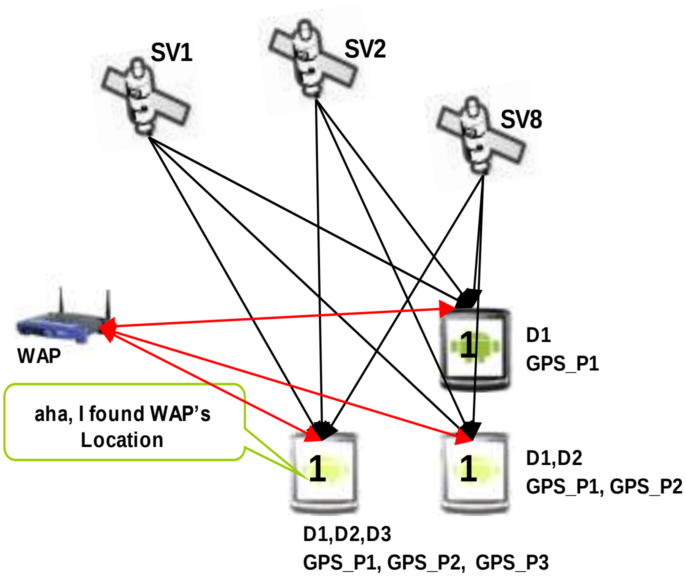


Figure 4-2: Locating WAPs Scenarios via single-SP scenario

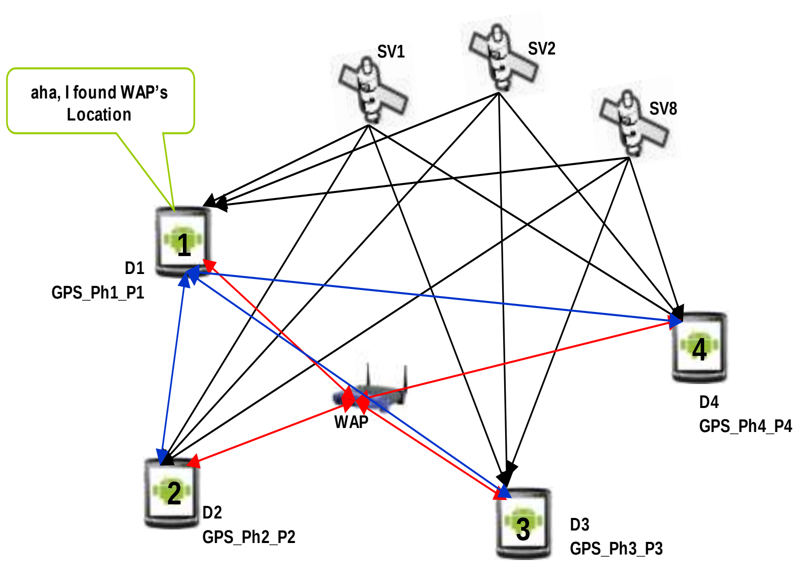


Figure 4-3: Locating WAPs Scenarios via multi-SP scenario.

2. **Test2:** in the second test, the SP has updated GNSS fixed position in seven continuous positions and it has measured the WAP's pseudoranges at each position. From the first three positions, the SP can calculate the WAPs position. Then at the next positions, the SP updates the WAP's position information via the most recent calculation.

3. The third (**Test3**) and fourth tests (**Test4**) are as same as the first and second tests respectively except that the WAP time is re-synchronised with the SP's GNSS clock every 10 seconds.

All these four tests are repeated for the 4-SPs scenario, while the SPs update the defined WAP's position at every position cooperatively (as shown in Figure 4-3).

4.5.2 Simulations Setup

An experiment area of 80m X 80m to simulate tests/scenarios is selected. Figure 4-4 and Figure 4-5 show the simulated test-bed experiments for the first and second scenarios, respectively.

Assumed, there are four WAPs; their signals are available for an SP regardless of multipath/loss signals. The single SP scenario starts to end route travel around the area of WAPs which is shown by the red spots. While in the second scenario, the four SPs start/finish routes travel are at different positions as shown by the red spots with the white lines.

To implement the proposed scheme, the following parameters are configured:

1. **For WAPs:** The WAP node functionalities are configured as in **section 3.5.2**. Such configurations as: constructing beacon frame at MAC Layer, transmitting beacon frame, data rate and timestamp generation for the transmitted beacon frame.

In addition, all other parameters which have been configured for both scenarios at this layer are the same configuration as explained in **section 3.5.2**. The parameters include clock offset, clock drift, and generating local-time. On the other hand, for all tests in both scenarios, two different synchronised WAPs local-times are used separately. This means, the WAPs local-times are synchronised based on Network time or GNSS time.

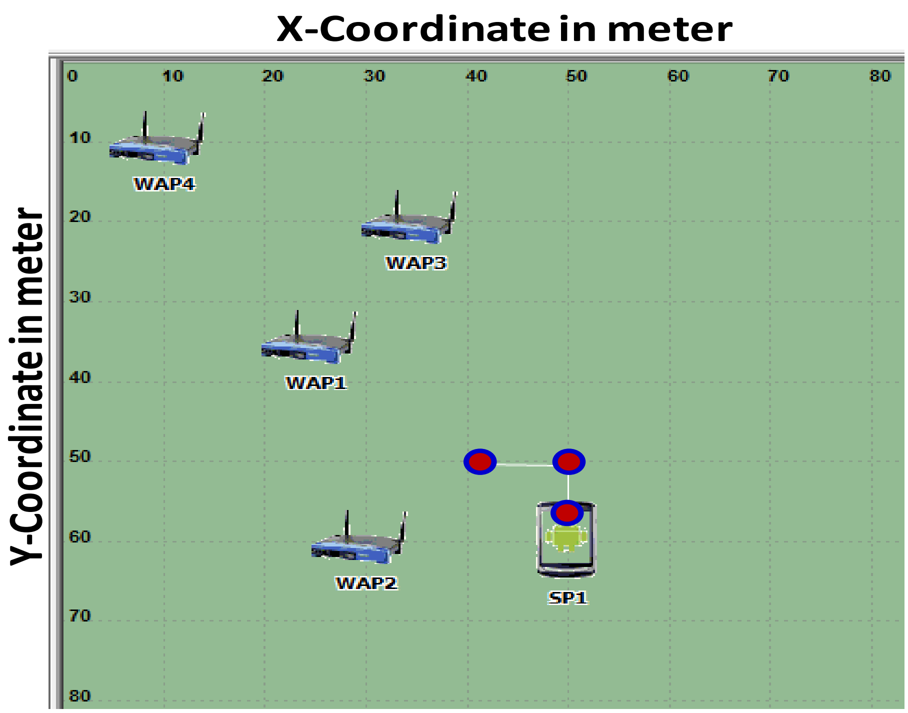


Figure 4-4: Simulated test-bed for the single-SP Scenario.

2. **For SPs:** The same WiFi-node functionalities have been used for: receiving beacon frames and generating timestamps for the received beacons based on GNSS time. So as to define WAPs position, the purpose of Locate_SPs model (**in section 3.5.2**) has been changed to calculate WAPs location. The SP(s) uses this model to define the procedure of Trilateration in iterative LLS and then to calculate WAPs location.

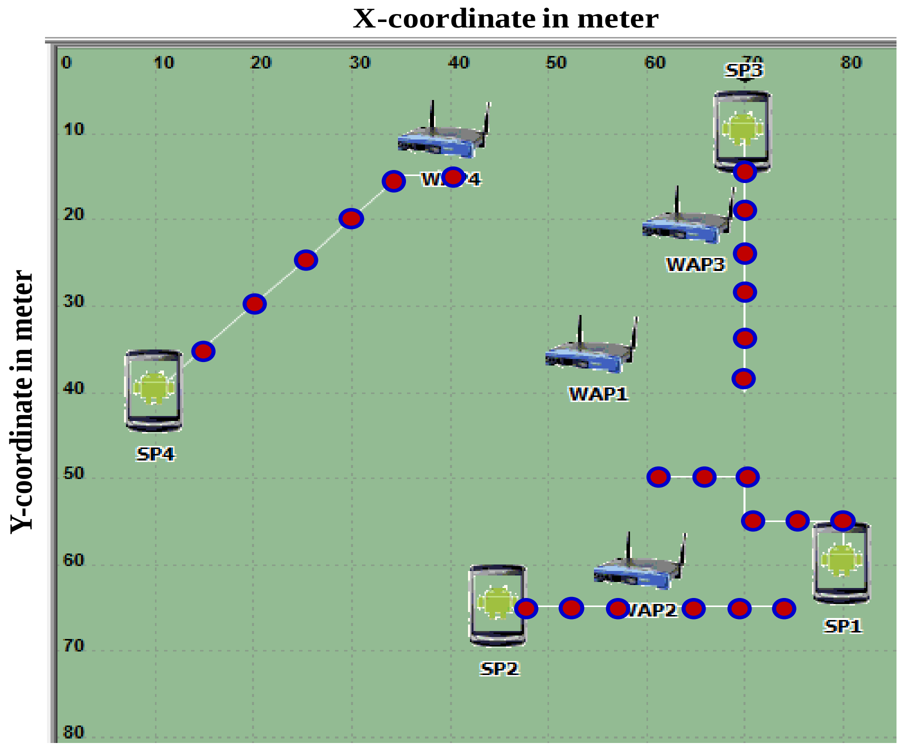


Figure 4-5: Simulated Test-bed for the multi-SPs scenario.

The location accuracy is considered as the performance metric to evaluate the WAPs synchronisation/localisation scheme in all scenarios and tests. The detailed of evaluation performance of the scheme within all the tests is discussed more in **section 4.5.4.**

4.5.3 WAP Clock Model

An accurate WAP clock model is needed to simulate the WAP localisation scheme within SPs-WAPs clock synchronisation algorithm in OPNET. Therefore, the DWC model is used for WiFi localisation applications so as to measure the WAP clock offset and clock drift accurately. This means the clock delay error relies on real clock noise sources of the WAP clock and using GNSS time (as a reference time). The clock model for generating WAP clock time can be computed through using equation (3.3), **see section 3.3.3**.

In both scenarios, the clock model is used to generate all WAPs clock time. In order to prove the novelty of using GNSS-time as an accurate reference time, a simulated network time (Net-time) is employed. The simulated Net-time is represented by synchronising all WAPs clock with a single/reference WAP clock in the network. So, both GNSS-time and Net-time are utilised as a reference time to synchronise WAPs clock time separately. This WAPs clock time is invoked to stamp the generated beacon frames. Note: it is assumed that the WAPs' clock offset has a static value in all cases. This means each WAPs clock delay error includes only the clock drift.

Figure 4-6 shows the WAP clock delay error generated for a single WAP, when the WAP clock is synchronised based on GNSS time and the Net-time. As it is shown in the figure, if GNSS-time is used to synchronise the WAP clock, the WAP clock delay error will be more stable. Similarly, Figure 4-7 shows the same behaviour of the WAP clock delay error, while WAPs' clock are resynchronised at every 10 seconds for the third and fourth tests. Note: re-synchronising or refreshing WAPs clock periodically has mitigated the effect of the WAPs clock delay error on this localisation scheme.

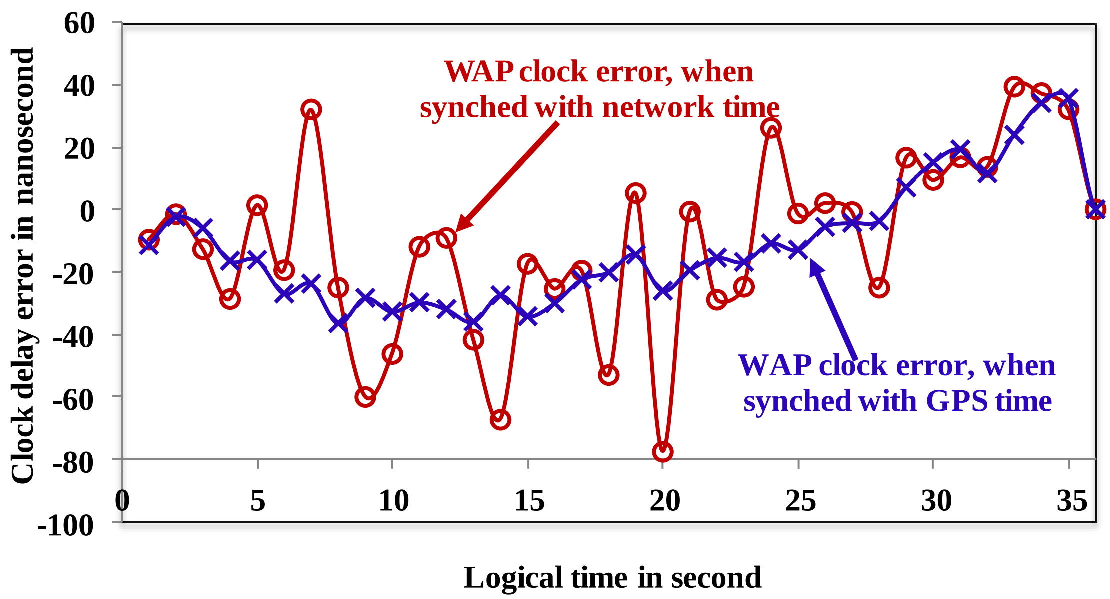


Figure 4-6: Generated a reachable WAP clock delay error based on GNSS time and Net-time, without re-synchronisation.

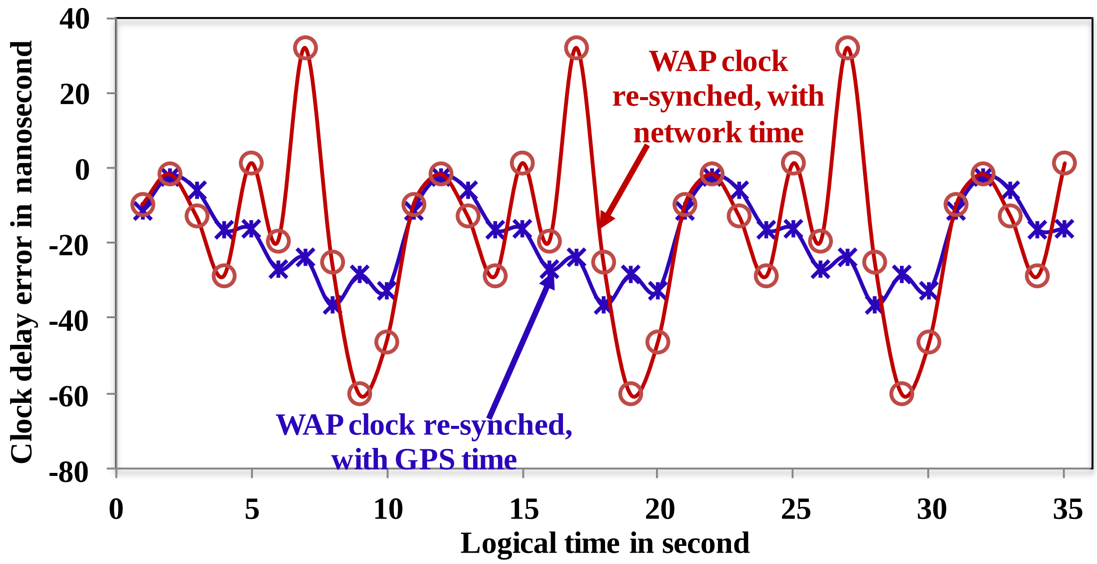


Figure 4-7: Generated a reachable WAP clock delay error based on GNSS time and Net-time, and resynchronised at every 10 seconds.

4.5.4 Evaluation of WAPs Positioning Accuracy

Table 4-1 shows the results of the WAP location accuracy through running all the tests in both scenarios. The first test accuracy is the worst, while the best accuracy is achieved in the fourth test. The results of the simulated scheme indicate that the multi-SPs scenario is more accurate than the single-SP scenario. This is due to having poor geometric condition in the single-SP scenario. Likewise, the best accuracy achieved was 0.9 meter through using GNSS-time, while it is 3.93 meter if the synchronisation depends on Net-time. This is because, in Trilateration, if more SPs are contributed as reference positions to locate WAPs, high accuracy will be achieved and vice-versa. In addition, if we look at on the results from tests of single-SP, there is no a reasonable improvement, while the results from tests of multi-SPs show that huge improvement can be obtained. This is due to having more SPs as reference positions (good geometric condition) to calculate WAPs locations.

Finally, better accuracy of WAPs location can also be obtained when the GNSS-Time is used as reference time for the clock synchronisation in all cases. In addition, the results show that, if the WAPs clock is re-synchronised every 10 seconds, the accuracy of WAPs' position will be improved more in both scenarios.

Table 4-1: Results of WAP localisation accuracy in different scenarios/tests.

Scenarios	Tests (as described in 4.1.1)	Mean of WAPs location accuracy in meter		STDV of WAPs location accuracy in meter	
		Synched based on GNSS time	Synched based on network time	Synched Based on GNSS Time	Synched based on network Time
Single-SP	Test1	**12.301**	**24.047**	**5.208**	**17.189**
	Test2	**12.536**	**17.838**	**4.609**	**14.960**
	Test3	**6.3**	**16.058**	**6.592**	**8.793**
	Test4	**3.985**	**9.368**	**4.295**	**6.009**
Multi-SPs	Test1	**11.862**	**18.958**	**4.203**	**12.610**
	Test2	**10.367**	**16.053**	**5.309**	**6.744**
	Test3	**2.806**	**8.691**	**2.641**	**4.552**
	Test4	**0.918**	**3.930**	**0.898**	**2.167**

4.6 Summary

Locating WAPs via SP is achievable. Storing such information in SPs' memory or on an Internet central-database can make such localisation system fully-automated. This is especially useful when SPs can use this information to locate themselves indoors. This chapter presented simulated-experiments proving the viability of such scheme based on the TOA technique.

The outcomes of the simulation show that the important parameter to achieve good positioning accuracy is the quality of the WAPs clock synchronisation. And in order to obtain high quality of clock synchronisation, an accurate reference time (like GNSS time) is necessary. Simulation scenarios based on OPNET implementation has proved the viability of this scheme.

Furthermore, the accuracy of WAP location by implementing multi-SPs scenario is much more accurate than the single-SP scenario because the DOP issue has been mitigated in the multi-SPs scenario. Moreover, continuous or periodical synchronising of WAPs clock (getting refresh WAPs clock offset and clock drift) can improve localisation accuracy. The simulation results for the fourth test showed an accuracy of the WAPs location within one meter.

Thus, the scheme is a good candidate for seamless LBS localisation applications on SPs without dependency on pre-installed, calibrated infrastructure and/or dedicated Internet based data/server. Furthermore, the novelty of the scheme is similar to GNSS technology, when SPs (like satellite-vehicles) moves around WAPs as well as all location calculation is done on SPs.

Implementing this scheme on SPs, in a particular LBS application, can be used as an auxiliary-step to help indoors-SP localisation. The synched and located WAPs can act as reference points for calculating the indoors-SP location, especially when GNSS signals disappear. However, to locate indoors-SPs, further research should be considered in terms of indoors-structures, achieving accurate indoors-SPs location and WAPs signal coverage. In next chapter, two new indoors-SPs localisation schemes are presented to tackle these issues. For example: 1) SILS can be used for locating indoors-SPs accurately and on the move based on smart/cooperated network of SPs, and 2) UNILS is used for locating indoors-SPs based on using inertial sensor measurements in a network of SPs when WAPs signal is not available.

CHAPTER5. INDOORS LOCALISATION SCHEMES BASED ON ON-THE-GO COOPERATIVE SMARTPHONES NETWORKS

5.1 Introduction

Seamless and on-the-go outdoors-indoors localisation solutions based on SPs devices/sensor are essential to realise the full potential of LBS application. However, on-the-go localisation solutions are not developed and still remained as a challenge. The main problems of SPs location estimation through using various seamless and on-the-go localisation solutions are described in **section 5.2**.

To offer such solutions, this chapter aims at presenting two localisation schemes:

1. SILS forms a BT network Piconet (see **section 5.2**) to locate the indoors-SPs, whereby SPs cooperate in the same outdoors and indoors vicinity. In order to obtain indoors-SPs location, SILS performs three functions: a) implementing the WAP synchronisation/localisation scheme to synchronise and locate all reachable WAPs (see **section 3.4.2** and **section 4.4.2**) via the outdoors SPs, b) exchanging a location information map of all SPs and time-offsets c) and calculating location of indoors-SPs based on hybridisation of GNSS, BT and WiFi measurements. Additionally, the SILS represents: new hybridisation scheme of various on-board SP devices/sensors for indoors localisation. Furthermore, SILS uses only existing WiFi-infrastructure to locate indoors-SP based on any available SPs and/or WAPs in the vicinity. The explanation of the three functions, experiments and testing different scenarios of the SILS are described in **section 5.4**.

2. UNILS introduces DR measurements to the SILS. This scheme utilises only available sensors on SPs including GNSS, inertial sensors and Bluetooth. This means, the UNILS is reliable even when communication between SPs and WAPs/ BT-anchors is considered unavailable. Moreover, the scheme combines relative-pseudorange measurements (using TOA technique) with distance-displacement and heading measurements (using DR technique) of the networked SPs. This

combination is acquired by exploiting the advantages of both techniques while compensating for their inaccuracy. UNILS fuses the measured relative-pseudoranges between SPs, outdoors-SPs positions and DR measurements of indoors-SPs by using EKF. UNILS has two new distinct characteristics. **Firstly**, new indoors localisation scheme does not require dedicated infrastructure and capitalises only onboard SPs devices. Examples of these devices are: inertial-sensors, BT and GNSS devices. **Secondly**, new fusing algorithm hybrids various localisation measurements to provide low cost, on-the-go, and location accuracy to within 3 meters, when SPs are deep indoors. This scheme can be deployed on any SP via a simple localisation application. The details of UNILS concepts followed by its implementation as well as obtained results from OPNET simulator are presented in **section 5.5**.

5.2 Blutooth Network (Piconet)

Group of Bluetooth devices (up to eight devices) are usually forming a network called piconets. The piconet includes a single master device and one or more slave devices. In addition, a dvice may be connected with more than one piconet, either as a master of one piconet and a slave in another or as a slave in both.

Bluetooth, like WiFi, operates in the unlicensed ISM frequency band, generally cluttered with signals from other devices: garage door openers and microwave ovens, to name just a few. To help Bluetooth devices coexist and operate reliably alongside other ISM devices, all Bluetooth slave-devices in a piconet is synchronised with to a specific frequency hopping squence. Note: the frequency hopping sequences are generated based on the master’s clock and MAC address. This pattern, moving through 1600 different frequencies per second, is unique to the particular piconet. Data packets are transferred based on each selected frequency hop within a slot time. One slot time in Bluetooth is equvilant to 625 microseconds. However, a packet may actually span up to five time slots, in these cases the frequency remains constant for the duration of that five time slots.

For localisation view, utilising such synchronised frequency hopping is preferred to exchange messages and then to calculate pseudoranges between connected devices in the piconet.

5.3 Problem Statement

On-the-go SPs based seamless outdoors-indoors localisation is essential to realise the full potential of the most LBS applications. The literature survey (see **section 2.5**) has proved that WiFi, BT and/or inertial-sensors technologies provide alternative solutions in GNSS-signal-denied areas (indoors) to define SPs location. However the following issues are some of the main challenges to design an on-the-go and seamless localisation solutions:

1. Restricted coverage and lack of information of physical reference-positions (WAPs/BT-anchors) within buildings.

2. No localisation protocols.

3. Available localisation techniques on SPs (as a stand-alone mode) do not provide high localisation performance. For example, RSS-based techniques provide good positioning accuracy. This is at the expense of pre-installed dedicated infrastructure. Pseudorange-based techniques suffer from jitters, instability, coverage and DOP issues. Further, the DR technique offers smooth location accuracy but its performance is quickly degrades over few seconds. This is due to the drift and noise of the utilised inertial sensors.

4. Current developed localisation solutions provide the precision and range demands of most LBS applications. Nevertheless, the infrastructures (such as those deployed by BLE-iBeaconing, WPS-Skyhook and RTLS-Ekahau) do not exist in today’s indoor environments. Moreover these infrastructures are not necessary for anything other than SPs-positioning. That is, such developed solutions incur huge cost [124].

Therefore, as an alternative solution, that utilising combined localisation techniques and utilising cooperation of a group SPs are suitable to solve the aforementioned issues. That is our new indoors-SPs positioning hypothesis.

Through localisation infrastructure-based solutions, the localisation measurements are between SPs, reference devices (like WAPs or BT-anchors) and dedicated servers. In other words, there is no communication between SPs. SPs need to achieve location information from one or more reference devices. In addition, SPs need either a wide coverage area for each reference devices or deploying large number of reference devices [125].

While through cooperative localisation solutions, the location of an SP can be defined by doing peer-to-peer communications with other SPs (that have already fixed location) in the vicinity. It means the request of communication between SPs and multiple fixed reference devices is not needed [126]. Therefore, in the cooperative localisation strategy neither reference devices to provide large coverage areas nor high density of fixed reference devices is required. Thus, cooperative localisation can provide unconstrained solution.

5.4 SILS Using Onboard BT, WiFi and GNSS

This section addresses the seventh research question (as it is listed in **section 1.4**): how accurately can indoors-SPs position be defined using hybrid onboard SP wireless technologies? Independent of pre-installed localisation infrastructure networks and on-the-go seamless outdoors-indoors navigations are essential functions for many LBS applications.LBS applications will be restricted due to the GNSS signals blockage and deploying large number of sensors (incurs huge cost). So as to avoid this restriction, there is a pre-installed localisation infrastructure of sensors in the vicinity that works with the current SPs. In addition, most of the current solutions attempt to hybridise multi-GNSS signals (GPS plus GLONASS) with cellular, WiFi, BT, and inertial sensors on SPs to offer accurate/seamless location. However, on-the-go localisation solutions, like SILS, are not developed and still remained as a challenge. That is, the provision of semi-accurate indoors SPs location, on-the-go and anywhere anytime, has proven somewhat problematic to deliver thus far [24].

Figure 5-1 illustrates a typical scenario used to describe SILS. It shows a cluster of cooperative Android-based SPs outside of Franciscan-building in Buckingham. These SPs use enhanced GNSS localisation algorithms, such as Google's Maps-matching to achieve accuracy to within 1 meter [93].

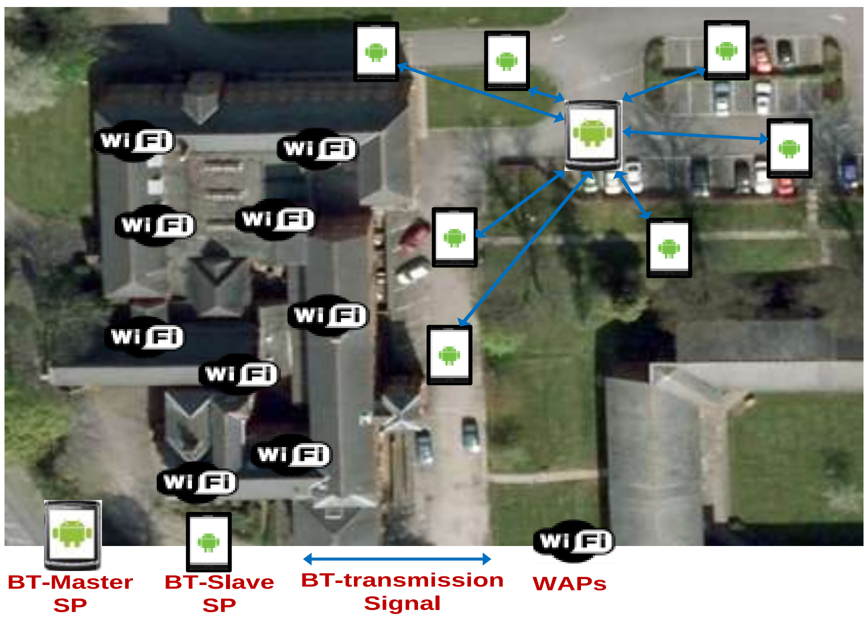


Figure 5-1: SPs outdoors network scenario.

To prove this assumption, during this study, two set of trial experiments have been performed in near indoors (around a building) on actual Android-based SPs. For instance, two Samsung Galaxy SII and SIII are used to collect the estimated location). During the first experiment, only GPS signals are acquired to locate the SPs. That is, without adding localisation information. The average result of three trials shows that the obtained SPs location accuracy is up to 20 meters, as shown in Figure 5-2. Accordingly, in such environments some methods of accuracy improvement are required. Google maps-matching technique, aided information including position/velocity/time via AGPS systems and/or multi-GNSS signals are all used to improve location accuracy. To show such improvement, the second experiment uses multi-GNSS signals (such as GPS and GLONASS signals) with aiding maps-matching information. Figure 5-3 demonstrates the average result of this experiment. As it is observed, the accuracy is within 1 to 2meter.

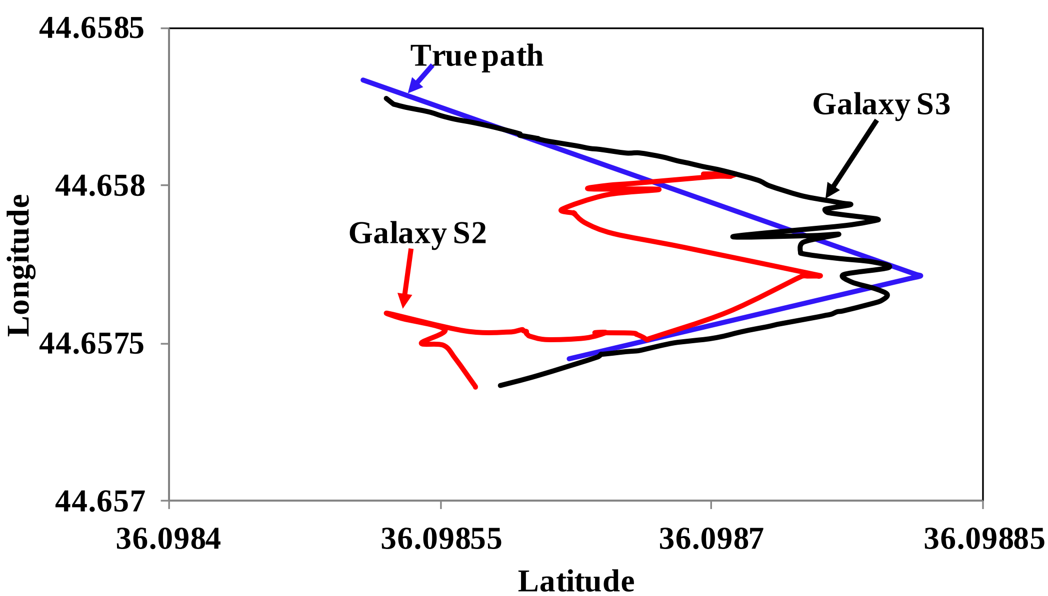


Figure 5-2: Estimated location of two SPs using only GNSS without any improvement-algorithm.

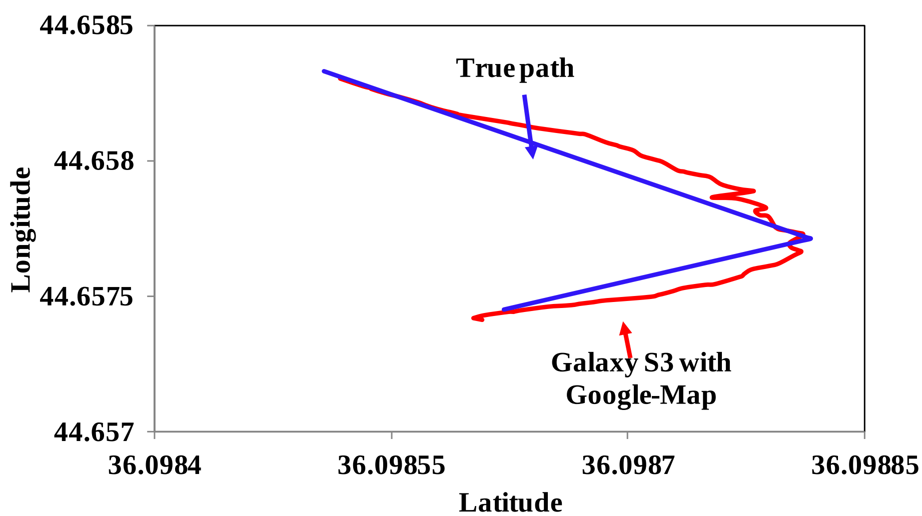


Figure 5-3: Estimated location of an SP using GNSS and Google-Maps matching.

When the outdoors SPs obtained GNSS fixed location, SILS then forms a smart BT network (Piconet) of the SPs. The Java code for Android-based SPs of this constructing Piconet is listed in **Appendix C.1.** The scheme then calculates the relative-pseudoranges between all SPs. Later on, it compares the pseudoranges to the GNSS obtained location. These are to form a virtual localisation map of this network. This map should be stored on SPs memory and shared with each SP in this network and/or newly joining SPs to the network. The map, in next step can be used as reference positions to synchronise all WAPs in the vicinity of the building. That is, the WAPs clock offset can be calculated from the received beacon signals from these WAPs. Further, the WAPs are positioned at assumed altitude on every SP on this network, as shown in Figure 5-4. This information may be required later for estimating initial position if any of the SPs in the network is in deep-indoors. Note: WAPs time is measured from the WAPs beacon signals. Such measured times that are generated at the millisecond level and received at the SPs in the MAC layer to avoid local interlayer nanoseconds delays. Once time synchronisation is achieved with more than 4 SPs, the SPs then try to locate every WAP based on Trilateration. This process is claimed and evaluated in chapter 4 (see **section 4.4**).

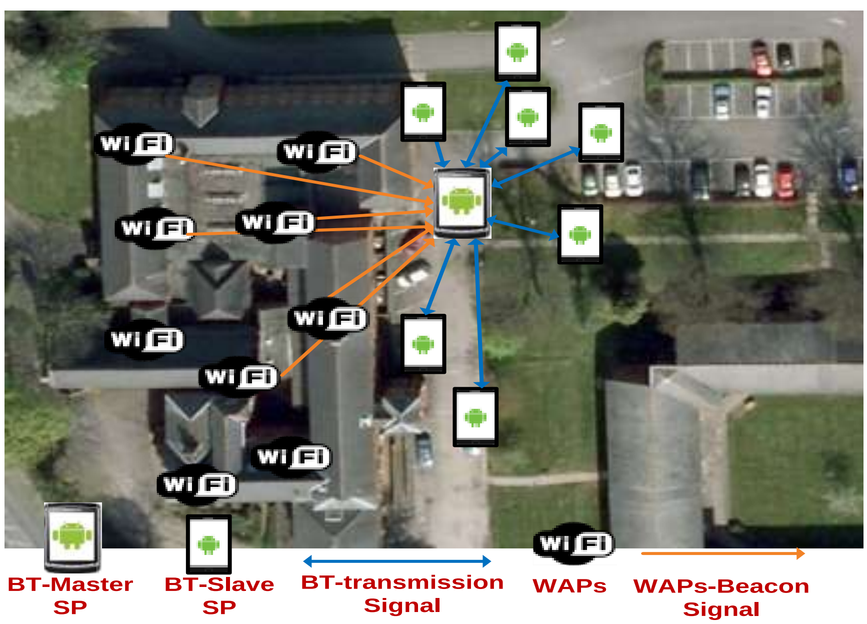


Figure 5-4: SPs outdoors network scenario showing available WAPs.

This SPs network is on the move all the time. In addition, some connected SP may be lost or new ones may join the network (within BT coverage range). SILS locates any of the networked SPs entering indoors (as shown in Figure 5-5) or if any existing indoors-SP joining the network, while at least 4 SPs are still located outdoors with good sky view. SILS locates the indoors-SPs to within 1 meter when at near-indoors and to within 2 meters when in deep indoors. Locating indoors-SPs via SILS is based on TOA technique.

BT communication is based on frequency-hop synchronisations [127].Utilising this synchronisation to count the hop-frequencies and make timestamp at that epoch to measure TOF for each frame transmission could be achieved precisely.

The main aim of the SILS is to offer an SP based localisation solution. That solution capitalises on existing infrastructure with no need for special external hardware/sensors. Even though, the SILS offers high localisation performance/accuracy at low cost.

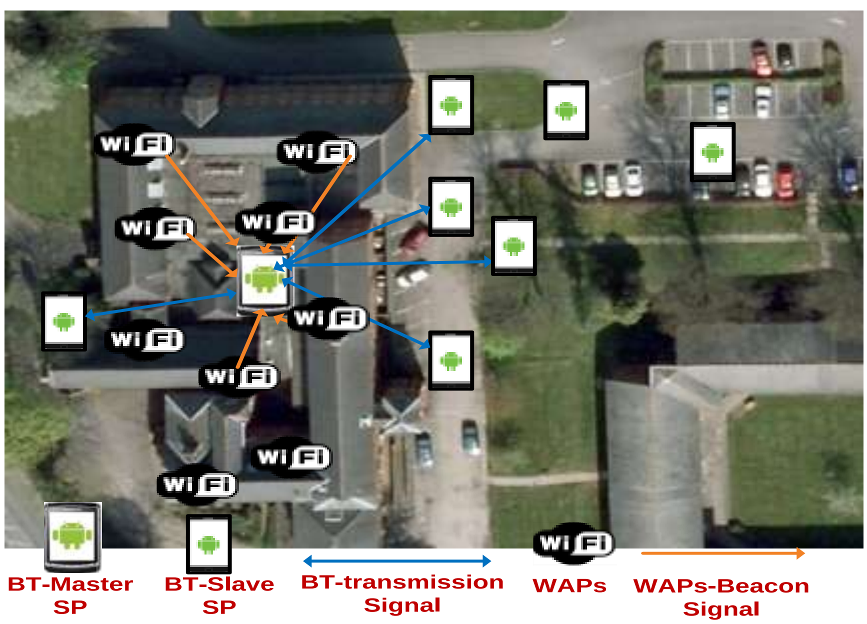


Figure 5-5: Collaborative SPs network to locate indoors-SP using WiFi, BT, and GNSS.

5.4.1 SILS Implementation

The SILS works in the following steps:

1. SILS first detects and forms a BT-Piconet on-the-go with any SPs in the vicinity for the localisation purposes and scenarios. These SPs would be cooperatively running the SILS and the required functions. Further, SILS does check if the SPs have a fresh GNSS fixed location. Thus these SPs assume outdoors SPs. The SPs can accurately obtain their geographical location using their GNSS-enabled receivers. The GNSS receivers are aided by whatever needed to obtain location within 1 meter accuracy, for example with multi-GNSS signals and Google's maps-matching.

2. The next step is synchronising WAPs clock time to GNSS time and defining the location of these WAPs within WiFi range to this SPs network. This is performed by using various outdoors SPs as reference positions for doing TOA calculation based on the beacon signals of these WAPs. In **section 4.5.4**, the WAPs synchronisation/localisation clearly showed that WAPs location accuracy of less than 1 meter could be achieved. As it is observed, the accuracy of WAPs positions depends on the number of participating SPs, SPs' position accuracy and geographical spread of their positions.

3. The refined WAPs positions are shared between all SPs in the network. An iterative procedure is applied to unify the position amongst the achieved positioning at regular intervals as desired. Currently, fresh position calculation is performed whenever a new SP joins the network. SILS uses the WAPs location information to help localisation of indoors-SPs that are at deep indoors with less than 4 outdoors-SPs in the network.

4. When any SP moves indoors from now on it is called SPm. SILS calculates SPm's position based on TOA technique using the outdoor SPs (from now will be referred to as SPos) as reference positions. Three algorithms based on BT-to-BT connectivity are used to assure the location accuracy of SPm: (these are described in the subsections below)

a. BT-to-BT pseudorange estimation algorithm based on hop-synchronisation between SPs, (as it is explained in the incoming **subsection 5.4.1.1**).

b. Switching BT master-slave role algorithm to reduce the error of the pseudoranging measurements, (as it is explained in the incoming **subsection 5.4.1.2**).

c. Permutation reference positions algorithm of outdoors-SPs to mitigate the impact of DOP issue. (as it is explained in the incoming **subsection 5.4.1.3**)

5. When an SPm is deep indoors and only 4 or less SPos are available in the BT network, the SILS uses the WAPs as reference points to help locating this SPm. Pseudoranges between such SPm and WAPs are calculated by using time synchronised beacon signals in SP monitor mode. This location is now optimised with pseudoranges calculated with SPos.

5.4.1.1 Pseudorange Measurement via BT Signals

This subsection addresses the eighth research question (as it is listed in **section 1.4**): Can BT-Hop synchronisation between the connected SPs in a BT-network provides pseudorange measurements? To the best of my knowledge, the Hop-synchronisation counting would be the most accurate time measurement. This is concluded by after empirically experimenting with various methods to do the pseudorange measurements between two BT nodes. This is because, in a Piconet, when the connections between Master and Slaves have been made, both Master and Slaves generate a set of frequency sequences. This set of frequency sequences is called Hop-Sequences. These Hop-sequences' values are generated by mixing of Master's clock offset and MAC address. It means that Slaves can use the same Master's clock to count the hop-frequencies and then synchronise with that clock [128].

Therefore, when a frame is being transmitted, both Master and Slaves should hop from one frequency to the next selected frequency at the same time. As illustrated in Figure 5-

6, time stamping of the epochs of all frequency changes should be precise when an accurate source time (like GNSS receiver time) is used. The pseudorange measurement algorithm utilises modified POLL-NULL frames to calculate the TOF for every BT transmission. The master periodically broadcasts POLL frames to check the connectivity with its slaves and each connected slave responds the master by sending NULL frames [129].

In Figure 5-6, master generates T1 and T3 as timestamps (epochs) during sending POLL frames and receiving the NULL frames, respectively. To measure the TOF, T3 is subtracted by both T1 and Δt delay, where Δt delay is the static time processing delay of the received frame plus one slot time [129]. Then the pseudorange between the master and any slave is equal to the TOF multiplied by speed of light. The c-language code of this algorithm in the OPNET modeller is listed in **Appendix B.1.**

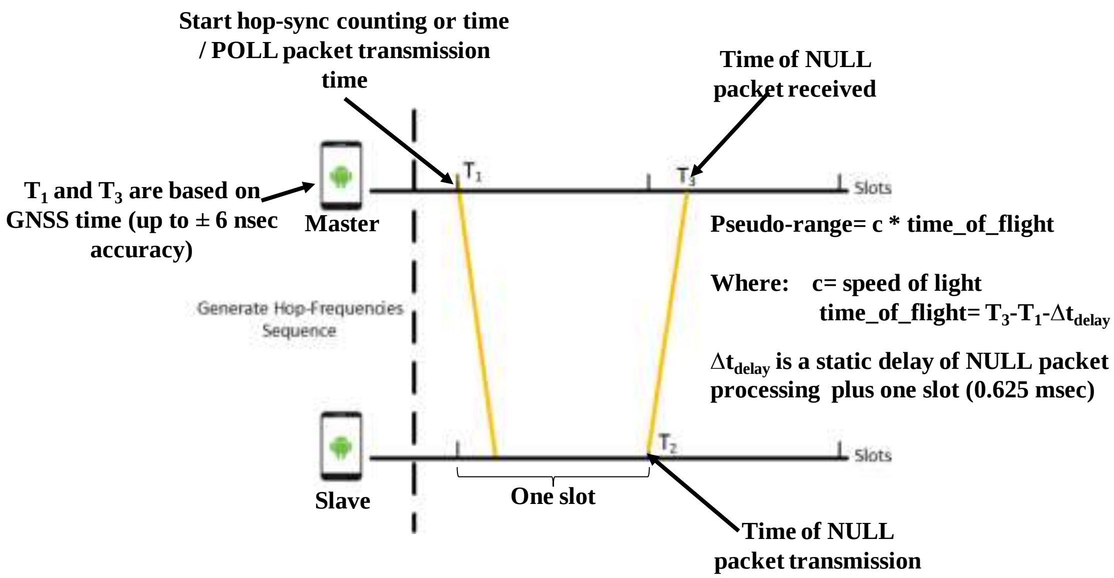


Figure 5-6: Pseudorange measurement using BT hop-synchronisation.

5.4.1.2 New Switching Master/Slave Role Algorithm (SMSR)

In the following steps, SILS applies a new SMSR algorithm between all SPs in the Piconet to reduce the error of all pseudoranges measurement. To achieve this improvement, SPm has to be the master of this Piconet and then generates a map of network location information. This map would be shared by all network SPs, for storing several measurements described in the following steps:

1. As a first step, SPm measures and stores its pseudoranges from all SPos in the network based on the hop synchronisation counting. Later, it collects and stores all SPos GNSS positions on its memory (as shown in Figure 5-7). These position coordinates are appended into the "reserved bits" of the NULL frames that which are sent from the SPos to the SPm. Finally, it generates a list of master-slave switching sequence based on the measured BT-RSS values from all SPos. In most cases, this list guarantees that switching rotation will be based on nearest to furthest order.

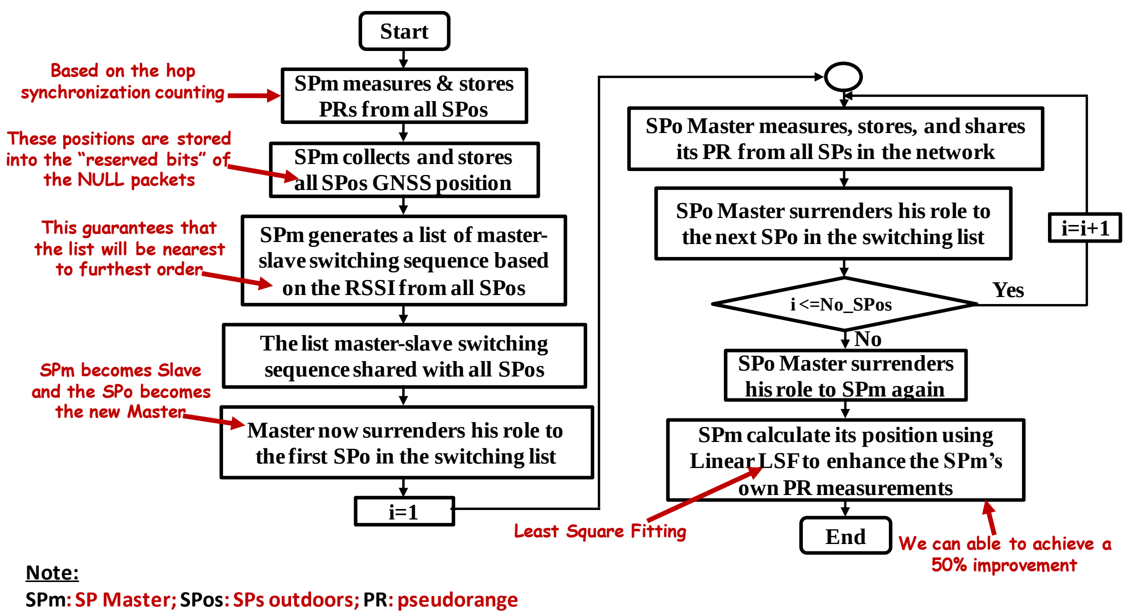


Figure 5-7: Flow-chart of SMSR steps

2. The location map is then shared with all SPos and stored on SPs memory.

3. This step includes:

a. The SPm surrenders his role to the first SPo in the switching list. That is, the SPm becomes slave and the SPo becomes the new master.

b. The new master measures and stores its pseudoranges from all SPs in the network.

4. Again the new updated map of location information through the new SPo is shared between all the SPs in the network

5. This SPo (within master role) will now surrender his role to the next SPo in the switching list, and so on until the order reached the SPm again.

6. The SPm is now equipped with all the estimated pseudoranges from all switching master-slave sequence list. This SPm calculate its position by using a LLS fitting technique to enhance the SPm's own pseudoranges measurements. Through conducting many experiments via known SPs positions; the proposed SMSR algorithm is able to achieve a 50% improvement to the accuracy of the SPm position calculation.

A diagram of running SMSR algorithm on SPs is illustrated in Figure 5-8. SPm then performs the permutation algorithm described in next subsection to enhance its calculated position.

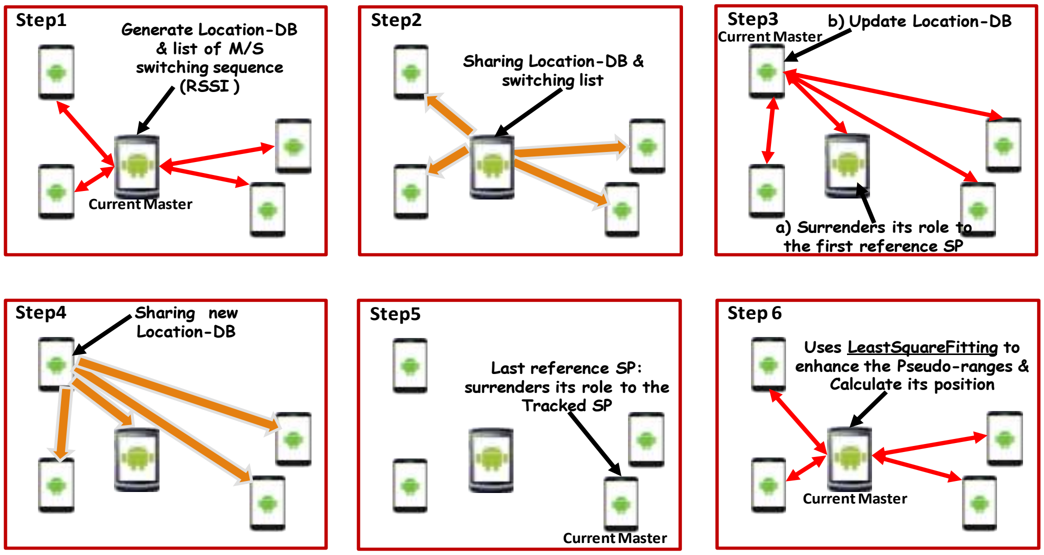


Figure 5-8: BT switching master/slave role between connected SPs in a network.

5.4.1.3 New Permutation Reference-Positions Algorithm (PRP)

While SPos are in bad geometry shape (DOP issues), SILS implements the following PRP algorithm to mitigate the error caused by Trilateration calculation, as shown in Figure 5-9:

1. SPm calculates an HDOP value for the current constellation of the network. This constellation is assigned with two sets of weight values.

2. An iteration process starts by omitting the GNSS position and pseudoranges of each SPos at a time to calculate a new SPm location. New weight values are assigned at every each iteration. This procedure is achieved by a training process that changes the current weight values based on mean and min-difference statistics associated with each position calculation iteration.

3. The appropriate final position is determined by the resultant HDOP and the set weight values.

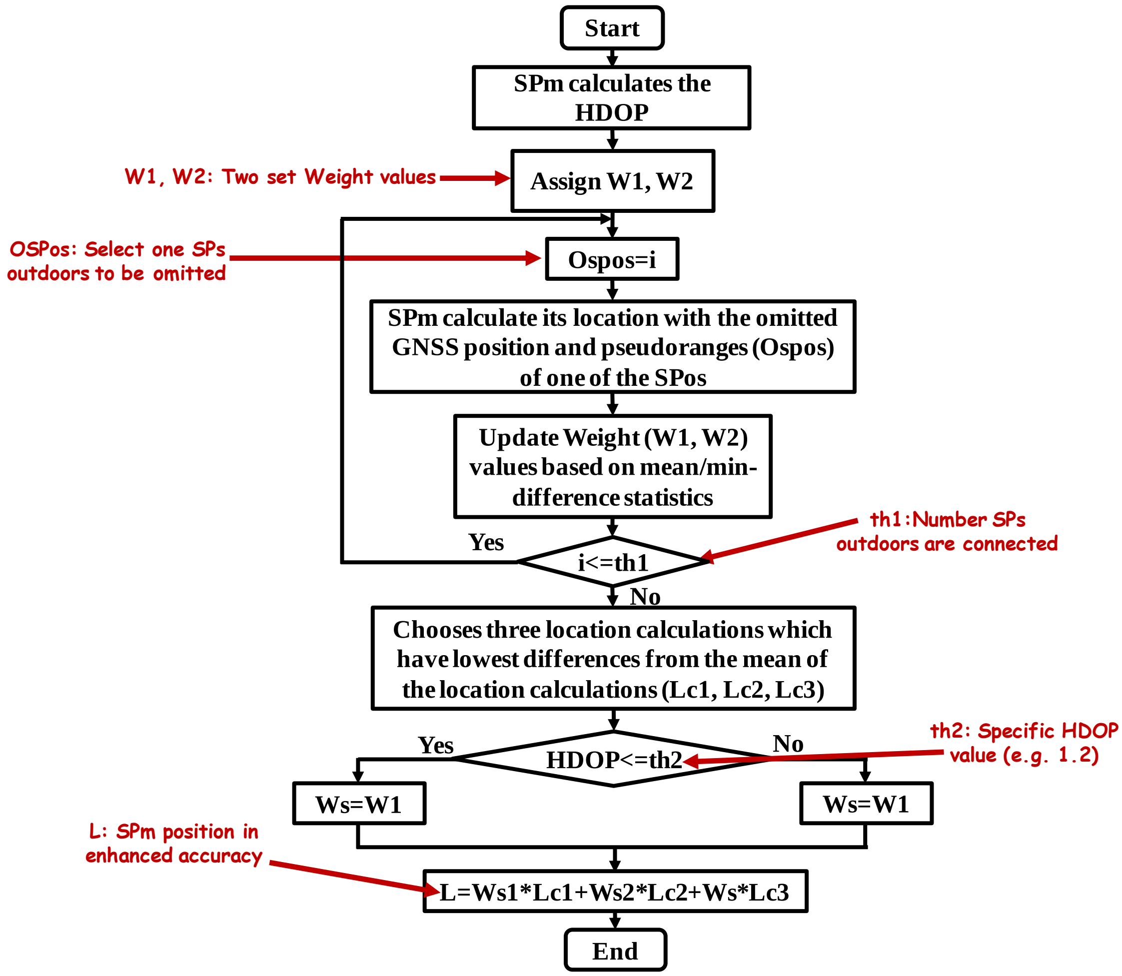


Figure 5-9: Flow-chart of PRP steps.

The diagram of the full permutation reference-positions functionality is illustrated in Figure 5-10.

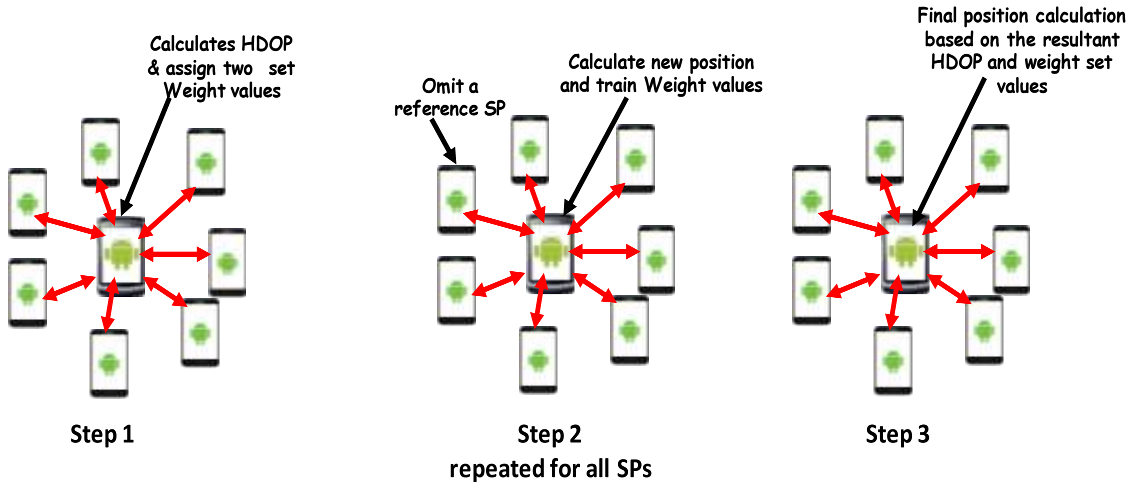


Figure 5-10: BT switching master/slave role between connected SPs in a network.

5.4.2 Test, Scenarios, and Simulation Results

To prove SILS, a scenario is simulated through using OPNET, where a group of 8 nodes as (SPs) start to run SILS in the outside of the building. One of these SPs moves indoors through light indoors area (signals are crossing 1 wall from the outside) and deep indoors (signals are crossing 3 walls deep inside the building). The movement of this SP is illustrated in Figure 5-11 by trajectory line. The other 7 SPs have moved around but stayed outdoors.

In this simulation scenario, a set of parameters is configured for the purpose of localisation, as follows:

1. **SPs**: For simulating both BT and WiFi functions for SILS purposes, the functionality of both technologies should be included in a single OPNET node. This is because in OPNET each technology has been implemented in a single node. Since OPNET standard libraries don't support BT model, SuiteTooth tool is used to simulate BT technology [130]. The tool includes PHY, MAC, Network, Transport and application layers, as it is illustrated in Figure 5-12. Specifically for SILS, only the PHY and MAC layers are used. A single SP in OPNET within supporting both BT and WiFi functionalities is shown in Figure 5-13.

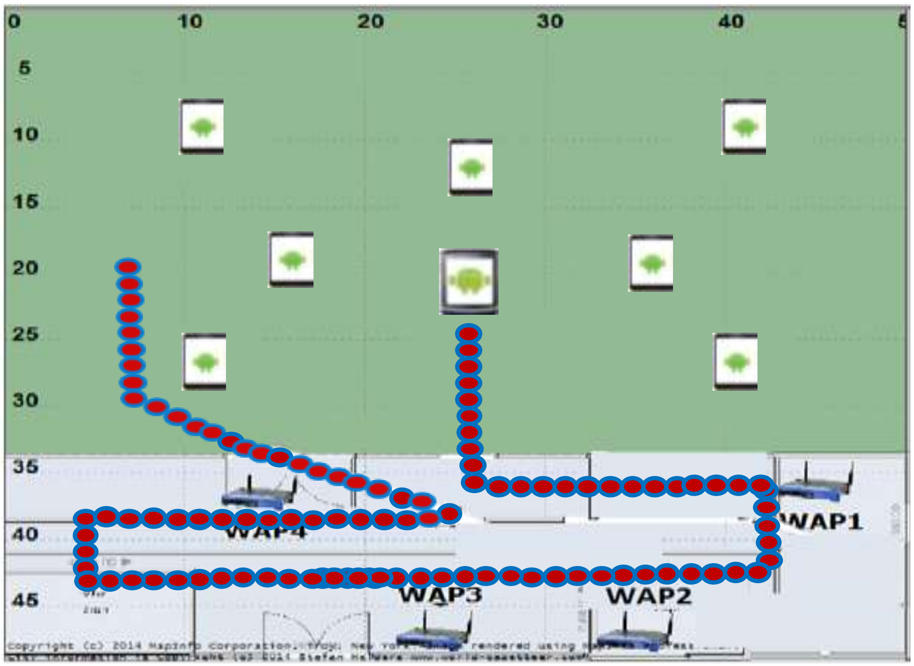


Figure 5-11: Piconet when Master moved from outdoors to indoors.

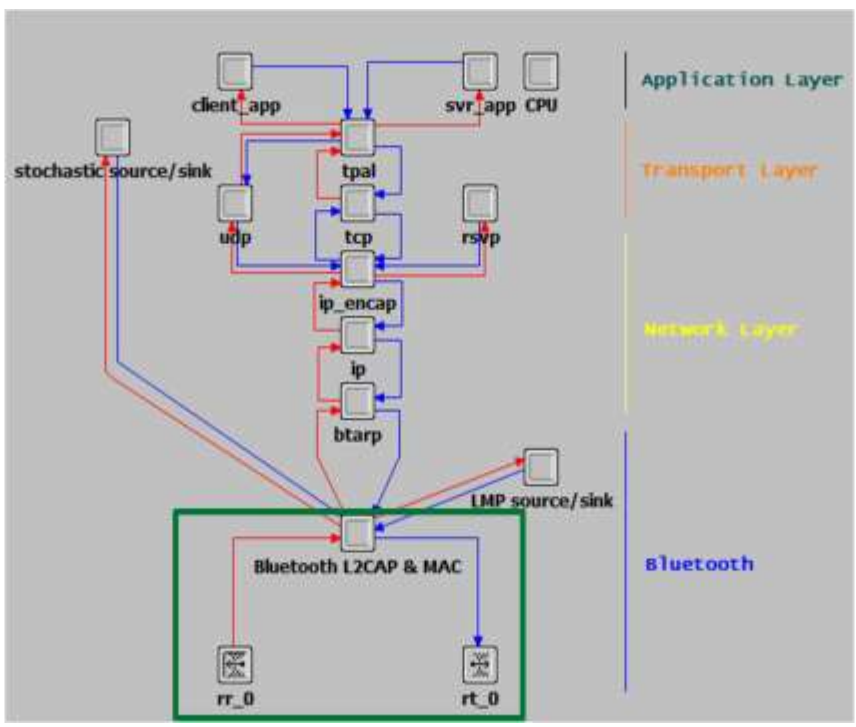


Figure 5-12: SuiteTooth OPNET node model.

The important parameters which are configured for network of SPs for SILS purpose include the followings:

a. **BT**: The MAC layer for the SP localisation scenarios is configured based on BT class 3. For example, i) the transmitted power is set to 0.1 Watt, ii) POLL-NULL frame have been utilised to exchange message between the slaves and master nodes, iii) and the interval of the exchanged frames is configured to 1000 slots, which is equal to 0.625 second [131].

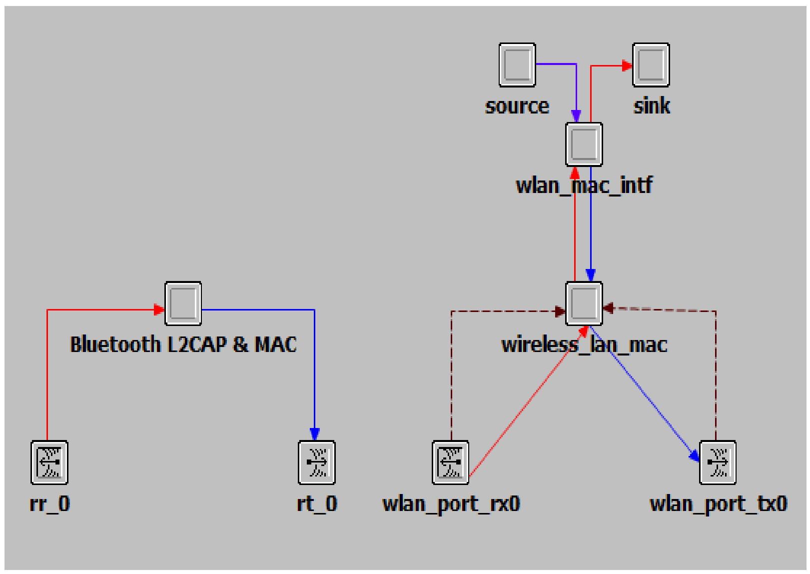


Figure 5-13: An SP example with support of both "BT & WiFi" MAC and PHY functions.

For localisation purpose, a set of new functions have been added, as demonstrated in Figure 5-14.

The functions include: Move, Locate_nd, Switch_Ro, and GNSS_Clck.

i. SP uses Locate_nd function to define the procedure of Trilateration in an iterative LLS and then to calculate its location when. In addition, this function has already implemented the PRP algorithm. The c-language code of this PRP algorithm in the OPNET modeller is listed in **Appendix B.2.**

ii. Move function is used to set a new location of the SP when the SP is moved in a selected trajectory. The selected trajectory works based on the new developed algorithm, as it is explained in **section 4.5.2**.

iii. Switch_Ro function is used to change the role of the SP from Master to Slave role or vice versa, based on the new SMSR algorithm. The c-language code of this function in the OPNET modeller is listed in Appendix B.3.

Finally, the SPs use GNSS_Clck to generate timestamp at nanosecond resolution during receiving and transmitting frames.

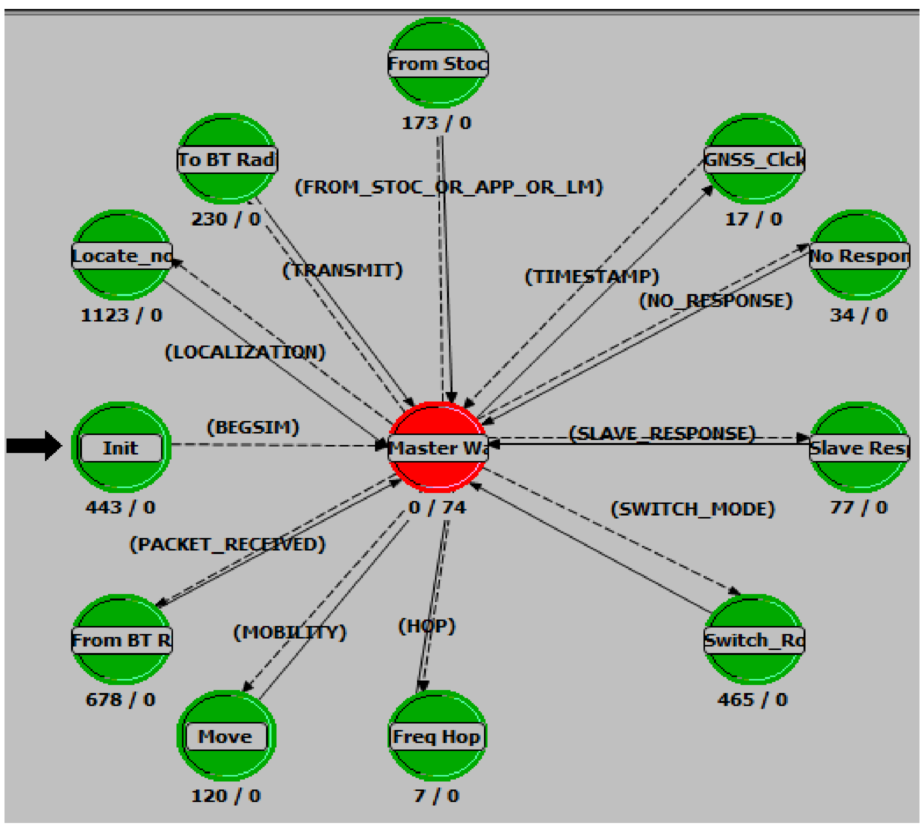


Figure 5-14: Modified SP Process model.

b. **WiFi**: Parameters which have been configured at the MAC layer are the same configurations as in chapter 3 (**see section 3.5.2**). Such configuration includes clock offset, clock drift, and generating local-time. Furthermore, the same WiFi-node model has been used for receiving beacon frame (as shown in Figure 5-13) and for generating timestamps for the received beacons (using GNSS receiver clock).

2- **WAP node model:** The WiFi MAC layer for the WAPs is configured as in chapter 4 (see section 4.5.2). Such configuration as constructing beacon frame, receiving/transmitting beacon frame and synchronising WAPs clock with GNSS time (WAPs clock offset & drift).

3- **Performance Metrics:** the performance metrics considered to evaluate the SILS include the quality of pseudorange measurements, number of connected SP in the network and obtained location accuracy. The detailed of evaluation performance is explained in next subsection (**section 5.4.3**).

5.4.3 SILS Performance Evaluation

Three simulated trial experiments are conducted to evaluate the performance of SILS. In all the following experimental figures, a trial is shown in a green line where the indoor zone is a single wall. The trial is shown by the blue line where the indoor zone is 2 walls. The deep indoors 3-walls trial is shown by the red colour. Figure 5-15 shows the Signal/Noise obtained by the SP node that is labelled “master” as it travels from outdoors to deep indoors and comes back.

Note that the indoor zones in this scenario are based on the indoor path loss model of COST-231 [132] for both WiFi and BT signals based on equation (5.1). According to this equation, regardless of number of floors, indoor zones are simulated in three different cases. There is only one thick wall (type of L_{w1}) (i.e. K_{w1}=1), one thick wall plus one thin wall (type of L_{w2}) (i.e. K_{w1}=1 K_{w2}=1), and one thick wall plus 2 thin walls K_{w1}=1 K_{w2}=2.

$$L = L_{FS} + L_c + \sum_{i=0}^{K_w} K_{wi} L_{wi} + n^{\left(\left(\frac{n+2}{n+1}\right) - b\right)} * L_f \qquad (5.1)$$

Where:

L_{FS} = Free space between transmitter and receiver ($10nLog(d)$)

L_c= Constant loss (e.g. at zero distance d_0)

K_{wi}= Number of penetrated wall of type I

n = Number of penetrated floor

L_{wi}= Loss of wall type I

L_f = Loss between adjacent floors

b = Empirical parameters

Note, in the simulation experiments, the above parameters are initialised as follows: the L_c normally to 40 dB, L_{w1} =6.9 dB, L_{w2}=3.4 dB and L_{FS} =0.0 dB

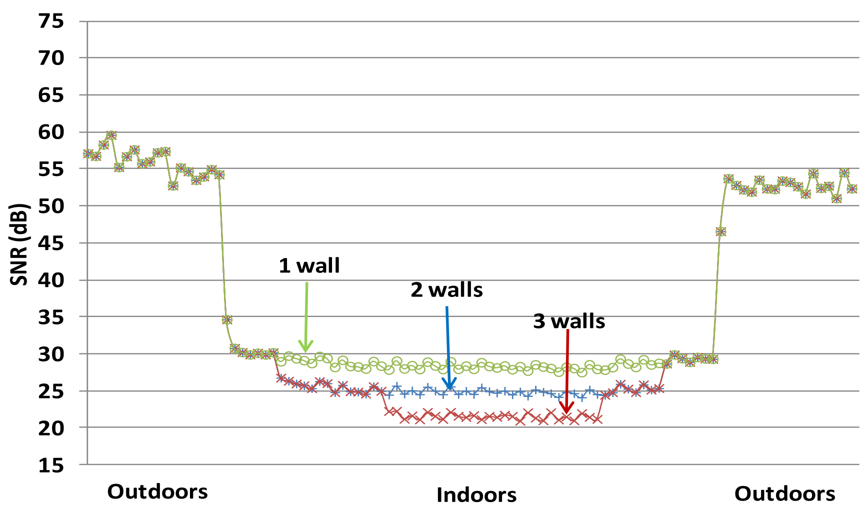


Figure 5-15: SNR measurements from outdoor to indoor.

In addition, Figure 5-16 shows the number of SPs connected in the BT network as the "master" node signals passed extra walls. Note: the location information of these connected SPs is used as reference position (reference nodes) to locate “master” node when Trilateration process is applied.

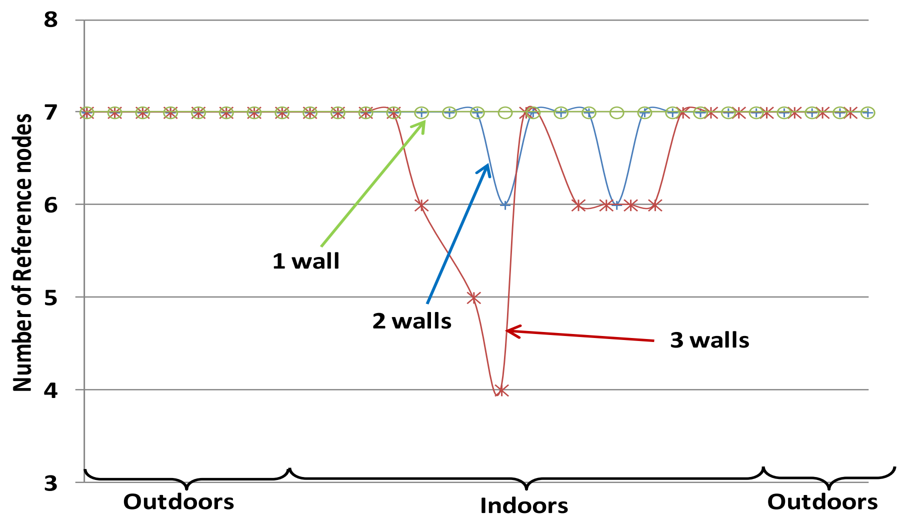


Figure 5-16: Number of SPs connected in the BT network.

The following experiments are chosen to demonstrate the achievements of the various algorithms implemented in the SILS:

1. Figure 5-17 shows the obtained location error without adopting the SMSR algorithm. That is, by only using basic hop synchronisation to estimate pseudoranges between SPs. The achieved location accuracy of the “master” node through the scenario path can be as high as 3.5 meters when the trial is for 3-walls deep indoors, and 2.5 meters for 2-walls deep trial.

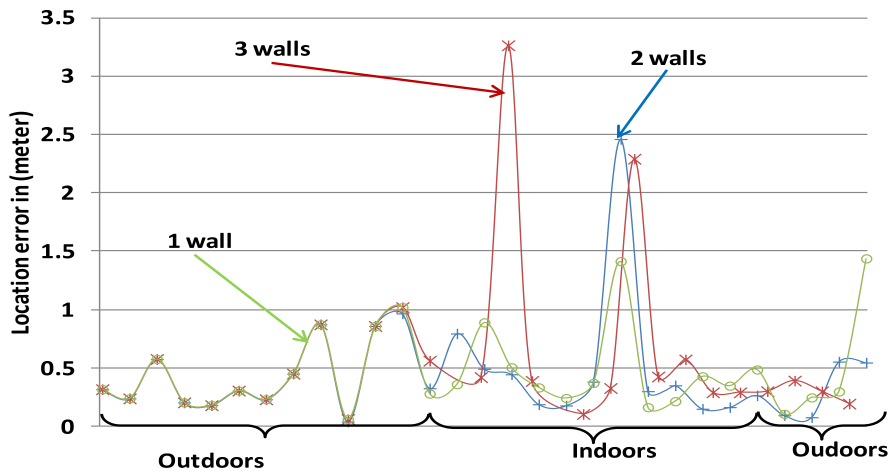


Figure 5-17: Location errors from outdoors to indoors without SMSR.

2. Figure 5-18 shows the location error when the SMSR and PRP algorithms are applied. It can be seen that for the 2-walls trail, the location error of the "master" node is being reduced to over 1meter.

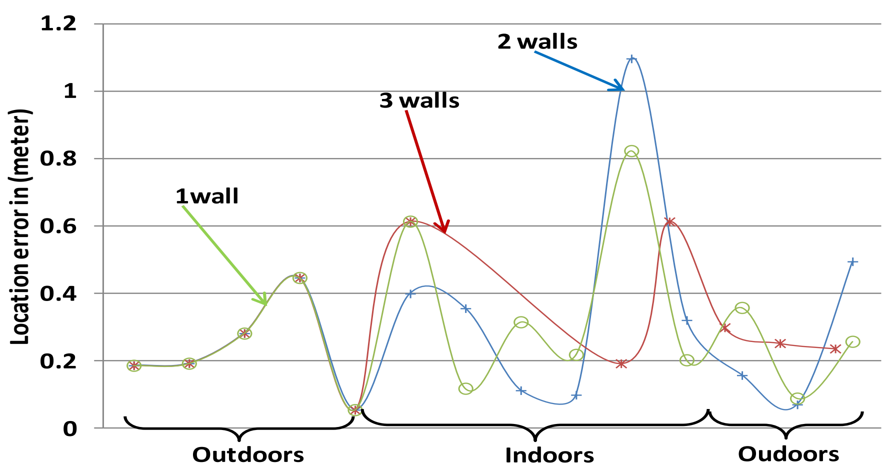


Figure 5-18: Location error during development of the SILS without WAPs.

Note: also Table 5-1 displays the pseudoranges enhancement via the SMSR algorithm in comparison to the basic hop synchronisation measurements. However, the SILS has failed to locate the "master" when it passed through 3-walls deep indoors. This is due to the restriction placed by the PRP algorithm that restricts fixing when 4 SPs or less are connected to the BT network. This issue has been solved by using available WAPs signals as shown in the next experiment.

Table 5-1: Pseudoranges measurement comparison.

Trial number	Pseudorange measurements				Error in percentage (%)	
	Actual distance		Hop-sync (only)	Hop-sync plus SMSR	Hop-sync (only)	Hop-sync plus SMSR
1	Short	4.50	**3.35**	**4.72**	**25.55556**	**4.888889**
2		7.28	**8.14**	**7.21**	**11.81319**	**0.961538**
3		21.92	**23.41**	**22.29**	**6.797445**	**1.687956**
4	Mid	58.83	**62.94**	**55.91**	**6.986232**	**4.963454**
5		50.75	**48.34**	**49.35**	**4.748768**	**2.758621**
6		52.39	**49.99**	**51.67**	**4.581027**	**1.374308**
7		60.09	**63.3**	**59.95**	**5.341987**	**0.232984**
8	Long	73.09	**75.07**	**74.32**	**2.708989**	**1.682857**
9		97.18	**99.22**	**96.5**	**2.099197**	**0.699732**

3. Figure 5-19 shows the SPm location error when the SILS works in the full-functionality mode with using beacon signals WAPs. It also demonstrates how the error of 3-walls deep indoors trial has now been improved by including the WAPs as reference positions. That is, the failure issue of the SILS is solved. The SILS can provide locating the "master" even when the BT network is 4 SPs or less. Furthermore, the error for this trial has been reduced from 3.5 meters to just over 2 meters.

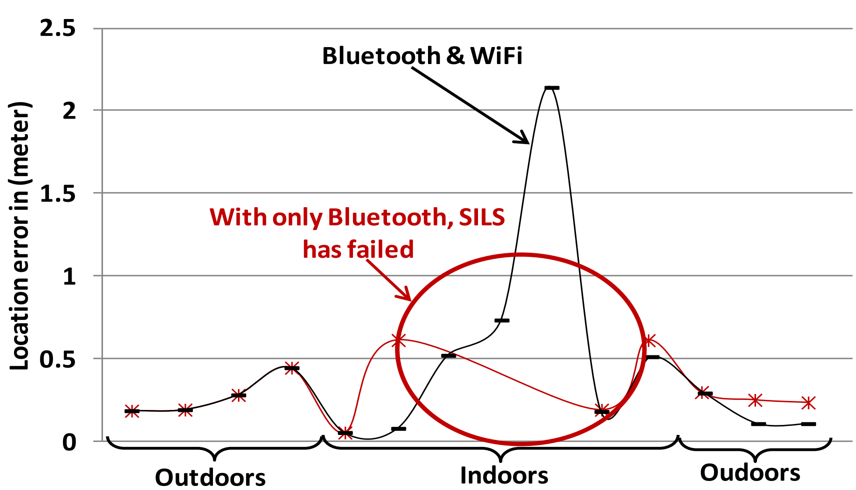


Figure 5-19: Location error for full SILS functionality.

5.4.4 SILS Performance Evaluation in Multi-Piconets

All the aformentioned experiments, in previous subsections, are tested for the single Piconet scenario. In order to evaluate the performance of the SILS with multiple Piconets, various scenarios and experiments are conducted. For example, 20 groups of 8 SPs scenario starts to run the SILS in the vicinity (outside of buildings). One of the SPs, in each group of SPs (the Piconets), moves indoors through light indoors area (signals are crossing 1 wall from the outside) and deep indoors (signals are crossing 3 walls deep inside the building). Due to the big size of the figures and table of the results, all the figures and evaluation results are moved to the **Appendix B.5.** In addition, the movement of the SPs and the parameters for both WiFi & BT technologies are configured as the same of single Piconet. The obtained results show that the SILS can be run on multiple Piconets in the vicinity and the SILS provides 2-3 meters accuracy as well.

Further, Table B.1 shows some statistics on the obtained location accuracy for all the indoors-SPs of the 20 Piconets. As it is observed that:

1. The SILS can provide locating the SPs even when multi-Piconets (in the vicinity) are run concurrently.

2. The maximum standard deviation of location errors for all the 20 Piconets is 2.9 meters when the SILS is run in development mode (only Hop-Synchronisation is applied to estimate pseudoranges). While the maximum standard deviation of the obtained location errors via SILS in full-functionality (SMSR+PRP are applied) is 0.57 meter. That is, again both SMSR and PRP algorithms enhance SPs location accuracy.

3. The maximum mean of the SPs’ location errors is 1.1 meter when the SILS is run in development mode, while the maximum mean value is 0.6 meter when the SILS is run full-functionality.

5.5 UNILS Using Onboard BT, Inertial Sensors and GNSS

This section addresses the ninth research question (**as it is labelled in section 1.4**): how can the position of indoors-SPs be obtained without using any localisation infrastructure, unconstraint and cooperatively?

In section 5.4, it is proved that whereby the participating SPs in the outdoors and indoors vicinity, the SILS forms a BT network. Together with WAPs, the BT network can provide a localisation accuracy to over 2 meters at 3 walls deep indoors. However, sometimes WAPs' radio-signals do not cover the vicinity. That is, the WAPs signals to/from SPs are blocked due to noisy and complex structure of indoors with many walls, doors, ceilings and floors. To compensate the blocking issue, a new version of the SILS is developed, which is called UNILS. The main aim of the UNILS is to introduce DR to the SILS so as to locate indoors-SPs and improve localisation accuracy. This solution is used as an unconstraint solution when communication with WAPs or BT-anchors is considered unreliable or unavailable. This means that, in deep indoors, the UNILS can utilise only available devices/sensors on SPs. However, DR measurements on SPs suffer from accumulated error due to existing sensor reading's drift.

Therefore, the implementation prototype of UNILS has two main functions. The **first** function is to use relative-pseudoranging between the cooperative network-SPs via using TOA technique, especially when the majority of the SPs are outdoors. **Secondly**, it fuses this pseudoranging with the uncertainty calculations from the inertial sensors measurements (using DR technique). This measurement's fusion exploits the advantages of each of both techniques while compensating their limitations.

In addition, the UNILS can be implemented as an application. When the UNILS is invoked, it renders the SP as cooperative for being located and helping in locating other SPs. the UNILS can do so anywhere at any time. In the next subsection the details of the UNILS implementation is described.

5.5.1 UNILS Functionalities

The UNILS works in the following steps (as shown in Figure 5-20):

1. The UNILS, like the SILS, first constructs a BT network on-the-go with any SPs in the vicinity. Moreover, it is assumed that outdoors SPs can accurately obtain their geographical location.

2. The UNILS estimates relative pseudoranges of all participating SPs based on the hop-synchronisation (see section 5.4.1.1). This estimation is conducted through the exchanging specific BT-frame format such as POLL-NULL frames. Furthermore, the SMSR is utilised to minimise the pseudoranges' error (see section 5.4.1.2).

3. All connected SPs in the network read their onboard inertial sensors measurements including acceleration and angular velocity. These readings together with GNSS fixed locations are shared between indoors-SPs and outdoors-SPs.

4. To calibrate the inertial sensors of all SPs, indoors-SPs use both live GNSS information available from networked outdoors-SPs and ZUPT algorithm. The calibration process is to estimate gyroscope bias and drift error for indoors-SPs.

5. Then, the UNILS calculated the indoors-SPs position by fusing the measured pseudoranges with DR measurements (distance-displacement and heading) by using Kalman filter. To do this, the UNILS uses a step-counting method to estimate distance-displacement and uses quaternion-based method to estimate indoors-SPs heading [15].

   The Kalman filter include two phases, at the first phase, predicts a state-vector of XY-coordinates and heading based on DR measurements. In the second phase, the filter updates the predicted state-vector by using three measurements. 1) The difference between estimated relative-pseudoranges and calculated Euclidean distance of SP-to-SP positions. 2) Estimated heading bias & drift. 3) Change of the heading between any two state-vectors.

In the following subsections, all aforementioned proposed algorithms are described including pseudorange measurements algorithm, DR algorithm (using onboard inertial sensors) and fusing-measurement algorithm.

5.5.1.1 Pseudorange Estimation

Like the SILS, the pseudorange measurement (via BT signals) based on the Hop-synchronisation counting is proposed to calculate the relative distance between any two SPs. That is, to measure the pseudoranges, the TOA technique is used. The UNILS also uses the master/slave role switching algorithm to enhance the pseudorange measurements.

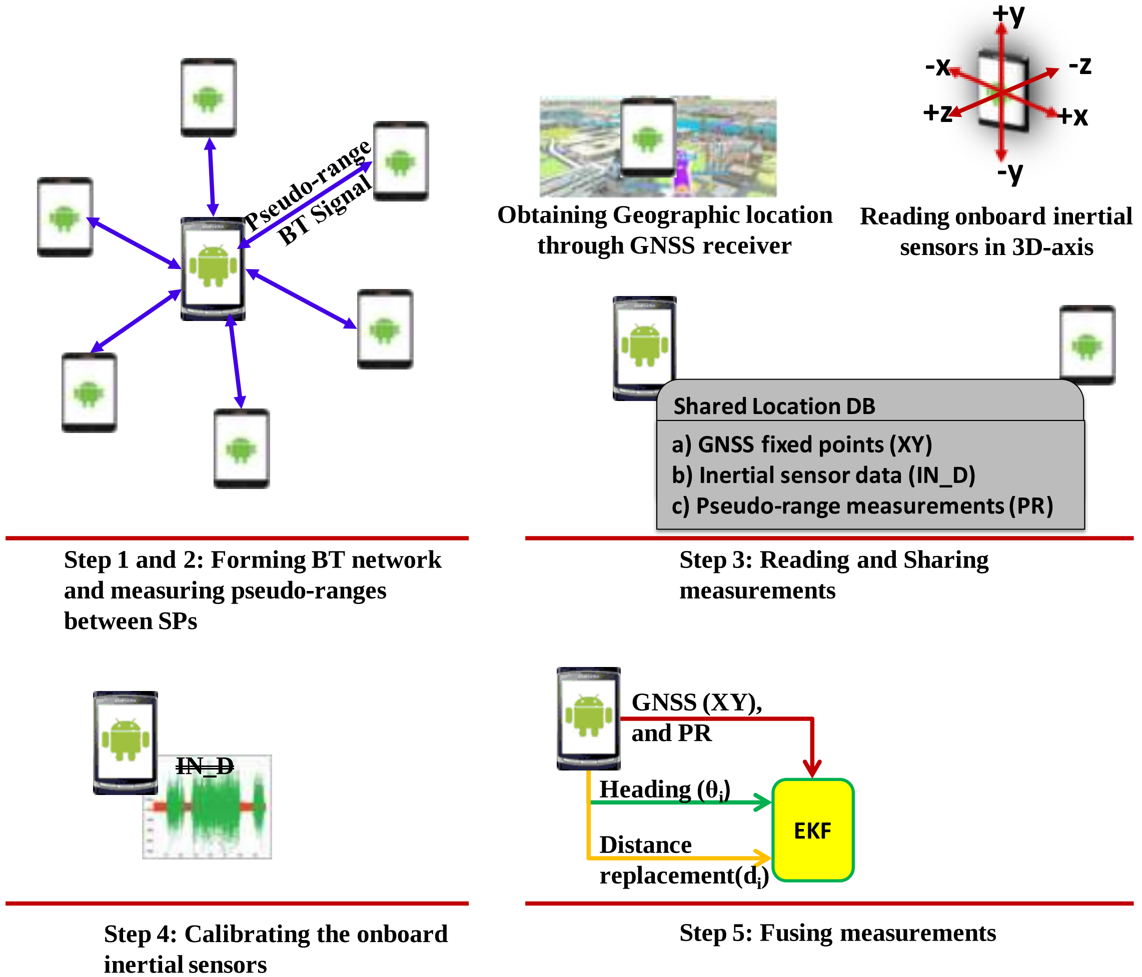


Figure 5-20: Steps of UNILS implementation.

In the UNILS, these measurements are used together with the calculated Eqludiance distance between the two SPs to update the DR measurements of indoors-SPs. The Eqludiance distance is the difference between XY-coordinates of the outdoors SPs (based on GNSS fixed points) and calculated XY-coordinates for indoors-SPs (using DR technique).

5.5.1.2 DR Using Onboard Inertial Sensors

In DR technique (as described in section 2.2.6), the process of estimating a current position is based upon a previously determined position, estimated step length (i.e.

distance-displacement) and an estimated heading. For example, an SP relative-position can be computed by adding the step length and the estimated heading to the previous position as in the following equations:

$$x_k = x_{k-1} + d_k * \ cos\theta_k$$

$$y_k = y_{k-1} + d_k * \ sin\theta_k \qquad (5.2)$$

Where $\boldsymbol{x_k}$ & $\boldsymbol{y_k}$ are XY coordinates position, $\boldsymbol{x_{k-1}}$ & $\boldsymbol{y_{k-1}}$ are previous determined XY coordinates position, $\boldsymbol{d_k}$ is an estimated distance-displacementby using accelerometer sensor and $\boldsymbol{\theta_k}$ is an estimated heading by using onboard SP gyroscope sensor at step $\boldsymbol{k}$.

The readings of accelerometer sensor, such as acceleration magnitude, are usually used to determine the step length or the distance-displacement ($\boldsymbol{d_k}$). Further, to calculate step length or distance-displacement, the following steps should be performed [133]:

1. The total acceleration magnitude from the 3-axis accelerometer must be filtered. When the pedestrian walks, the measured acceleration magnitude is a combination of downward gravity and horizontal acceleration. Therefore, a high-pass filter is used to eliminate the influence of gravity followed by a low-pass filter to eliminate the noise influence. Both filters are implemented with equations (5.3a and 5.3b) respectively.

$$a_{gravity} = a_{mag}(t) * (1 - 0.95) + a_{gravity} * 0.95$$

$$a_{hpass}(t) = a_{mag}(t) - a_{gravity} \qquad (5.3a)$$

Where, the gravity is a low-frequency signal component that causing offset shift up the y-axis, about 9.8 m/s^2.

$$a(t) = \frac{1}{N} * \sum_{j=-(N-1)/2}^{(N-1)/2} a_{hpass}(t + j) \qquad (5.3b)$$

Where, N is a window size or moving window, i.e. the number of the samples used in the moved average. Note: for the conducted experiments, N is initialised to 5.

Thus the obtained result, after applying these filters, is a signal free from gravity with low noise.

2. Next, to detect the step of the SP-pedestrian, the UNILS uses peak-detection algorithm (as expressed in equation 5.3c).

$$peak = \begin{cases} min\ (a(t{:}\,t + s_{sample}) > th & min - peak \\ max\ (a(t{:}\,t + s_{sample}) > th & max - peak \end{cases} \qquad (5.3c)$$

Where $a(t)$ is the acceleration, s_{sample} is the number of samples between any two steps, min-peak & max-peak are the lower and upper peaks respectively at every step, and the th is the empirical threshold to identify the upper and lower peaks' boundary.

3. Finally, the length of the detected step can be calculated as follows:

$$d_k = K * \frac{\frac{\sum_{i=t}^{t+N} |a_i|}{N} - a_{min}}{a_{max} - a_{min}} \qquad (5.3d)$$

Where K is a constant parameter, for example K would be equal to 0.6 meter, a_{max} and a_{min} are respectively the maximum and minimum value of acceleration of N acceleration measurements during one step (i.e. between any two detected peaks), and a_i is the value of the acceleration at every measurements. The Java code in Android-based SPs to detect the step and then to calculate step length SPs-heading is listed in **Appendix C.2.**

The angular rates of x, y, and z coordinates are measured by three-axis gyroscope readings. The attitude of the SP is obtained from the integration of the quaternion-based rigid body kinematic. Note: The Java code in Android-based SPs for estimating SPs-heading is listed in **Appendix C.3**.The quaternion estimation is a four-dimensional vector which is calculated by vector-values of gyroscope readings and a rotation angle [134]. The four dimensional vector is given as:

$$\boldsymbol{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos\,(\frac{\theta}{2}) \\ sin\left(\frac{\theta}{2}\right) * (^{w_x}/_{\mu}) \\ sin\left(\frac{\theta}{2}\right) * (^{w_y}/_{\mu}) \\ sin\left(\frac{\theta}{2}\right) * (^{w_z}/_{\mu}) \end{bmatrix} \tag{5.4}$$

Where $\mu = \sqrt{w_x^2 + w_y^2 + w_z^2}$ and $\theta = T_s\left(\sqrt{w_x^2 + w_y^2 + w_z^2}\right)$

Where w_x, w_y, and w_z are the angular rate of x, y, and z coordinates respectively. T_s is the interval time between any two consequent gyroscope readings. According to the relationship between the attitude rotation matrix and the quaternion, the rotation matrix can be calculated as:

$$R_b^n = \begin{bmatrix} (1 - q_2^2 - q_3^2) & (q_1q_2 - q_3q_0) & (q_1q_3 + q_2q_0) \\ (q_1q_2 + q_3q_0) & (1 - q_1^2 - q_3^2) & (q_2q_3 - q_1q_0) \\ (q_1q_3 - q_2q_0) & (q_2q_3 + q_1q_0) & (1 - q_1^2 - q_2^2) \end{bmatrix} * gyro\_matrix \tag{5.5}$$

$$Where \qquad gyro\_matrix = X_m * Y_m * Z_m$$

Where $\boldsymbol{X_m, Y_m}$ $\boldsymbol{and\ Z_m}$ are the rotation matrices about x, y and z directions, which are described in [134]. The $\boldsymbol{R_b^n}$ matrix is also used to transform the gyroscope readings from the body frame (b) to a navigation frame (n). The azimuth (heading), roll and pitch values can be then obtained by applying the *atan2* and *asin* math functions on the values of the $\boldsymbol{R_b^n}$ propagated inside the sensors navigation equations:

$$Orientations = \begin{cases} Azimuth: & atan2(R_{1,2}, R_{1,1}) \\ Pitch: & asin(-R_{3,2}) \\ Roll: & atan2(R_{3,1}, R_{3,3}) \end{cases} \quad (5.6)$$

However, due to the integral calculation of the gyroscope readings, the drift of gyroscope is rapidly accumulated over time. That is, this calculation causes large positioning error. To mitigate this issue, there is a necessity to apply filters or drift compensating algorithms on the gyroscope measurements. Such as: ZUPT algorithm [135], estimating drift and bias error based on some reference measurements like the GNSS heading and velocity [136], or fusing with wireless signals' measurements including WiFi, UWB and BT [137].

The ZUPT is a known algorithm to detect the standing phase and reduce the cumulative error. The reduction assumes that the angular velocity equals zero when the bottom of the sole has complete contact with the ground. In addition, the four known ZUPT detector algorithms are: ARE detector, AMV detector, GLRT and AM Detector. For this research, ARE is implemented as it is proved that it outperforms the other detector algorithms [138]. This is because; the gyroscope readings hold the most reliable information for zero-velocity detection under the experimental conditions. The ARE can be implemented by using the equation (5.7).

$$T(t) = \begin{cases} 0 & < \gamma \\ 1 & > \gamma \end{cases} \quad (5.7)$$

$$Where \quad T(t) = \frac{1}{N} \sum_{j=-(N-1)/2}^{(N-1)/2} \left\| \mu_{(t+j)} \right\|^2$$

Where $\boldsymbol{\mu}$ (as described in equation 5.4) is the magnitude of the angular rate, $\boldsymbol{\gamma}$ is an empirical threshold, N is the moving window, and $\boldsymbol{T(t)}$ is the angular rate energy detector at time $\boldsymbol{t}$ that decides to zero when the velocity is below the threshold otherwise is equal to 1.

In the UNILS, a hybrid approach is developed to mitigate gyroscope-measurements' drift. First, the UNILS applies the ZUPT on the obtained angular velocity to take out the noisy measurements when the SPs are in standing phase. Second, to reduce the contributed noise during walking, the UNILS fuses the BT-based pseudorange measurements with the result of the ZUPT algorithm. The details of fusing measurements algorithm is illustrated in the next subsection.

5.5.1.3 Pseudorange & DR Integration Using EKF

This subsection addresses the tenth research question: Can fused onboard SPs wireless transceivers and sensor measurements prevent the accumulated error of the defined indoors-SPs position? In the UNILS, the EKF is used to fuse SPs-cooperated localisation measurements. These measurements include pseudoranges via the BT transceivers, the GNSS fixed positions from GNSS receivers and DR measurements from inertial sensors.

The EKF is the nonlinear version of the Kalman filter which works by linearising the state space around the mean and covariance of the current state estimation. The discrete dynamic equations of the EKF can be expressed by two phases: prediction phase and update (or correction) measurements phase [139], as illustrated in Figure 5-21.

In the Figure 5-21, $\hat{x}_k^-$ is the predicted state-vector which includes XY-coordinates and the heading $[x, y, \theta]$ using the DR technique, B_k is the state space model matrix which can be expressed in equation (5.8):

$$B_k = \begin{bmatrix} d_k * \cos(\theta_k) \\ d_k * Sin(\theta_k) \\ \Delta\theta_k \end{bmatrix} \tag{5.8}$$

Moreover, P_k^- is the error covariance matrix which includes: a) predicted error noise (Q) for both distance- displacement and drift error for the estimated heading b) the A is the state transition matrix, as expressed in equation 5.9.

$$A = \begin{bmatrix} 1 & 0 & -d_k * \sin(\theta_k) \\ 0 & 1 & d_k * \cos(\theta_k) \\ 0 & 0 & 1 \end{bmatrix} \qquad (5.9)$$

Note: $\boldsymbol{P_k}$ is updated at every iteration using three parameters. The parameters are predicted error covariance matrix (P_k^-), obtained Kalman-Gain (K_k) and the updated measurements ($\boldsymbol{H}$).

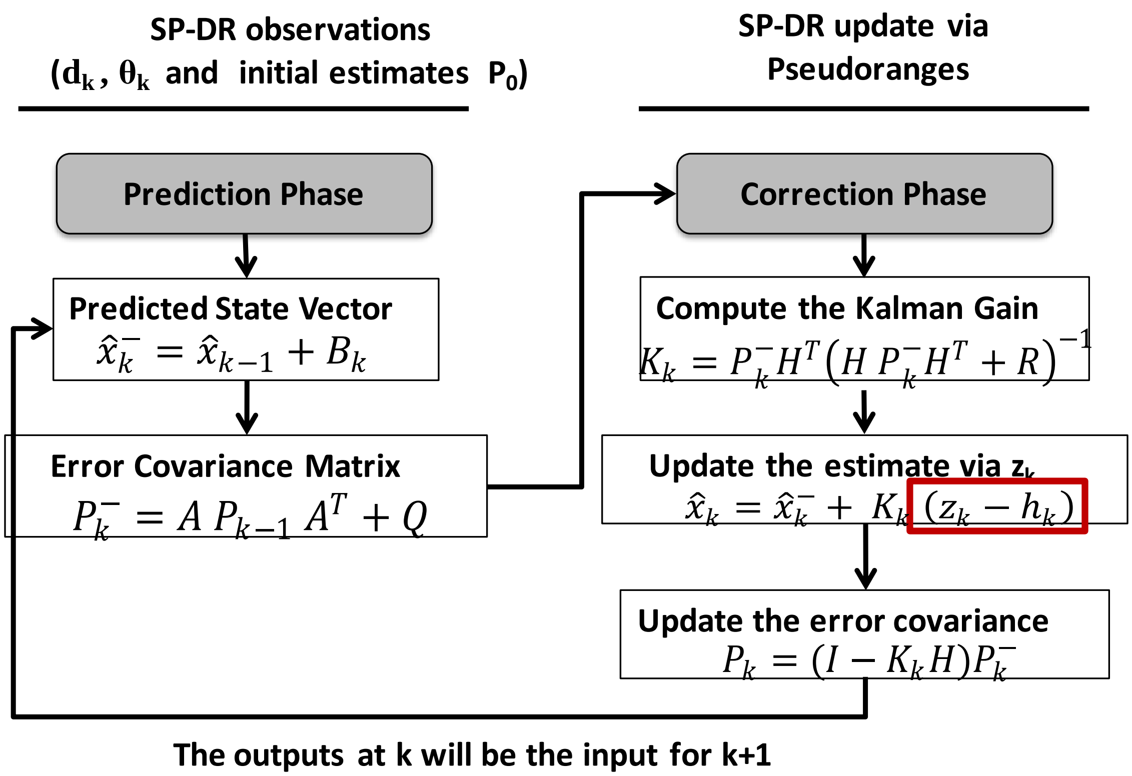


Figure 5-21: Iterative based Extended Kalman Filter equations to fuse smartphones measurements [139].

Consider the measurement-update model which describes the pseudorange measurements taken between any two SPs. The update model works based on estimated pseudoranges and location information of both indoors-SPs and outdoors-SPs (Euclidian-distance h_k), as shown in equation (5.10).

$$\hat{x}_k = \hat{x}_k^- + K_k\,(z_k - h_k) \qquad (5.10)$$

The pseudoranges can be estimated by calculating TOF between any two SPs and then multiplying it by speed of light (c), as expressed in equation (5.11):

$$z_k = TOF * c \qquad (5.11)$$

The Eqludiance distance between the predicted XY coordinates of indoors-SPs and the obtained XY coordinates of ourdoors SPs can be expressed as in equation (5.12):

$$h_k = \sqrt{(x_k - x_g)^2 + (y_k - y_g)^2} \qquad (5.12)$$

Where $\boldsymbol{x_k}$ and $\mathbf{y_k}$ are predicted XY coordinate of indoors-SPs based on DR measurements, and $\mathbf{x_g}$ and $\mathbf{y_g}$ are obtained XY coordinate of outdoors-SPs via the GNSS. In addition, to apply the extended Kalman filtering algorithm, $\boldsymbol{h_k}$ equation should be linearised, as it is expressed in equation (5.13):

$$H = \begin{bmatrix} \frac{\mathrm{x_k} - \mathrm{x_g}}{\mathrm{sqrt}\left((\mathrm{x_k} - \mathrm{x_g})^2 + (\mathrm{y_k} - \mathrm{y_g})^2\right)} \\ \frac{\mathrm{y_k} - \mathrm{y_g}}{\mathrm{sqrt}\left((\mathrm{x_k} - \mathrm{x_g})^2 + (\mathrm{y_k} - \mathrm{y_g})^2\right)} \\ 0 \end{bmatrix} \qquad (5.13)$$

Finally, according to this procedure, the predicted measurements can be iteratively updated when any new measurement-reading is available. A full block diagram for fusing the outdoors-plus-indoors SPs' measurements using EKF is illustrated in Figure 5-22.

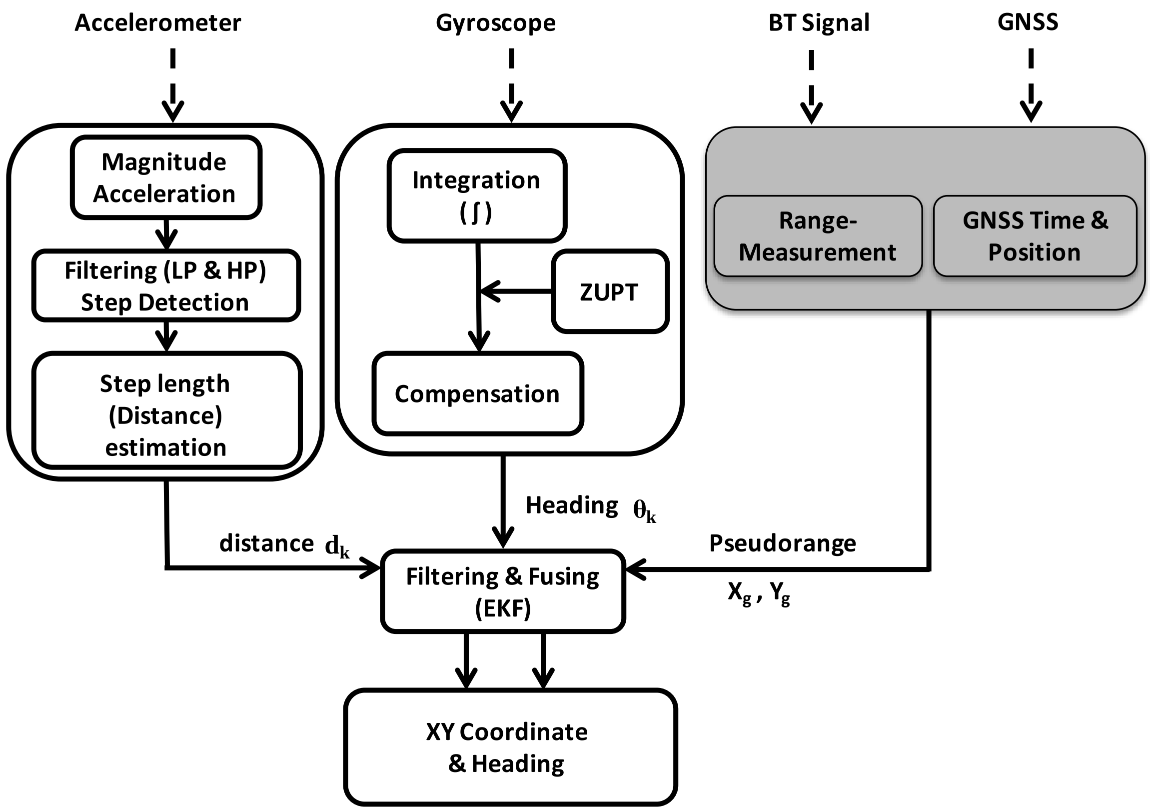


Figure 5-22: Block diagram of UNILS functions.

5.5.2 Simulation Results and Discussion

The OPNET is used to simulate the UNILS implementation. To prove the validity of the UNILS, a single scenario is chosen where a group of 8 SPs form a BT-network inside/around a building (as shown in Figure 5-23). Three of these SPs (SP_IND1_Master, SP_IND2, and SP_IND3) move through deep indoors, for 5 minutes, and the other five SPs (SP_OUT1, SP_OUT2, SP_OUT3, SP_OUT4, and SP_OUT5) move around the building. In the simulated scenario, the deep indoor is represented when SP-BT signals are crossing 3 walls inside the building. The movement of SP_IND1_Master is illustrated in Figure 5-23 by trajectory line. In addition, the configuration parameters for SPs in the BT-network for the purpose of the UNILS are the same as the SILS. Except for the UNILS purpose, the sensor-model simulation for accelerometer and gyroscope is added or implemented, as shown in Figure 5-24.

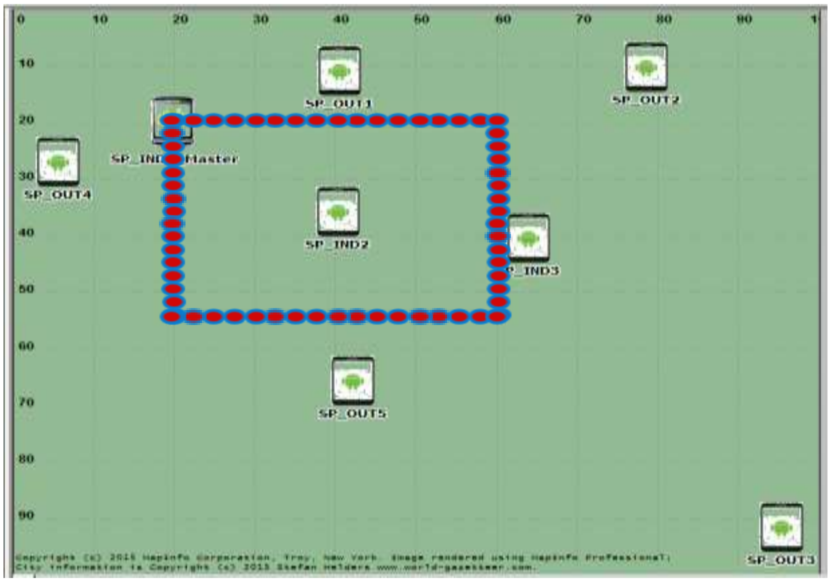


Figure 5-23: The trajectory of SP-IND1_Master within a Piconet, when indoors.

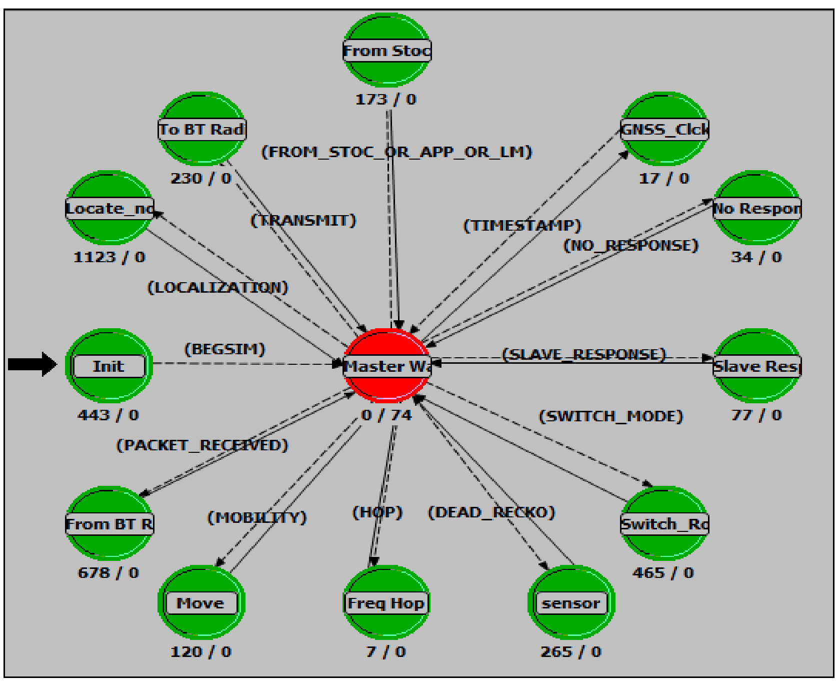


Figure 5-24: Modified SP Process model including sensor model.

This simulated sensor-model is interrupted at every step when the SPs are moved from one location to the next one. This movement takes 0.6 meter/second speed (assumed as a normal walking speed of SPs-users). The c-language code of this model in the OPNET modeller is listed in **Appendix B.4.** This model also implements the fusion algorithm using the Kalman filter. Furthermore, the simulated sensor-model, for calculating XY coordinates and headings, is based on several empirical/trials measurements, for example:

1. For accelerometer sensor, the acceleration includes both bias & noise errors and real acceleration due to the SP-moving. Figure 5-25 shows an example of the magnitude of 3-axis accelerations for both simulated (in red colour) and trial measurements (in blue colour). As it can be seen, the provision of simulated measurements is near to the real accelerometers-sensor readings.

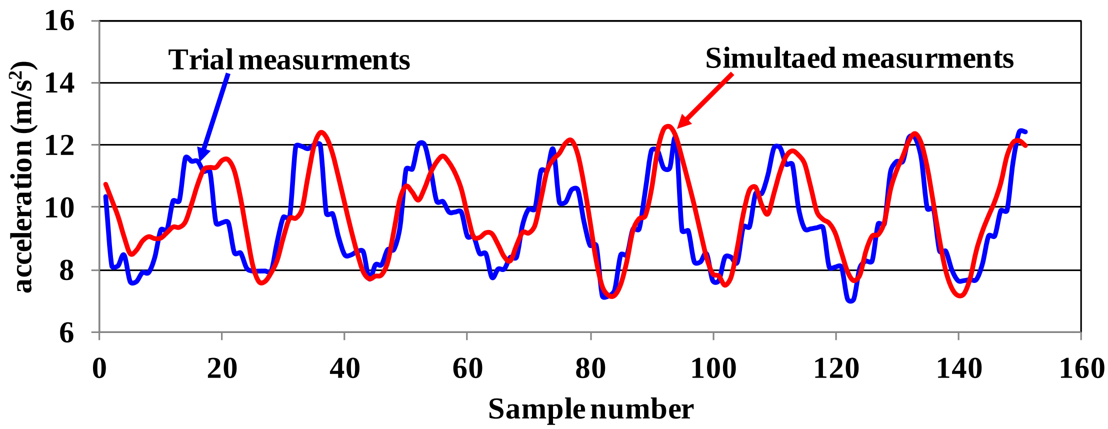


Figure 5-25: Example of magnitude acceleration of SP_IND1_Master.

The following procedure should be carried out to estimate the distance replacement of an SP from one location to the next one:

a. The magnitude of acceleration should be filtered by high-pass and low-pass filters. These filtrations are to take out the gravity and noise errors, respectively, based on equations (5.3a) and (5.3b).

b. To detect the steps and then to estimate steps' length, equations 5.3c and5.3d should be sequentially performed. Figure 5-26 shows an example of 14 real steps during walking. As it can be seen, there is a missed step due to instability of the number of samples per each step. As a fact, this issue happens when user walking-steps have not the same size of the samples of the sensor readings. In addition, the peaks of vertical acceleration correspond to the step occurrences because the vertical acceleration is generated by vertical impact when the foot hits the ground.

Figure 5-26: Example of detected steps during standing/walking phases of SP_IND1_Master.

From Figure 5-26, the peaks of the acceleration's amplitude correspond to thirteen walked steps. Therefore, if one of such peaks is detected, we can conclude that one step has been taken. This relationship is defined in the first rule, which is "Each step includes a peak (local maximum) and the next valley (local minimum) of the acceleration amplitude. In other words, if one peak and one valley are detected, one step should be counted.

Several trials and simulation experiments are conducted in different test-beds to verify step detecting/counting and step's length estimation algorithms. Table 5-2 displays experimental results for five test-beds including: a room within 5X5m, five rounds 'R5' of the room (5X5m), room within 6.5X13m, three rounds 'R3' of the room (6.5X13 m) and a corridor within 38X48m. It can be noticed, that the accuracy of these algorithms in worst case is within 2% distance error, for 172 meter travelled distance.

Table 5-2: Step counting and Step length estimation in different trials.

Test-beds	True Steps	Detected Step		True Step Size	Estimated step size	
		Simulated	Trial		Simulated	Trial
5X5m	40	39	40	0.5 m	0.51 m	0.503 m
R5 (5X5m)	200	200	198		0.5015 m	0.507 m
6.5X13m	78	78	79		0.5145 m	0.530 m
R3 (6.5X13m)	234	231	232		0.490 m	0.526 m
38X48m	344	347	341		0.515	0.515

2. For gyroscope sensor, the angular velocity includes both bias & drift errors and real angular rate during the SP-turning & the SP-walking. Figure 5-27 shows an example of the angular velocity of z-axis gyroscope for both simulated (in red colour) and trial measurements (in blue colour). It is observed that simulated angular measurements are near to the real sensor reading.

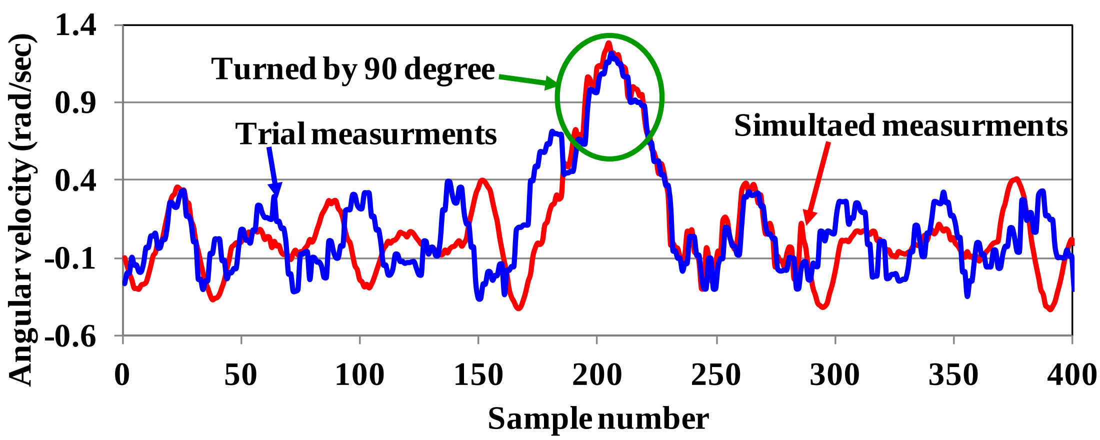


Figure 5-27: Example of angular velocity measurements of SP_IND1_Master.

To estimate the heading of an SP from one location to the next one, the quaternion-based equations (from 5.4 to 5.5) should be performed on the angular velocity. According to the simulated scenario, Figure 5-28 shows the estimated heading which is obtained by the SP_IND1_Master node as it travels during 5 minutes. It can be observed that there is a huge drift accumulation of the estimated heading, as it is available with real gyroscope measurements.

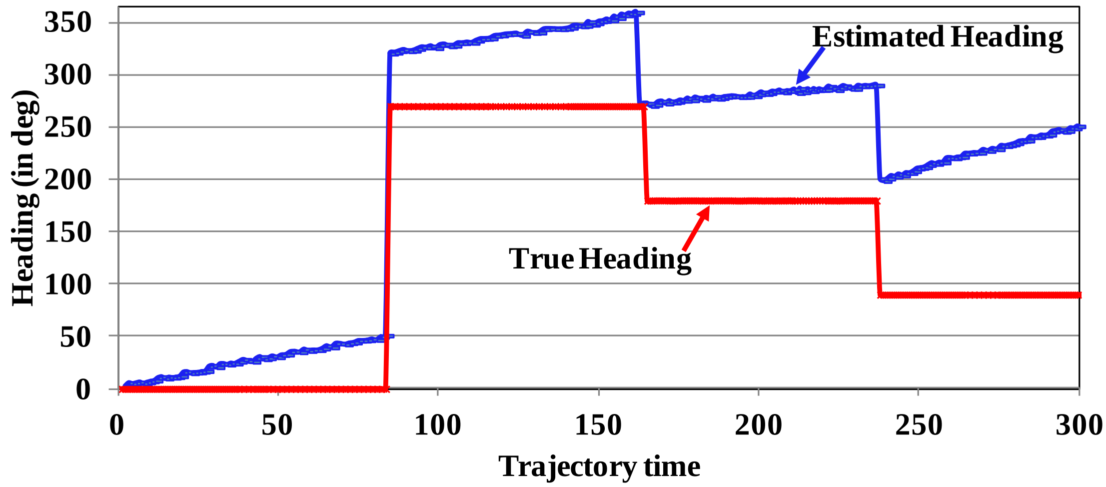


Figure 5-28: Estimated Heading of SP_IND1_Master node.

Since the UNILS is for indoors environments, BT signals between SPs during receiving and transmitting should be simulated for indoor zones. The indoor zone is simulated by using the indoor path loss model COST-231 (see equation 5.1). To show the validity of the simulated indoor zone, SNR measurements are good candidate calculations. Figure 5-29 displays the SNR (in red line) obtained by the SP_IND1_Master node as it travels in deep indoors. In addition, the figure shows the number of SPs connected (in blue line) in the BT network as the "SP_IND1_Master" nodes cross extra walls.

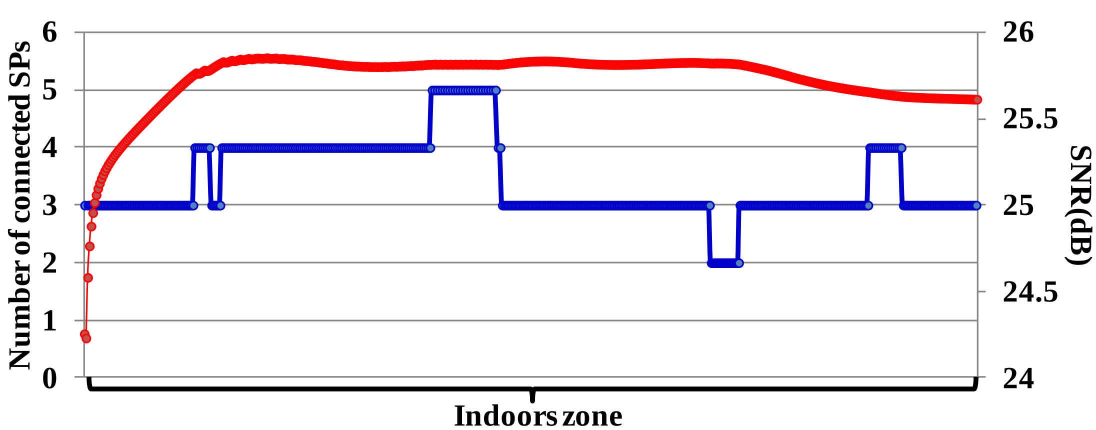


Figure 5-29: SNR measurements & Number of reference nodes in the network.

To show the accuracy of the UNILS in comparison with the SILS and standalone-DR technique, an experiment is tested as shown in Figure 5-30. In this experiment, four trials are conducted. The **first** trial is to test the SILS under developed functionality (i.e. only Hop-synchronisation counting and master/slave role switching algorithms are included) shown in red line. The **second** trial is to test the SILS with full-functionality mode shown in purple. The **third** trial is to test the UNILS shown by the black line. The **fourth** trial is to test the standalone-DR shown by the light-green line. The results show that:

1. The SILS cannot continuously define SP_IND1_Master location. This is because the number of available reference SPs is not enough (less than 3 reference SPs) to perform the Trilateration procedure,

2. Standalone DR technique has huge location error due to the accumulation of the heading-drift error from the beginning of the trajectory,

3. While the UNILS shows better stability and provides SP_IND1_Master location continuously, even when a single connected-SP is available.

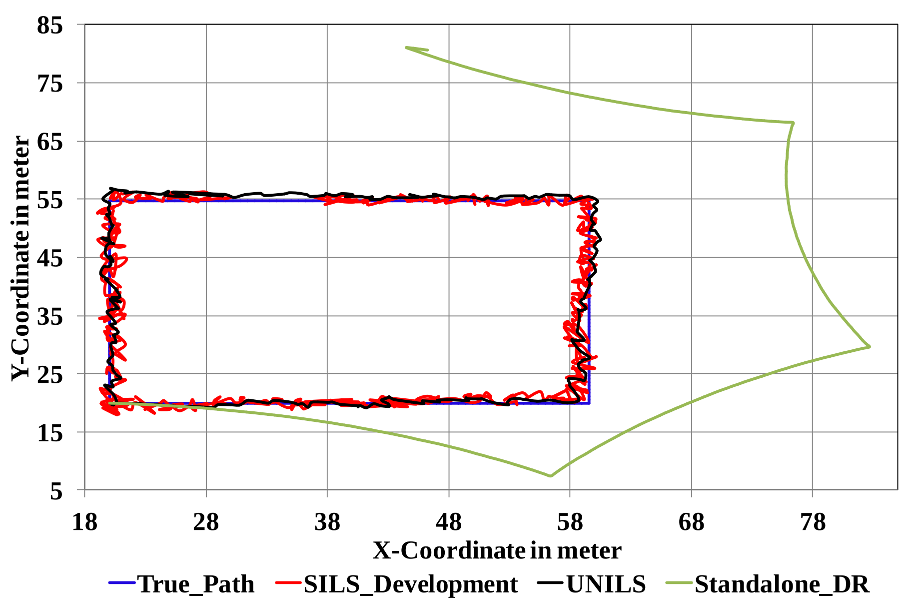


Figure 5-30: Estimated XY coordinates based on indoors localisation schemes.

Figure 5-31 demonstrates the reliability and accuracy of both SILS and UNILS schemes. It can be observed that:

1. The SILS in development mode can achieve location accuracy within 4 meters accuracy when at least three reference SPs are available.

2. The SILS in full-functionality (i.e. including functions of SILS in development mode plus PRP algorithm) has improved the location accuracy and achieved less than 1 meter accuracy, but when there are more than 4 reference SPs are available in the vicinity.

3. The UNILS defines the SP location continuously (even when a single reference SP is available).Further, it is more stable than SILS as well as overcomes the constraint of the number of reference SPs.

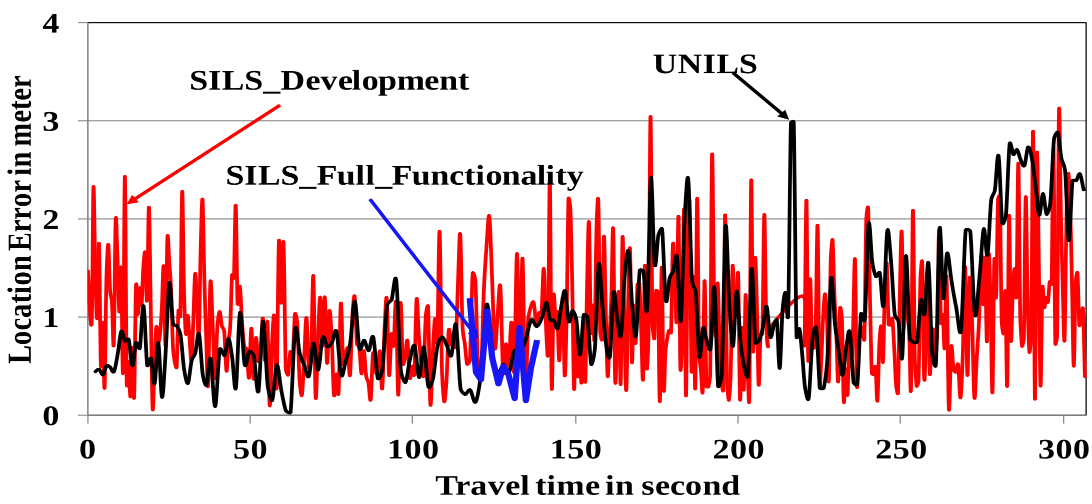


Figure 5-31: Location errors for indoors localisation schemes.

Table 5-3 gives further discussions and comparison about the features of the SILS, the UNILS schemes and the DR stand-alone according to the conducted simulated experiments (including experiments in section 5.4.3, as well)

Table 5-3: SILS & UNILS comparison with standalone DR

Schemes	Accuracy	Stability	Constrained
SILS (Development mode)	Up to 4 meters	Up to 3.39 meters	More than 2 reference positions in (2D)
SILS (Full-functionality)	Over 2 meters	Up to 1.5 meters	More than 4 reference positions in (2D)
Standalone DR	>15 meters (within 1 minute)	1 step (based on walking style)	Unconstrained
UNILS	2.5 meters	Up to 1.27 meters	Unconstrained (even, when a single SP is available)

5.6 Summary

The SILS and the UNILS schemes are presented to offer on-the-go and seamless outdoors-indoors SPs positioning solutions. This is achieved by hybridising onboard SPs GNSS receivers with inertial sensors, BT and WiFi transceivers. These schemes do not need any pre-installed and calibrated localisation infrastructure or prior geographic surveying (works on-the-go). Thus, these schemes are enabled to be viable for use as low cost solution for various LBS. Hop-synchronisations with the GNSS time can be used as an accurate method to estimate the TOF between enabled-BT SPs.

The SILS uses the SMSR and the PRP algorithms to improve the obtained location accuracy. The advancement of processing and memory onboard SPs have made such scheme like the SILS as a good candidate for enabling any indoors-SP to locate itself. The SILS makes many assumptions regarding the cooperation of other SPs in the vicinity to make the SILS reliable and believe it is a realistic scenario for future solutions. Simulations results clearly show that the SILS accuracy to within around 2 meters is possible for 3-walls deep indoors trials.

The UNILS is to integrate DR to the SILS so as to locate indoors-SPs continuously and accurately, even when deep indoors. The UNILS is a good candidate to be implemented on SPs for having low cost and providing 3-meter accuracy. Additionally, the UNILS is more stable even where the signals of WAPs or BT-anchors were not available.

CHAPTER6. DISCUSSIONS, CONCLUSIONS AND FUTURE WORK

6.1 Overview

This chapter concludes the work presented and emphasise the important research contributions of the thesis. First, it summarises the achievements presented in the previous chapters. After highlighting research-achievements, subsequent sections discuss conclusions of the significant findings and future research directions.

6.2 Discussions

The beginning of the thesis describes the research gaps of the existing SPs localisation solutions. Proposing an accurate, on-the-go and seamless outdoors-indoors SPs localisation approach is identified as the primary goal of this work. Current SPs (e.g. Android-based SPs) with no additional hardware or only software & firmware modifications are chosen as the desired computing platform.

After identifying the SP localisation concepts and problems, the methodology starts with formulating the problems and developing solutions. By then, the research comparison framework is followed. In the comparison framework, existing localisation techniques such as Cell-ID, AOA, time-based, RSS-based, DR and MM are explained. Furthermore, the main implementation challenges and capabilities of these techniques are analysed. The list of onboard SP wireless/sensors technologies including GNSS, Cellular, Wi-Fi, BT, acoustic signal, NFC and inertial sensors are investigated that each one of these technologies has its own limitation. Each of these technologies is researched and reviewed separately in the literature review (see chapter two). It is observed that no localisation solution using these technologies/techniques has the ability to locate an SP seamlessly from outdoors into indoors with high accuracy and at low cost, yet. Additionally, it is noted in the literature reviews that hybridisation of these technologies & combining/enhancing applicable techniques can potentially pass this high performance.

Therefore, it is the best of our knowledge and from the literature, this hybridisation is not attempted with SPs anywhere before.

Next, the research design and procedure including a concise description of the main achievements of this research are discussed.

6.1.1 SPs-WAPs Clock Synchronisation

In particular, the **first** achievement in this research study is the SPs-WAPs clocks synchronisation algorithm that uses obtained GNSS time on SPs (when outdoors). This algorithm is used as a preliminary step to help indoors-SPs/WAPs positioning using TOA technique (refer chapter three). The clock synchronisation algorithm is based on the developed accurate WAP clock model. The accuracy of the WAP clock model comes from using:

1. Accurate GNSS time on SPs within few nanoseconds (e.g. ±6 nanoseconds) when outdoors as a reference time to synchronise SPs-WAPs clocks. This reference time is to assure the accuracy of the TOA technique, especially during SPs-WAPs pseudorange estimation

2. Dynamic WAPs clock drift which includes all existing clock noise sources in actual WAPs clocks. Such clock noise sources as: white noise on frequency, flicker noise on frequency, white noise on phase, flicker noise on phase and random walk on frequency. This is necessary to make the WAP clock model as accurate as the real hardware clock inside WAPs.

The quality of SPs-WAPs clock synchronisation is the main factor to provide less WiFi-time based positioning error. Further, performed periodical synchronising of SPs-WAPs clock (getting refresh WAPs clock offset and clock drift) can improve localisation accuracy. To show the quality of the proposed WAPs-SPs clocks synchronisation algorithm a simulated experiment is chosen. This experiment is for various clock-synchronisation intervals based on the DWC model and the static model (as a reference model). The obtained result shows that the proposed model exhibits time errors in different levels (realistic behaviour). On the other hand, time errors of the reference model are static as expected.

6.1.2 Synched-WAPs Localisation

The next step, locating synchronised-WAPs algorithm using time-based localisation technique is proposed as the **second** achievement. Storing such WAPs location information in SPs' memory and/or sharing between a cooperative-group of SPs can make localisation solution fully automated. This is especially useful when SPs can use these WAPs location information (as reference positions) to determine their indoors position.

In chapter four, these two aforementioned algorithms are implemented and evaluated together as a WAP synchronisation/localisation scheme. This is to aid indoors-SPs localisation. Simulation results show that an accuracy of the WAPs location within less than 1 meter can be achieved. This achievement is through participated a group of cooperative-SPs for defining the WAPs location at various distances.

6.1.3 Smart Network for Localisation

To locate indoors-SPs, a proposed indoor localisation scheme, the SILS, is described as the **third** achievement in chapter five (see section 5.4). The SILS focuses on integration of the GNSS, WiFi, and BT measurements on a group of SPs that utilise only available infrastructures. This integration forms a smart network via SP-BT signal connectivity that connects SPs. Therefore, the connected-SPs can cooperatively exchange a map containing all their fixed location and time information to improve indoors-SPs location.

6.1.4 SPs-Pseudoranges Estimation

In addition, the SILS uses BT-Hop-synchronisation with the GNSS time as an accurate method as a **fourth** achievement. Hereby, the SILS estimates pseudoranges between connected-SPs in the network. The accuracy of this method is obtained through:

1. Using the GNSS time as an accurate time-function source to generate timestamps when Master-SPs send and receive POLL-NULL frames, respectively.

2. Generating sequence Hop-frequency values by both Master-SPs and Slaves-SPs. These sequence values guaranteed the fact that both Master-SPs and Slave-SPs clocks are synched in time.

6.1.5 Switch Master/Slave Role Algorithm

To obtain further pseudoranges accuracy improvement between connected-SPs, the SILS implemented a new SMSR algorithm as the **fifth** achievement. All SPs, cooperatively, use this role switching algorithm to perform three tasks. First, the SPs surrender their Master-role. Second, they re-calculate pseudoranges between themselves. Third, the SPs share their calculated pseudoranges with other SPs, in one round within the same location. Hence, this algorithm enables the SPs equipped with all the pseudoranges from all switching master-slave sequence list. By then, SPs can improve their own pseudoranges measurements by applying a LLS fitting technique. The sequence of master-slave list is ordered according to the measured RSS value of the exchanged frames between the SPs. Experimental evaluation of the proposed SMSR algorithm shows that the estimated pseudorange can be improved within 50% in comparison to applying only hop-synchronisation method.

6.1.6 Permutation Reference-Positions Algorithm

The SILS scheme also implements the PRP algorithm as the **sixth** achievement to mitigate DOP issue. This algorithm reduces the indoors-SPs position error caused by Trilateration calculation when outdoors-SPs and/or WAPs are deployed in a poor geometry shape. This novel algorithm uses the calculated HDOP value to evaluate and select the best reference positions that formed a good geometric condition. Simulation experiments of the implemented algorithm during running the SILS on the SPs prove that the location accuracy can be further improved to with around 1.2 meter. Especially, when more than 4 outdoors-SPs (as reference positions) are available in the network.

However, the SILS has failed in locating indoors-SPs, when indoors-SPs are in deep indoors (e.g. 3-walls scenario). This is due to the placed constraint of the PRP algorithm that restricts fixing when the network has 4 (or less) SPs connected. To solve the issue, the number of the SPs availability in the network has been compensated by utilising located synched-WAPs. Moreover, several indoors-scenarios are tested to show the validity of the SILS from many combinations of increasing the number of the participating SPs and WAPs at various distances. Obtained results from these simulated scenarios of the SILS based on the implemented novel algorithms show that around 2-meters accuracy can be achieved, when the number of available reference positions is greater than 4.

6.1.7 Unconstrained Indoors Localisation

The **seventh** new contribution of this thesis is the proposed UNILS in section 5.5. The UNILS realises the full potential of an unconstrained indoors-localisation solution. The UNILS still offers continuous and high SPs-location accuracy even when deep indoors. The UNILS hybridises the measurements of onboard SPs-devices such as BT, inertial sensors and GNSS. This hybridisation works even when broadcasted signals of deployed WAPs and/or BT-anchors are unavailable. This means that the distinct feature of the UNILS provides new infrastructure-free localisation scheme to locate indoors-SPs via any available/cooperative SPs in the vicinity.

The UNILS is also designed to aid inaccuracy calculations of onboard SPs inertial sensors by:

1. Using relative-pseudorange measurements between cooperative/unrestricted network of SPs via BT-signals

2. Utilising/sharing the GNSS-fixed location information when most of the connected-SPs are outdoors.

6.1.8 Wireless/Sensor Measurements Fusion

The **eighth** new achievement is achieving indoors-SPs position based on the UNILS accurately by fusing the following measurements:

1. The measured distance-displacement and heading measurements which come from inertial sensors readings using DR technique.

2. The estimated relative-pseudoranges (based on TOA technique) between connected-SPs via BT-signals.

The process of fusion is done by using Kalman filter. The fusion takes advantages of each of these techniques while mitigating their limitations. For example, estimated pseudorange between SPs based on TOA is accurate but they are not stable. On the other hand, distance-displacement and heading measurements can be smoothly estimated with calibration of inertial sensors. However, their accuracy is degraded quickly over few seconds. The proposed Kalman-based fusion periodically compensates this inaccuracy of DR technique by using BT-pseudoranges and the GNSS fixed location. Furthermore, it reduces the instability of the estimated pseudoranges by using the enhanced DR measurements.

Simulation results of the UNILS (based on Android-SPs network implementations) have clearly showed that it can provide seamless indoors-SPs positioning within 3-meters accuracy. In the experiments, it is assumed that the UNILS can locate indoors-SPs at various deep indoors situations, even when WAPs or BT-anchors signal is considered not able to be used.

6.3 Conclusions

On-the-go and seamless outdoors-indoors localisation solutions have become an important research field over the last few years. The solutions accomplish the full capacity of most SPs-LBS applications. Moreover, it is generally accepted that outdoor localisation solutions are accurate to locate SPs within few meters, using the onboard GNSS technology as an example. Nonetheless, indoors localisation solutions do not satisfy reliability and availability of current LBS applications. In addition, all existing indoors localisation solutions are purposely designed with specialised hardware. Therefore, this research study proposes novel solutions to hybridise the GNSS with other wireless/sensor technologies onboard SPs that does not require dedicated hardware. The dedicated hardware includes host server/database, Internet connection and pre-calibration/surveying. The proposed solutions can be performed anywhere, anytime, and with any SP.

The novelties of this research study centre on:

1. Achieving better SP-position accuracy solution

2. Providing outdoors-indoors seamless SPs localisation solution

3. Offering on-the-go, easy-deployable and low-cost localisation solution

In particular, when these solutions are implemented inside any LBS applications on SPs the following advantages can be taken:

1. Reducing the cost due to no need of any pre-installed and calibrated localisation infrastructure.

2. Minimising the required memory, connection/interaction traffic and processing time owing to no need to use dedicated Internet based database/server. Thus it saves battery-power consumption on SPs.

3. Increasing the utilisation of SPs geographic location. Additionally, it meets the LBS-users' requirements because the solutions work on-the-go, anytime and anywhere.

4. Easily integrating the solutions to any SPs LBS applications, once the software component is installed.

Different hypotheses and scenarios based on the simulation (using OPNET) and live trials implementation have proved the viability of these schemes. Therefore, all the proposed schemes, in this study, as a localisation solution is a good candidate for seamless LBS localisation applications on SPs. Because the proposed schemes all together are accurate, low-cost, on-the-go, anytime, anywhere and locate SPs without using any pre-installed and calibrated infrastructure. In addition, the research achievements of this study can be exploited to produce commercial products to be implemented on SPs within a particular LBS application. Further conclusions on this work study in terms of commercial impact, social impact and security issues are discussed in next subsections.

6.2.1 Commercial Impact and Promising Applications

The research contributions of this thesis can be exploited to produce commercial products to be implemented on SPs within a particular LBS application. This is because:

1. The proposed schemes are enabled to be applicable for use as low cost solution for most LBS applications (such as patient or disabled monitoring, security & safety and POI applications). Because of the schemes can be implemented on SPs without deploying or calibrating additional hardware.

2. The schemes provide accurate, on-the-go and seamless positioning solutions. Moreover, several LBS applications are depending on these features. For example, an advertisement in large stores or guidance in museums with SPs is feasible if a precise location of the SPs is obtained seamlessly from outdoors into indoors.

6.2.2 Social Impact

The proposed schemes can LBSN applications. The LBSN applications are part of a new feature of social networking tools. In addition, the LBSN applications are the aggregation between the LBS and OSN. LBSN applications enable the cooperated group-of-users looking up the location of another "friend" using SPs [140].

The schemes bring more accurate and cost-effective health and social care to the infirm/old people's houses just by using their SPs. The social and health care provide such people with a pleasant time at home. For example, AAL is a new project [139]. The AAL concerns how technology is smartly used to serve disabled and elderly people in a chosen circumference. As a result, the preferred project can provide a new invented settlement with a great impact on the goodness of life.

6.2.3 Security and Safety Impact

The proposed schemes can supply new features/factors like accurate/always location and time to enhance authentication process in most mCommerce applications. The preferred project is able to provide new privacies or factors such as constant location and time to promote authentication procedure in many mCommerce applications. For instance, these days, many banking systems that require the user's location as a modern factor of certificate [141]. The location-based certificate has several advantages. For instance, if the owner of the account likes to work on his/her account, it needs confirming his/her location at the beginning. When he/she is at the bank office or at home, he can access the account. Nonetheless, if he/she is in another place he cannot enter his/her bank account.

The cooperative feature of the proposed schemes is also a good candidate solution to monitor indoors-firefighters locations accurately with the enhancement of their safety in many conditions. For instance, firefighters always try to enter buildings with full of smoke or without light and, in such low-vision; look for the environment to establish a shared knowledge of the condition, and to determine the mission of the involvement [142].

6.4 Future Directions and Research Opportunities

The simulation and experimental results provide a proof-of-concept of the proposed seamless outdoors-indoors SP localisation schemes. Several important aspects of these schemes based on hybridisation of the GNSS with the WiFi, the BT and the onboard inertial sensors have been studied in this thesis. Nonetheless, there are still some open research problems worth further consideration.

6.3.1 Solving Hardware Implementation Issues

We hope that we can fully-access all of the hardware implementation issues of the SILS on off-the-shelf SPs (Android-based SPs) such as:

1. Accessing to the accurate timing from the onboard SP-GNSS receiver used in the SILS and the UNILS.

2. Modifying the WiFi transceiver firmware to support WAP-passive scanning. That is, supplying monitor mode for the WiFi transceivers on SPs for WAP-beacon scanning.

3. Accessing BT MAC and BT Baseband layer functions to support and use the GNSS time as a timestamps during sending and receiving POLL/NULL frames. As well as, implementing the proposed algorithms including SMSR and PRP at this layer.

Overcoming these limitation-issues are future challenges to support full-functionality of the SILS & UNILS in the LBSs applications on the SPs.

6.3.2 More Environmental Factors Consideration

Restrictions forced by the environmental factors will be added into the SILS and the UNILS in the future. Such restrictions as obstacles, moving objects/persons, dynamic indoors structure and other sources incorporate the NLOS issue. In addition, calculating

3D SP-position, i.e. locating SPs in XYZ coordinates, provides tracking SPs between the floors of buildings.

6.3.3 Enhancing Trilateration Algorithm

The Trilateration algorithm (based on TOA technique) is used to locate WAPs when SPs are outdoors. It is also used to locate indoors-SPs when WAPs or outdoors-SPs location information is defined. The Trilateration algorithm presented in the thesis is simple. Likewise, it can be enhanced or modified in several ways, such as:

1. In the present algorithm, to estimate pseudoranges only timestamp measurements of WAPs and/or SPs can affect their pseudoranges estimation. However, the concept of weighting from multiple WAPs/SPs signals based on previous measurements are not considered in enhancing the TOA measurements. Such enhancement can reduce the instability of the measurements. New weight values for each received signals will be assigned at each iteration. This will be achieved by a training process that will change the current weight values based on some statistics. The statistics should be associated with a set of previous measurements by using conditional filtering (e.g. Particle Filtering).

2. The estimated position of a WAP is chosen simply by measuring the pseudoranges from multiple fixed locations of a single-SP or from multiple-SPs. These pseudoranges are measured by using received WAPs beacon signals. Better results can be potentially obtained by fusing onboard SPs DR measurements with WAPs beacon signals. For example, using the measured distance-displacement of SPs (via accelerometer sensor) and sharing this information cooperatively enhance the estimated WAPs position on each connected-SP in the network.

3. Map information can be also fused by the Trilateration algorithm to aid the pseudorange estimation. For example, the number of obstacles (e.g. walls) and other noise sources between the SPs and the WAPs will be avoided which affect the pseudorange estimation.

6.3.4 DR Measurements Improvement

Further SPs-location accuracy improvements can be obtained for the UNILS scheme by using other source information. The improvements keep the unconstrained feature, yet. For example:

1. Map-buildings information can eliminate small errors in the estimated heading. These small errors introduce large errors in the SP-location estimation over few seconds. By using this map information, the heading error can be removed earlier when the path of the walk with corridors found in the map is possibly aligned. In addition, the step length can be calibrated by comparing the length of the paths with the numbers of the steps needed to complete the path. This is important when relative-pseudoranges estimation between SPs is considered unavailable.

2. User-walking slowly/quickly, taking stairs and elevator should be taken in more consideration into the integration of DR and TOA measurements. Adding these considerations into the UNILS will realise and complete indoors-SPs localisation scenarios. Note: this will include appropriate changes in the used Kalman filter states, prediction and update equations.

.

REFERENCES

[1] Agency, European GNSS, "GNSS Market Report," no. issue 4, 2015.

[2] S. E. C. Ltd, "Next is treating yourself right. Features & Specifications of Samsung Galaxy 6," Samsung Premium Care. , 2015.

[3] Vladimir, Lazovic and Nikola, Minic and Milan, Tair and others, "Location-Based Applications for Smartphones," in *International Scientific Conference of IT and Business-Related Research*, 2015.

[4] J. Ryoo, H. Kim and S. Das, "Geo-fencing: Geographical-fencing based energy-aware proactive framework for mobile devices," in *Quality of Service (IWQoS), 2012 IEEE 20th International Workshop on*, 2012.

[5] Kaur, P. and Sharma, S., "Google Android a mobile platform: A review," in *Engineering and Computational Sciences (RAECS), 2014 Recent Advances in*, March 2014.

[6] Hejc, Gerhard and Seitz, Jochen and Vaupel, Thorsten, "Bayesian sensor fusion of Wi-Fi signal strengths and GNSS code and carrier phases for positioning in urban environments," in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, 2014.

[7] Nilsson, John-Olof and Rantakokko, Jouni and Handel, Peter and Skog, Isaac and Ohlsson, Martin and Hari, KVS, "Accurate indoor positioning of firefighters using dual foot-mounted inertial sensors and inter-agent ranging," in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, 2014.

[8] Lymberopoulos, Dimitrios and Liu, Jie and Yang, Xue and Choudhury, Romit Roy and Handziski, Vlado and Sen, Souvik, "A realistic evaluation and comparison of indoor location technologies: experiences and lessons learned," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, 2015.

[9] De Angelis, Guido and Baruffa, Giuseppe and Cacopardi, Saverio, "GNSS/Cellular Hybrid Positioning System for Mobile Users in Urban Scenarios," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 1, pp. 313--321, 2013.

[10] Kaushal, Aakash and Malhotra, Anupriya and Vohra, Hanish, "Location-based services & positioning technique using SUPL (Secure User Plane Location)," *L&T*

Technology Services, 2015.

[11] M. T. López, "THE FUTURE OF POSITIONING: OPPORTUNITIES AND CHALLENGES," TUT W3C Web Technology Day: Location and Positioning, TUT, Tampere, Finland, 11th September, 2013.

[12] Kannan, Balajee and Kothari, Nisarg and Gnegy, Chet and Gedaway,, "Localization, Route Planning, and Smartphone Interface for Indoor," in *Cooperative Robots and Sensor Networks*, Springer Berlin Heidelberg, 2014, pp. 39-59.

[13] Binsabbar, Mohammed and Zhang, Ning, "An iPhone Application for Providing iBeacon-based Services to Students," 2014.

[14] M. BOTTA and M. SIMEK, "Adaptive Distance Estimation Based on RSSI in 802.15.4 Network," in *RADIOENGINEERING*, 2013.

[15] Tian, Zengshan and Zhang, Yuan and Zhou, Mu and Liu, Yu, "Pedestrian dead reckoning for MARG navigation using a smartphone," *EURASIP Journal on Advances in Signal Processing*, vol. 1, pp. 1-9, 2014.

[16] R. Henniges, "Current approches of Wifi Positioning," in *SERVICE-CENTRIC NETWORKING*, 2012.

[17] Schauer, L. and Dorfmeister, F. and Maier, M., "Potentials and limitations of WIFI-positioning using Time-of-Flight," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, Oct 2013.

[18] Yi Sun and Yubin Zhao and Schiller, J., "An indoor positioning system based on inertial sensors in smartphone," in *Wireless Communications and Networking Conference (WCNC), 2015 IEEE*, March 2015.

[19] Boonsriwai, S. and Apavatjrut, A., "Indoor WIFI localization on mobile devices," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, May 2013.

[20] Kirchhof, Nicolaj, "Optimal placement of multiple sensors for localization applications," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, 2013.

[21] Li, X Rong and Zhao, Zhanlue, "Measures of performance for evaluation of

estimators and filters," in *International Society for Optics and Photonics*, 2001.

[22] Liu, Jingbin and Chen, Ruizhi and Pei, Ling and Guinness, Robert and Kuusniemi, Heidi, "A hybrid smartphone indoor positioning solution for mobile lbs," *Sensors,* vol. 12, no. 12, pp. 17208--17233, 2012.

[23] Bisio, I. and Lavagetto, F. and Marchese, M. and Sciarrone, A., "Energy efficient WiFi-based fingerprinting for indoor positioning with smartphones," in *Global Communications Conference (GLOBECOM), 2013 IEEE,* Dec 2013.

[24] Cullen, Gary and Curran, Kevin and Santos, Jose, "Cooperatively extending the range of Indoor Localisation," in *Signals and Systems Conference (ISSC 2013), 24th IET Irish*, 2013.

[25] Jizhe Zhou, Beihang University, China; Yueying Zhang, "A BDS Based Positioning framework with CoMP over LTE-A Networks," in *ION GNSS+ 2014,* Tampa, Florida, USA, 2014.

[26] Strout, Aaron and Schneider, Mike, Location Based Marketing For Dummies, John Wiley & Sons, 2011.

[27] Anuar, Faiz and Gretzel, Ulrike, "Privacy Concerns in the Context of Location-Based Services for Tourism," in *ENTER 2011 Conference, Innsbruck, Austria,* 2011.

[28] Lopez-de-Ipina, Diego and Klein, Bernhard and Guggenmos, Christian and Perez, Jorge and Gil, Guillermo, "User-Aware semantic location models for service provision," in *International Symposium on Ubiquitous Computing and Ambient Intelligence, Riviera Maya, Mexico*, 2011.

[29] Yun, Haejung and Han, Dongho and Lee, Choong C, "Understanding the Use of Location-Based Service Applications: Do Privacy Concerns Matter?," *Journal of Electronic Commerce Research,* vol. 14, no. 3, p. 215, 2013.

[30] A. Bensky, Wireless Positioning Technologies and Applications, 685 Canton Street, Norwood, MA 02062: ARTECH HOUSE, INC., 2008.

[31] S. Koenig, M. Schmidt and C. Hoene, "Multipath mitigation for indoor localization based on IEEE 802.11 time-of-flight measurements," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a,* 2011.

[32] Chengkai Huang and Gong Zhang and Zhuqing Jiang and Chao Li and Yupeng Wang and Xueyang Wang, “Smartphone-based indoor position and orientation tracking fusing inertial and magnetic sensing,” in *Wireless Personal Multimedia Communications (WPMC), 2014 International Symposium on*, Sept, 2014.

[33] D. Park and J. Park, “An Enhanced Ranging Scheme Using WiFi RSSI Measurements for Ubiquitous Location,” in *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on*, 2011.

[34] A. Roxin, J. Gaber, M. Wack and A. Nait-Sidi-Moh, “Survey of wireless geolocation techniques,” in *Globecom Workshops, 2007 IEEE*, 2007.

[35] E. Mok, “Using outdoor public WiFi and GPS integrated method for position updating of knowledge-based logistics system in dense high rise urban environments,” in *Supply Chain Management and Information Systems (SCMIS), 2010 8th International Conference on*, 2010.

[36] Huber, Denis, “Background Positioning for Mobile devices-Android vs. iphone,” in *Joint Conference of IEEE Computer & Communication Societies*, 2011.

[37] R. Padilla, “Apple Retail Stores to Integrate iBeacon Systems to Assist with Sales and Services,” *MacRumores* , November 16, 2013.

[38] Sen, Souvik and Lee, Jeongkeun and Kim, Kyu-Han and Congdon, Paul, “Avoiding multipath to revive inbuilding wifi localization,” in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, 2013.

[39] Alam, N. and Dempster, A.G., “Cooperative Positioning for Vehicular Networks: Facts and Futur,” in *Intelligent Transportation Systems, IEEE Transactions on*, Dec 2013.

[40] Zhou, Z. and Yang, Z. and Wu, C. and Shangguan, L. and Cai, H. and Liu, Y. and Ni, L., “WiFi-based Indoor Line-Of-Sight Identification,” *Wireless Communications, IEEE Transactions on,* vol. PP, no. 99, pp. 1-11, 2015.

[41] J. Lee, Z. Lin, P. Chin and K. Yar, “One way ranging time drift compensation for both synchronized and non-synchronized clocks,” in *System Science and Engineering (ICSSE), 2010 International Conference on*, 2010.

[42] Jaime Lloret, Jesus Tomas, Alejandro Canovas, Irene Bellver, “A Geopositioning System Based on WiFi Networks,” in *The Seventh International Conference on Networking and Services (ICNS 2011)*, Venecia (Italia), 10 May 2011.

[43] Hoflinger, F. and Rui Zhang and Hoppe, J. and Bannoura, A. and Reindl, L.M. and Wendeberg, J. and Buhrer, M. and Schindelhauer, C., “Acoustic Self-calibrating System for Indoor Smartphone Tracking (ASSIST),” in *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, Nov 2012.

[44] Martin-Escalona, I. and Barcelo-Arroyo, F. and Zola, E., “Software based measurement of round trip time observables for location in IEEE 802.11 networks,” in *Telecommunications (ConTEL), 2013 12th International Conference on*, June 2013.

[45] S. Jin, Global Navigation Satellite Systems: Signal, Theory and Applications, InTech, Chapters published, February 03, 2012.

[46] Elbes, Mohammed and Al-Fuqaha, Ala and Anan, Muhammad, “A precise indoor localization approach based on particle filter and dynamic exclusion techniques,” *Network Protocols and Algorithms*, vol. 5, no. 2, pp. 50--71, 2013.

[47] Chatterjee, S. and Banerjee, P. and Nasipuri, M., “Enhancing Accuracy of Localization for Primary Users in Cognitive Radio Networks,” in *Computational Intelligence and Networks (CINE), 2015 International Conference on*, 2015.

[48] Yi Sun and Yubin Zhao and Schiller, J., “An autonomic indoor positioning application based on smartphone,” in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, 2014.

[49] Yassin, M. and Rachid, E., “A survey of positioning techniques and location based services in wireless networks,” in *Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International Conference on*, 2015.

[50] Mariakakis, Alex and Sen, Souvik and Lee, Jeongkeun and Kim, Kyu-Han, “Single access point based indoor localization,” in *Proc. of ACM MobiSys*, 2014.

[51] Deng Zhongliang and Yu Yanpei and Yuan Xie and Wan Neng and Yang Lei, “Situation and development tendency of indoor positioning,” *Communications, China*, vol. 10, no. 3, pp. 42-55, 2013.

[52] C. Jeffery, An Introduction into GNSS, First Edition ed., 1120-68th Avenue N.E.

Calgary, Alberta, Canada, TeE8S5: NovAtel Inc., 2010.

[53] Wang, Lei and Groves, Paul D and Ziebart, Marek K, “GNSS shadow matching: improving urban positioning accuracy using a 3D city model with optimized visibility scoring scheme,” *Navigation*, vol. 60, no. 3, pp. 195--207, 2013.

[54] N. Mahiddin, N. Safie, E. Nadia, S. Safei and E. Fadzli, “INDOOR POSITION DETECTION USING WIFI AND TRILATERATION TECHNIQUE,” in *The International Conference on Informatics and Applications (ICIA2012)*, 2012.

[55] C. Rizos, G. Roberts, J. Barnes and N. Gambale, “Locata: A new high accuracy indoor positioning system,” in *Proc. Int. Conf. on Indoor Positioning \& Indoor Navigation, Zurich, Switzerland*, 2010.

[56] N. Kohtake, S. Morimoto, S. Kogure and D. Manandhar, “Indoor and Outdoor Seamless Positioning using Indoor Messaging System and GPS,” in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011.

[57] Subhan, Fazli and Hasbullah, Halabi and Rozyyev, Azat and Bakhsh, Sheikh Tahir, “Indoor positioning in bluetooth networks using fingerprinting and lateration approach,” in *Information Science and Applications (ICISA), 2011 International Conference on*, 2011.

[58] Teixeira, Renata and Moon, Sue B and Uhlig, Steve, Passive and Active Network Measurement: 10th International Conference, PAM 2009, Seoul, Korea, April 1-3, 2009, Proceedings, Springer, 2009.

[59] Della Rosa, Francescantonio and Pelosi, Mauro and Nurmi, Jari, “Human-induced effects on rss ranging measurements for cooperative positioning,” *International Journal of Navigation and Observation*, p. 13, 2012.

[60] Xiao, Zhuoling and Wen, Hongkai and Markham, Andrew and Trigoni, Niki, “Robust pedestrian dead reckoning (R-PDR) for arbitrary mobile device placement,” in *nternational Conference on Indoor Positioning and Indoor Navigation*, 2015.

[61] L. Pei, R. Chen, J. Liu, Z. Liu, H. Kuusniemi, Y. Chen and L. Zhu, “Sensor assisted 3D personal navigation on a smart phone in GPS degraded environments,” in *Geoinformatics, 2011 19th International Conference on*, 2011.

[62] Paul Craven, Ronald Wong, Neal Fedora, and Paul Crampton, “Studying the

Effects of Interference on GNSS Signals”.

[63] Cherian, Suma S and Rudrapatna, Ashok N, “LTE location technologies and delivery solutions,” *Bell Labs Technical Journal,* vol. 18, no. 2, pp. 175--194, 2013.

[64] I. Adusei, K. Kyamakya and K. Jobmann, “Mobile positioning technologies in cellular networks: an evaluation of their performance metrics,” in *MILCOM 2002. Proceedings*, 2002.

[65] A. Waadt, G. Bruck and P. Jung, “An overview of positioning systems and technologies,” in *Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on*, 2009.

[66] T. FORD, “TruePosition, ABI test network-based vs. handset-based 911 solutions,” *NETWORK INFRASTRUCTURE,* 2011.

[67] Farid, Zahid and Nordin, Rosdiadee and Ismail, Mahamod, “Recent advances in wireless indoor localization techniques and system,” *Journal of Computer Networks and Communications,* p. 12, 2013.

[68] X. Liu, S. Zhang, J. Quan and X. Lin, “The experimental analysis of outdoor positioning system based on fingerprint approach,” in *Communication Technology (ICCT), 2010 12th IEEE International Conference on*, 2010.

[69] Radaelli, Laura and Jensen, Christian S, “Towards fully organic indoor positioning,” in *Proceedings of the Fifth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, 2013.

[70] Subbu Meiyappan, Arun Raghupathy, and Ganesh Pattabiraman, “Positioning in GPS Challenged Locations -NextNav's Metropolitan Beacon System,” in *International Conference on Indoor Positioning and Indoor Navigation*, 2013.

[71] Faragher, R and Harle, R, “SmartSLAM--an efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing,” in *ION GNSS*, 2013.

[72] J. Xiong and K. Jamieson, “ArrayTrack: A Fine-Grained Indoor Location System,” *RN,* vol. 11, p. 19, 2011.

[73] T. Gallagher, B. Li, A. Kealy and A. Dempster, “Trials of commercial Wi-Fi positioning systems for indoor and urban canyons,” in *IGNSS 2009 Symposium on*

GPS/GNSS, 2009.

[74] P. Zandbergen, “Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning,” *Transactions in GIS,* vol. 13, no. s1, pp. 5-25, 2009.

[75] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter and others, “Place lab: Device positioning using radio beacons in the wild,” *Pervasive Computing,* pp. 301-306, 2005.

[76] Sensewhere LTD, “Indoor Location Technology Leaders,” 2011.

[77] R. a. I. C. I. The Communications Security, “Indoor Location Test Bed Report,” E9-1-1 Location Accuracy, March 14, 2013.

[78] TruePosition, “U-TDOA Enabling New Location-based Safety and Security Solutions,” *White Paper, USA,* October 2008.

[79] M. Youssef, A. Youssef, C. Rieger, U. Shankar and A. Agrawala, “Pinpoint: An asynchronous time-based location determination system,” in *Proceedings of the 4th international conference on Mobile systems, applications and services*, 2006.

[80] C. Hoene and J. Willmann, “Four-way TOA and software-based trilateration of IEEE 802.11 devices,” in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, 2008.

[81] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun and A. Aggarwal, “Efficient, generalized indoor wifi graphslam,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011.

[82] T. Manodham, L. Loyola and T. Miki, “A novel wireless positioning system for seamless internet connectivity based on the WLAN infrastructure,” *Wireless Personal Communications,* vol. 44, no. 3, pp. 295-309, 2008.

[83] Y. Zhao, M. Li and F. Shi, “Indoor Radio Propagation Model Based on Dominant Path,” *Int'l J. of Communications, Network and System Sciences,* vol. 3, no. 3, pp. 330-337, 2010.

[84] C. Feng, W. Au, S. Valaee and Z. Tan, “Compressive sensing based positioning using RSS of WLAN access points,” in *INFOCOM, 2010 Proceedings IEEE*, 2010.

[85] S. Koenig, M. Schmidt and C. Hoene, “Multipath mitigation for indoor localization based on IEEE 802.11 time-of-flight measurements,” in *World of Wireless, Mobile*

and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a, 2011.

[86] I. Shafer and M. L. Chang, “Movement detection for power-efficient smartphone WLAN localization,” in *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*, New York, NY, USA, 2010.

[87] A. G. D. B. Li and C. Rizos, “Positioning in environments where GPS fails.,” Sydney (Australia), 2010.

[88] S. Zirari, P. Canalda and F. Spies, “WiFi GPS based combined positioning algorithm,” in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, 2010.

[89] Torok, Agoston and Nagy, Akos and Kovats, Laszlo and Pach, Peter, “DREAR-Towards Infrastructure-Free Indoor Localization via Dead-Reckoning Enhanced with Activity Recognition,” in *Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014 Eighth International Conference on*, 2014.

[90] Iwase, T. and Shibasaki, R., “Infra-free indoor positioning using only smartphone sensors,” in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, 2013.

[91] Renaudin, Valerie and Demeule, Vincent and Ortiz, Matt, “Adaptative pedestrian displacement estimation with a smartphone,” in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, 2013.

[92] Taniuchi, Daisuke and Liu, Xiaopeng and Nakai, Daisuke and Maekawa, Takuya, “Spring Model Based Collaborative Indoor Position Estimation With Neighbor Mobile Devices,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 9, no. 2, pp. 268--277, 2015.

[93] Ruijun Fu and Yunxing Ye and Pahlavan, K., “Heterogeneous cooperative localization for social networks with mobile devices,” in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, 2012.

[94] Kaikai Liu and Qiuyuan Huang and Jiecong Wang and Xiaolin Li and Wu, D.O., “Improving GPS Service via Social Collaboration,” in *Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference on*, 2013.

[95] T. Oshin, S. Poslad and A. Ma, “Improving the Energy-Efficiency of GPS Based Location Sensing Smartphone Applications,” in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on,* 2012.

[96] B. Li, A. G. Dempster and C. Rizos, “Positioning in environments where GPS fails.,” Sydney (Australia), 2010.

[97] B. Ogilvie, “Clock solutions for WiFi (IEEE 802.11),” *Saronix (tm) application note,* 2003.

[98] Bahillo, Alfonso and Mazuelas, Santiago and Lorenzo, Ruban M and Fernandez, Patricia and Prieto, Javier and Duran, Raman J and Abril, Evaristo J, “Hybrid RSS-RTT localization scheme for indoor wireless networks,” *EURASIP Journal on Advances in Signal Processing,* p. 17, 2010.

[99] Yang, Ting and Yang, Qing and Lim, Alvin, “Driver layer approach to time-of-arrival ranging in IEEE 802.11 g networks,” in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, 2012.

[100] C. Arackaparambil, S. Bratus, A. Shubina and D. Kotz, “On the reliability of wireless fingerprinting using clock skews,” in *Proceedings of the third ACM conference on Wireless network security,* 2010.

[101] Orosz, Peter and Skopko, Tamas, “Performance evaluation of a high precision software-based timestamping solution for network monitoring,” *International Journal on Advances in Software,* vol. 4, no. 1, 2011.

[102] Kim, R and Nagayama, T and Jo, H and Spencer Jr, BF, “Preliminary study of low-cost GPS receivers for time synchronization of wireless sensors,” in *International Society for Optics and Photonics,* 2012.

[103] Olson, Daniel D and Bissonette, John A and Cramer, Patricia C and Green, Ashley D and Davis, Scott T and Jackson, Patrick J and Coster, Daniel C, “Monitoring wildlife-vehicle collisions in the information age: how smartphones can improve data collection,” *PLoS ONE 9(6): e98613.* , June 4, 2014.

[104] Shoab, M. and Jain, K. and Anulhaq, M. and Shashi, M., “Development and implementation of NMEA interpreter for real time GPS data logging,” in *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, 2013.

[105] F. Ring, A. Nagy, G. Gaderer and P. Loschmidt, “Clock synchronization simulation for wireless sensor networks,” in *Sensors, 2010 IEEE*, 2010.

[106] G. Gaderer, A. Nagy, P. Loschmidt and T. Sauter, “Achieving a Realistic Notion of Time in Discrete Event Simulation,” *International Journal of Distributed Sensor Networks,* vol. 2011, 2011.

[107] M. Shin, C. Park and S. Lee, “Atomic clock error modeling for GNSS software platform,” in *Position, Location and Navigation Symposium, 2008 IEEE/ION*, 2008.

[108] P. Ranganathan and K. Nygard, “Time synchronization in wireless sensor networks: a survey,” *International journal of UbiComp (IJU),* vol. 1, no. 2, p. 92, 2010.

[109] H. Zhu, L. Ma and B. Ryu, *System and method for clock modeling in discrete-event simulation,* Google Patents, 2008.

[110] Mukherjee, Biswajeet and Vishwakarma, Dinesh K, “Application of Finite Difference Time Domain to Calculate the Transmission Coefficient of an Electromagnetic Wave Impinging Perpendicularly on a Dielectric Interface with Modified MUR-I ABC,” *Defence Science Journal,* vol. 62, no. 4, pp. 228-235, 2012.

[111] A. Mahmood, G. Gaderer and P. Loschmidt, “Software support for clock synchronization over IEEE 802.11 wireless LAN with open source drivers,” in *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2010 International IEEE Symposium on*, 2010.

[112] F. Cristian and C. Fetzer, “Probabilistic internal clock synchronization,” in *Reliable Distributed Systems, 1994. Proceedings., 13th Symposium on*, 1994.

[113] D. Mills, “Simple network time protocol (SNTP) version 4 for IPv4, IPv6 and OSI,” 2006.

[114] T. Cooklev, J. Eidson and A. Pakdaman, “An implementation of IEEE 1588 over IEEE 802.11 b for synchronization of wireless local area network nodes,” *Instrumentation and Measurement, IEEE Transactions on,* vol. 56, no. 5, pp. 1632-1639, 2007.

[115] K. Noh, E. Serpedin and K. Qaraqe, “A new approach for time synchronization in wireless sensor networks: Pairwise broadcast synchronization,” *Wireless*

Communications, IEEE Transactions on, vol. 7, no. 9, pp. 3318-3322, 2008.

[116] Youn, Seongwook, “A comparison of clock synchronization in wireless sensor networks,” *International Journal of Distributed Sensor Networks,* vol. 2, no. 1, p. 10, 2013.

[117] Hui Tian and Linyuan Xia and Mok, E., “A Novel Method for Metropolitan-scale Wi-Fi Localization Based on Public Telephone Booths,” in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, 2010.

[118] S. Chun, S. Lee, J. Nah, J. Choi and J. Park, “Localization of Wi-Fi Access Point using smartphone's GPS information,” in *Mobile and Wireless Networking (iCOST), 2011 International Conference on Selected Topics in,* 2011.

[119] N. Kodippili and D. Dias, “Integration of fingerprinting and trilateration techniques for improved indoor localization,” in *Wireless And Optical Communications Networks (WOCN), 2010 Seventh International Conference On,* 2010.

[120] Jones, Russel K and Alizadeh-Shabdiz, Farshid and Morgan, Edward J and Shean, Michael G, *Server for updating location beacon database,* Google Patents, US Patent 8,837,363, 2014.

[121] A. Ibrahim and D. Ibrahim, “Real-time GPS based outdoor WiFi localization system with map display,” *Advances in Engineering Software,* vol. 41, no. 9, pp. 1080-1086, 2010.

[122] J. Koo and H. Cha, “Localizing WiFi access points using signal strength,” *Communications Letters, IEEE,* vol. 15, no. 2, pp. 187-189, 2011.

[123] D. Han, D. Andersen, M. Kaminsky, K. Papagiannaki and S. Seshan, “Access point localization using local signal strength gradient,” *Passive and Active Network Measurement,* pp. 99-108, 2009.

[124] Gupta, Shashank Kumar and Box, Simon and Wilson, R Eddie, “Low cost infrastructure free form of indoorpositioning,” in *2014 International Conference on Indoor Positioning and Indoor Navigation*, 2014.

[125] Cullen, Gary and Curran, Kevin and Santos, Jose and Maguire, Gearoid and Bourne, Denis, “CAPTURE: Cooperatively applied positioning techniques utilizing range extensions,” in *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on,* 2014.

[126] Cho, Seong Yun and Kim, Joo Young and Enkhtur, Munkhzul, “P2P ranging-based cooperative localization method for a cluster of mobile nodes containing IR-UWB PHY,” in *Electronics and Telecommunications Research Institute*, 2013.

[127] D'mello, Sejal and Aalam, Zahir, “AFH Techniques--Survey Based on Probabilistic Channel Usage,” *International Journal of Research in Computer Engineering & Electronics*, vol. 3, no. 2, pp. 1-6, 2014.

[128] Huang, Jun and Albazrqaoe, Wahhab and Xing, Guoliang, “BlueID: A practical system for Bluetooth device identification,” in *INFOCOM, 2014 Proceedings IEEE*, 2014.

[129] B. Seymour, “Bluetooth Master/Slave Communications and Sniff/Sniff Sub-Rating Modes White Paper,” *Bluetooth SIG*, p. 21, 14 8 2008.

[130] Pang, Qixiang and Leung, Victor, “Improved channel classification and scheduling for non-collaborative Bluetooth/WLAN coexistence,” in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, 2006.

[131] Chen, Ling-Jyh and Kapoor, Rohit and Sanadidi, Medy Young and Gerla, Mario, “Enhancing Bluetooth TCP throughput via link layer packet adaptation,” in *Communications, 2004 IEEE International Conference on*, 2004.

[132] Seradio, Carlos and Coutinho, Luis and Reigoto, Luis and Matias, Joao and Correia, Aldina and Mestre, Pedro, “A lightweight indoor localization model based on motley-keenan and cost,” *International Association of Engineers*, 2012.

[133] A. R. H. R. a. o. Pratama, “Smartphone-based pedestrian dead reckoning as an indoor positioning system,” in *System Engineering and Technology (ICSET), 2012 International Conference on*, 2012.

[134] V. a. C. C. a. P. F. Renaudin, “Quaternion based heading estimation with handheld MEMS in indoor environments,” in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, 2014.

[135] D. S. a. N. J. a. H. P. Colomar, “Smoothing for ZUPT-aided INSs,” in *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, 2012.

[136] Zhuang, Yuan and Chang, Hsiu Wen and El-Sheimy, Naser, “A MEMS multi-sensors system for pedestrian navigation,” in *China satellite navigation conference*

(CSNC) 2013 proceedings, 2013.

[137] Liu, Xingchuan and Man, Qingshan and Lu, Henghui and Lin, Xiaokang, "Wi-Fi/MARG/GPS integrated system for seamless mobile positioning," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, 2013.

[138] Skog, Isaac and Nilsson, John-Olof and Handel, Peter, "Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, 2010.

[139] Ando, Bruno and Baglio, Salvatore and Campisi, Sebastiano and Lombardo, Cristian O and Marletta, Vincenzo and Pergolizzi, Elisa, "A Novel Tracking System for AAL Based on Smartphone Technology," *Ambient Assisted Living, Springer,* vol. Part VI, pp. 243-250, 2014.

[140] Fusco, Sarah Jean and Abbas, Roba and Michael, Katina and Aloudat, Anas, "Location-based social networking: impact on trust in relationships," *Technology and Society Magazine, IEEE,* vol. 31, no. 2, pp. 39-50, 2012.

[141] Rohit Joshi, Prince Gupta and Mahendra Hinde, "Location Based Authentication For E-Banking," *International Journal of Latest Trends in Engineering and Technology (IJLTET),* vol. 5, no. 2, pp. 90-93, 2015.

[142] Nilsson, John-Olof and Rantakokko, Jouni and Handel, Peter and Skog, Isaac and Ohlsson, Martin and Hari, KVS, "Accurate indoor positioning of firefighters using dual foot-mounted inertial sensors and inter-agent ranging," in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, 2014.

APPENDIX A. WAPS SYNCHRONISATION/LOCALISATION SHEME

A.1 The c-Language code in OPNET modeller for WAP clock model

```
/**
    WAPS SYNCHRONISATION/LOCALISATION SHEME IN OPNET
    WAP_Localisation.prj
    Purpose: WAP clock model

    @author Halgurd S Maghdid
    @version 1.0 Januanry 2012
*/
char type[20];
op_ima_obj_attr_get (node_id, "c_type",&type);
if(strcmp(type,"c_Ref")==0)
{
tsf_from_clock= Ref_local_node_time();
}
else if(strcmp(type,"c_4S")==0)
{
tsf_from_clock= our_local_node_time_4S();
}
else if(strcmp(type,"c_all")==0)
{
tsf_from_clock= our_local_node_time();
}
...
static double Ref_local_node_time()
{
       double uu_r;
       double drift_time=0.0, time=op_sim_time();
       FIN(Ref_local_node_time());
       uu_r=op_sim_time()*op_sim_time()*((10.0*1e-
6/(1.0*365*24*3600)))*(1.0/2.0)+myabss(GetRand_R(3,-3)*1e-8);
              op_stat_write_t (drift_clock_Ref, uu_r*1000000000,op_sim_time());
       FRET(drift_time);
}
static double our_local_node_time()
{
double current_time=op_sim_time(),i;
double drift_time=0.0, time=old_time+0.1;
double intv=0.0;
FIN(our_local_node_time());
for ( i=old_time;i<=time;i=i+0.001)
{
       if( mymod(my_round((i+1)*1000.0,3),1)==0)
       {
           uu=myabss(S_WP(i+1)*randn(0,1));
       }
if( mymod(my_round((i+1)*100.0,3),1)==0 )
       {
```

```
    uu=myabss(S_FP(i+1)*(randn(0,1)));
       }
if( mymod(my_round((i+1)*50.0,3),1)==0)
       {
    uu=-1*myabss(S_WF(i+1)*(randn(0,1)));
       }
if( mymod(my_round((i+1)*10.0,3),1)==0)
       {
    uu=S_FF(i+1)*(randn(0,1));
       }
if(mymod(my_round(i+1,3),1)==0)
       {
    uu=S_RWF(i+1)*(randn(0,1));
       }
drift_time = olddrifttime +uu*10*pi;
olddrifttime = drift_time;
intv=intv+0.001;
}
old_time=time;
op_stat_write_t (drift_clock, drift_time*1000000000,op_sim_time());
time=drift_time;
FRET(time);
}
static double our_local_node_time_4S()
{
       double intv=0.0,i;
double drift_time=0.0, time=old_times+0.1;

FIN(our_local_node_time_4S());
for ( i=old_times;i<=time;i=i+0.001)
{
             if( mymod(my_round((i+1)*1000.0,3),1)==0)
       {
    uus=myabss(S_WP(i+1)*randn(0,1));
       }
if( mymod(my_round((i+1)*100.0,3),1)==0 )
       {
    uus=myabss(S_FP(i+1)*(randn(0,1)));
       }
if( mymod(my_round((i+1)*10.0,3),1)==0)
       {
    uus=S_FF(i+1)*(randn(0,1));
       }
if(mymod(my_round(i+1,3),1)==0)
       {
    uus=S_RWF(i+1)*(randn(0,1));
       }
drift_time = olddrifttimes +uus*10*pi;
olddrifttimes = drift_time;
intv=intv+0.001;
}
old_times=time;
op_stat_write_t (drift_clock_4S, drift_time*1000000000,op_sim_time());
time=drift_time;
FRET(time);}
```

A.2 The c-Language code in OPNET modeller for GNSS_Clck model

```
/**
    WAPS SYNCHRONISATION/LOCALISATION SHEME IN OPNET
    WAP_Localisation.prj
    Purpose: Generating timestampe based on GNSS-receiver clock

    @author Halgurd S Maghdid
    @version 1.0 Januanry 2012
*/

int i;
op_pk_nfd_get(pkptr , "mac_header", &header );
op_pk_nfd_get(pkptr , "body_frame", &pk);
get_locationsp(pk,location);

for( i=1.0;i<=4.0;i++)
{
if(location[2]==i)
	{
		GNSS_time[i-1]=tdrift[rand()%10][i-1];
		op_pk_fd_get(pk,0, &WAPs_clocksD[i-1]);
		op_stat_write (WAPs_clocksdrift[i-1], WAPs_clocksD[i-
1]*1e9+GNSS_time[i-1]);
	}

}
...
static void get_locationsp(Packet *pk,double location[])
{
	op_pk_fd_get(pk,1, &location[0]);
	op_pk_fd_get(pk,2, &location[1]);
	op_pk_fd_get(pk,3, &location[2]);
	op_pk_fd_get(pk,4, &location[3]);
}
```

A.3 The c-Language code in OPNET modeller for Locate SP/WAPs model

```
/**
    WAPS SYNCHRONISATION/LOCALISATION SHEME IN OPNET
    WAP_Localisation.prj
    Purpose: Locating SPs or WAPs by implementing Trilateration (using
             iterative LLS

    @author Halgurd S Maghdid
    @version 1.0 June 2012
*/

double dop=0.0;
Objid my_node_id = op_topo_parent(op_id_self ());
double ref[4][3]={{0}};
FILE *outfile;
double x_p,y_p;
double gu[3],rms=0.0,er_x=0.0, er_y=0.0;
if(count_beacon==40)
{
printf("beacon counts is %d \n",count_beacon);
op_ima_obj_attr_get (my_node_id, "x position",&x_p);
op_ima_obj_attr_get (my_node_id, "y position",&y_p);
op_stat_write (x_pos, x_p);
op_stat_write (y_pos, y_p);
flag=1;
for( i=0;i<4;i++)
       {
       printf("WAP_x value is %g \n",WAPs_x[i]);
       ref[i][0]=WAPs_x[i]; ref[i][1]=WAPs_y[i];
       pr[i]=pr[i]/10;
       op_stat_write (WAP_X[i], WAPs_x[i]);
       op_stat_write (WAP_Y[i], WAPs_y[i]);
       op_stat_write (TOF[i], pr[i]);
       }
sprintf(name,"%s.txt","WAPs");
printf("The name is %s\n",name);
outfile= fopen(name, "w");
if (outfile == NULL)
       {
              printf("Unable to open file.");
       }
else
       {
              for( i=0;i<4;i++)
              {
                     fprintf(outfile, "%lf\n%lf\n%lf\n",
WAPs_x[i],WAPs_y[i],pr[i]);
              }
       }
fclose(outfile);
WiFi_locat_Nodesp(gu,ref,pr,flag,old_x_est,old_y_est);
op_stat_write (x_est, gu[0]);
```

```
op_stat_write (y_est, gu[1]);
gu[2]=0;
dop=WiFi_DOPsp(gu,ref);
op_stat_write (hdop_value, dop);
printf("the dop value is %g\n",dop);
 er_x=((gu[0]-x_p));
 er_y=((gu[1]-y_p));
 rms=sqrt(0.5* ( pow(er_x,2)+pow(er_y,2) ) );
op_stat_write (rms_error_p,rms );
op_stat_write (WiFi_location_error,rms );
old_x_est=gu[0]; old_y_est=gu[1];
for( i=0;i<4;i++)
       {
       WAPs_x[i]=0; WAPs_y[i]=0; pr[i]=0;
       }
}
}// end of beacon count 200
...
void WiFi_locat_Nodesp(double gu[],double ref[][3],double mypr[],int
myflag,double old_x,double old_y)
{
       double Re_pinv[3][4] = { { 0 } };
       double rao[4] = { 1 }; //psueodorange
       double dl[3] = { 0 }; //delta for x, y, z
       double alpha[4][4] ={ { 1,2,1,1 },{ 3,3,1,1 },{ 5,7,1,1 },{ 9,2,1,1 } };
       double erro = 1.0;
       double drao[4] = { 0 };
       int count = 0,i,j,k;
       double sum = 0.0;

       gu[0]=50.0; gu[1]=50.0; gu[2]=0.0;

       for ( j = 0; j<4; j++)
              rao[j] = pow((pow((gu[0] - ref[j][0]), 2) + pow((gu[1] -
ref[j][1]), 2) + pow((gu[2] - ref[j][2]), 2)), 0.5);

       while (erro>0.01)
       {
              for ( j = 0; j < 4; j++)
              {
                     for ( k = 0;k<3;k++)
                     alpha[j][k] = (gu[k]-ref[j][k]) / (rao[j]);
              }
              for ( j = 0; j < 4; j++)
                     drao[j] = mypr[j] - rao[j]
              Wpinvsp(Re_pinv, alpha, 4, 3);

              for ( i = 0; i < 3; i++)
              {
                     sum = 0.0;
                     for ( j = 0; j < 4; j++)
                     {
                            sum=sum+Re_pinv[i][j]*drao[j];
                     }
                     dl[i] = sum;
```

```
            }

            for ( k=0;k<3;k++)
                  gu[k] = gu[k] + dl[k];
            erro = pow(dl[0], 2) + pow(dl[1], 2) + pow(dl[2], 2); // find
error
            for ( j = 0; j<4; j++)
                  rao[j] = pow((pow((gu[0] - ref[j][0]), 2) + pow((gu[1] -
ref[j][1]), 2) + pow((gu[2] - ref[j][2]), 2)), 0.5);
            count++;
      }

}
...
double WiFi_DOPsp(double gu[],double ref[][3])
      {
      double Re_pinv[4][4] = { { 0 } }; double T[4][4], f_alpha[4][4];
      double rao[4] = { 1 }; //psueodorange
      int i,k,j;
            double dop;
            double alpha[4][4] ={ { 1,2,1,1 },{ 3,3,1,1 },{ 5,7,1,1 },{
9,2,1,1 } };
      double drao[4] = { 0 };
      for ( j = 0; j<4; j++)
            rao[j] = pow((pow((gu[0] - ref[j][0]), 2) + pow((gu[1] -
ref[j][1]), 2) + pow((gu[2] - ref[j][2]), 2)), 0.5);

      for ( j = 0; j < 4; j++)
            {
                  for ( k = 0;k<3;k++)
                        alpha[j][k] = (gu[k]-ref[j][k])/ (rao[j]);// find
first 3 colums of alpha matrix
            }

      Wtranssp(T,alpha, 4,4);
      Wmultiplicationsp(f_alpha, T, alpha, 4, 4, 4, 4);
      Wpinvsp(Re_pinv, f_alpha, 4, 3);
       dop=sqrt(Re_pinv[0][0]+Re_pinv[1][1]);
      return dop;
}
```

A.4 The c-Language code in OPNET modeller for Move model

```
/**
    WAPS SYNCHRONISATION/LOCALISATION SHEME IN OPNET
    WAP_Localisation.prj
    Purpose: Mobility model to generating waypoints to make trajectory of SPs

    @author Halgurd S Maghdid
    @version 1.0 March 2012
*/
FILE *infiled;
double xp,yp;
double xx,yy;
int rest_count=0;
char line[100], name[20];
Objid my_node_id = op_topo_parent(op_id_self ());
Objid obj_type = op_id_to_type (my_node_id);
op_ima_obj_attr_get (my_node_id, "name", &name);
if(strcmp(name,"SP1")==0)
{
if(count_beacon==100)
{
mobility_count++;
infiled= fopen("Mobility_WAP.txt", "r");
       if(infiled==NULL)
       {
              printf("halo kaka cannot readddddddddddddddd\n");
       }
       else
       {
              while( fgets( line,100,infiled) && rest_count<(mobility_count*5))
              {
                     if( 1==sscanf(line,"%lf",&xp) )
                     {
                            printf("xp is: %lf ", xp);
                     }
                     fgets( line,100,infiled);
                     if( 1==sscanf(line,"%lf",&yp) )
                     {
                            printf("yp is: %lf ", yp);
                     }
                     rest_count++;
              }
       }
       fclose(infiled);
       xx=oldyp+xp; yy=oldzp-yp;
       if (obj_type == OPC_OBJTYPE_NODE_MOB||obj_type ==
OPC_OBJTYPE_SUBNET_MOB)
       {
              op_ima_obj_pos_set_geocentric (my_node_id, oldxp, xx, yy);
       }
                                  count_beacon=0;
}
}
```

APPENDIX B. BOTH THE SILS AND THE UNILS SCHEMES

B.1 The c-Language code in OPNET modeller for measuring Pseudorange between SPs

```c
/**
    SILS SHEME IN OPNET
    Blue_WAP.prj
    Purpose: Estimate pseudoranges between SPs in Piconet

    @author Halgurd S Maghdid
    @version 1.0 December 2013
*/
//During the sending POLL packet
double next_hop;
/* call next hop for each timeslot that has passed since the last hop */
for (i=0; i<(hop_slot - Last_hop_slot); i++)
        next_hop = (double) bt_next_freq_hop ();
op_stat_write (Hop_value_stat, next_hop);
change_frequencies(next_hop);
time_hop_selection=op_sim_time();
Last_hop_slot = Current_slot;

//During the received NULL packet
if (pk_stats[TYPE] == NULL_PKT)
{
double x_p,y_p,tsf=0.0;

        if(active[source_addr]!=1) // && flagchange!=1
                {
                op_stat_write (Nulls_received_stat, 1);
                op_pk_nfd_get(bt_pkt , "x_pos", &x_p );
                op_pk_nfd_get(bt_pkt , "y_pos", &y_p );
                x_posi[source_addr]=x_p;
                y_posi[source_addr]=y_p;
                op_stat_write (x_position[source_addr], x_p);
                op_stat_write (y_position[source_addr], y_p);
                tsf=( (op_sim_time()-time_hop_selection)-0.000035000)*1000000000;
                if(tsf>1000)
                        tsf=tsf-1000;
        pr_tsf[source_addr]=tsf;

                rcvd_power[source_addr]=op_td_get_dbl (bt_pkt,
OPC_TDA_RA_RCVD_POWER)*1000000;
                op_stat_write (power[source_addr], rcvd_power[source_addr]);
                sprintf(str,"receive   nulllllllll at %g \n",op_sim_time());
                active[source_addr]=1;
                }
                count_nulls++;

} //end of null received
```

B.2 The c-Language code in OPNET modeller for PRP algorithm

```
/**
    SILS SHEME IN OPNET
    Blue_WAP.prj
    Purpose: Implementing PRP algorim to reduce location erro which caused by
             DOP issue

    @author Halgurd S Maghdid
    @version 1.0 February 2014
*/
for(i=0;i<Number_of_slaves;i++)
{
        if(active[i]==1)
        {
                active[i]=0;
        locat_Node_wlls(gu,ref,cur_dis,Number_of_slaves,segma2,active,flag);
                gx[xy_index]=gu[0];
                gy[xy_index]=gu[1];
                er_x=0; er_y=0;
                er_x=myabs((gu[0]-xpp));
                er_y=myabs((gu[1]-ypp));
                impv_rms=sqrt(0.5* ( pow(er_x,2)+pow(er_y,2) ) );
                op_stat_write(location_error,impv_rms );
                in=1;
                xy_index++;
                }
        else{
                locat_Node_wlls(gu,ref,cur_dis,Number_of_slaves,segma2,active,fla
        g);
                gx[xy_index]=gu[0];
                gy[xy_index]=gu[1];
                er_x=0; er_y=0;
                er_x=myabs((gu[0]-xpp));
                er_y=myabs((gu[1]-ypp));
                impv_rms=sqrt(0.5* ( pow(er_x,2)+pow(er_y,2) ) );
                op_stat_write(location_error,impv_rms);
                xy_index++;
                }
        if(in==1)
                {
                active[i]=1;
                in=0;
                }
for(i=0;i<xy_index;i++)
        {
        meanx=meanx+gx[i];
        meany=meany+gy[i];
        }
meanx=meanx/xy_index;
meany=meany/xy_index;
xdif=(double *) malloc(xy_index*sizeof(double));
xdif_temp=(double *) malloc(xy_index*sizeof(double));
ydif=(double *) malloc(xy_index*sizeof(double));
```

```
weights=(double *) malloc(4*sizeof(double));
if(bluedop>=1.2)
	{
		weights[0]=0.1;weights[1]=0.1;weights[2]=0.8;
	}
else
	{
		weights[0]=0.9;weights[1]=0.09;weights[2]=0.01;
	}
tempo=(meanx-gx[0]);
for(i=0;i<xy_index;i++)
{
	xdif[i]=abs(meanx-gx[i])+abs(meany-gy[i]);
}
xy[0]=gx[0]; xy[1]=gy[0];
for(i=0;i<xy_index;i++)
{
	xdif_temp[i]=xdif[i];
}
for (i = 0 ; i < xy_index-1 ; i++)
{     // Trip-i begins
	for (j = 0 ; j < xy_index-1 ; j++)
	{
		if (xdif_temp[j] < xdif_temp[j+1])
		{ // Interchanging values
		t = xdif_temp[j];
			xdif_temp[j] = xdif_temp[j+1];
			xdif_temp[j+1] = t;
			t = gx[j];
			gx[j] = gx[j+1];
			gx[j+1] = t;
			t = gy[j];
			gy[j] = gy[j+1];
			gy[j+1] = t;
		}
		else
			continue ;
	}
}
for(i=0;i<3;i++)
{
	xf=xf+gx[i]*weights[i];
	yf=yf+gy[i]*weights[i];
}
gu[0]=xf;
gu[1]=yf;
er_x=0; er_y=0;
er_x=myabs((gu[0]-xpp));
er_y=myabs((gu[1]-ypp));
impv_rms=sqrt(0.5* ( pow(er_x,2)+pow(er_y,2) ) );
op_stat_write (impv_location_error[My_piconet_address], impv_rms);
```

B.3 The c-Language code in OPNET modeller for SMSR algorithm

```
/**
    SILS SHEME IN OPNET
    Blue_WAP.prj
    Purpose: Implementing SMSR algorim to reduce error of the estimated
             pseudoranges

    @author Halgurd S Maghdid
    @version 1.0 January 2014
*/
//Master-Side
if( name[0]=='M' && start_switching==1 && sumactv>4)
{
dist_fit = (double **)malloc(Number_of_slaves*sizeof(double *));
for( hg=0;hg<Number_of_slaves;hg++)
dist_fit[hg]=(double *) malloc(3*sizeof(double));
for( fit=0;fit<Number_of_slaves;fit++)
{
dist_fit[fit][0]=pr_old[fit];
}
for( kk=0;kk<Number_of_slaves;kk++)
{
if(p_active[kk]==1)
{
if(My_piconet_address==0)
sprintf(slavefile,"Slave_%d.txt",kk);
else
sprintf(slavefile,"Slave_%d%d.txt",My_piconet_address,kk);
rest_count=0;
infiled= fopen(slavefile, "r");
if(infiled==NULL)
{
printf("halo kaka  cannot readddddddddddddddd\n");
}
else
{
rest_count=-1;

while( fgets( line,100,infiled) && rest_count<kk)
      {
      if( 1==sscanf(line,"%lf",&prev_dist) )
            {
            //printf("prev_dist is: %lf ", prev_dist);
            }
      rest_count++;
      }
}
fclose(infiled);
dist_fit[kk][1]=prev_dist;
}
else
{
dist_fit[kk][1]=0;
```

```
}
}
sprintf(str,"Master_%d.txt",My_piconet_address);
infiledm= fopen(str, "r");
if(infiledm==NULL)
{
printf("halo kaka cannot  read\n");
}
else
{
rest_count=-1;
while( fgets( line,100,infiledm) && rest_count<(Number_of_slaves-1))
       {
       if(p_active[rest_count+1]==1)
       {
       if( 1==sscanf(line,"%lf",&prev_distm) )
              {
       //     printf("prev_dist is: %lf ", prev_distm);
              }
       rest_count++;
       dist_fit[rest_count][2]=prev_distm;
       }
       else
              {
              rest_count++;
              dist_fit[rest_count][2]=(50.0/1000000000)*(speed_light);
              }
       }
}
fclose(infiledm);
////////// fitting the set of distances ////////////////
for (fs=0;fs<Number_of_slaves;fs++)
{
if(p_active[fs]==1)
{
double Y[3]; double X[3]={1,2,3};
double XY[3];double X_2[3];      double re[3]; double a,b;
double xsum=0.0,ysum=0.0,xysum=0.0,x2sum=0.0;
int n=3,r;
double t,median;
Y[0]=dist_fit[fs][0]; Y[1]=dist_fit[fs][1]; Y[2]=dist_fit[fs][2];
op_stat_write (tsf_filtered[fs], (Y[0]/(speed_light))*1000000000);
op_stat_write (tsf_filtered[fs], (Y[1]/(speed_light))*1000000000);
op_stat_write (tsf_filtered[fs], (Y[2]/(speed_light))*1000000000);
XY[0]=Y[0]*X[0]; XY[1]=Y[1]*X[1]; XY[2]=Y[2]*X[2];
X_2[0]=X[0]*X[0]; X_2[1]=X[1]*X[1]; X_2[2]=X[2]*X[2];
for( su=0;su<3;su++)
       {
       xsum=xsum+X[su];
       ysum=ysum+Y[su];
       xysum=xysum+XY[su];
       x2sum=x2sum+X_2[su];
       }
       b=((x2sum*ysum)-(xsum*xysum)) / ((3*x2sum)-(pow(xsum,2)));
a=((3*xysum)-(xsum*ysum)) / ((3*x2sum)-(pow(xsum,2)));
```

```
for(r=0;r<3;r++)
	{
	re[r]=a*X[r]+b;
	}
for (i = 0 ; i < n-1 ; i++)
	{     // Trip-i begins
	for (j = 0 ; j < n-1 ; j++)
		{
		if (re[j] < re[j+1])
			{ // Interchanging values

			t = re[j];
			re[j] = re[j+1];
			re[j+1] = t;
			}
		else
			continue ;
		}
	} // sorting ends
// calculation of median
if ( n % 2 == 0)
	median = (re[n/2] + re[n/2+1])/2.0 ;
else
	median = re[n/2 + 1];
if(bluedop>1.2)
	cur_dis[fs]=(re[0]+re[1]+re[2])/3;//median;
else
	cur_dis[fs]=median;//(re[0]+re[1]+re[2])/3;
}

op_stat_write (tsf_filtered[fs], (cur_dis[fs]/(speed_light))*1000000000);
}

//Blue Location-sheet exchange/share to enhance location accuracy
//////////// share distances /////////////

sprintf(filename,"%s.txt",name);
outfiled= fopen(filename, "w");
if (outfiled == NULL)
	{
	printf("Unable to open file.");
	}
else
	{
		if(name[0]=='M' && start_switching==1)
		{
			start_switching=0;
			op_intrpt_schedule_self (op_sim_time () +0.080,
SETLOCATION);
			for(i=0;i<Number_of_slaves;i++)
				{
					p_active[i]=0;
					pr_old[i]=0;
				}
			}
```

```
                    else
                    {
                            for(i=0;i<Number_of_slaves;i++)
                                    {
                                    fprintf(outfiled, "%lf\n", pr_tsf[i]);
                                    }
                            if(name[0]=='M')
                            {
                                    start_switching=1;

                            }
                    }

        }
fclose(outfiled);
///////// Blue Switching to enhance the estimated pseudoranges //////////////
if(name[0]=='M')
{
op_stat_write (Master_node, 7);
}
else if(endsWith(name,"0")==1)
{
op_stat_write (Master_node, 0);
}
else if(endsWith(name,"1")==1)
{
op_stat_write (Master_node, 1);
}
else if(endsWith(name,"2")==1)
{
op_stat_write (Master_node, 2);
}
else if(endsWith(name,"3")==1)
{
op_stat_write (Master_node, 3);
}
else if(endsWith(name,"4")==1)
{
op_stat_write (Master_node, 4);
}
else if(endsWith(name,"5")==1)
{
op_stat_write (Master_node, 5);
}
else if(endsWith(name,"6")==1)
{
op_stat_write (Master_node, 6);
}
op_intrpt_schedule_self (op_sim_time ()+0.1075+(Number_of_slaves*2+6+Tsco) *
SLOT_TIME, MASTERMODE);

//////////          END END END END     ///////////////////

if(name[0]=='M')
{
```

```
        if(temp_active[0]==0 && respons[0]==-1)
        {
                Poll_interval=20;
                op_ima_obj_attr_set (my_node_id, "next_master", 1);
                timeis=op_sim_time();
                for(i=0; i<Number_of_slaves; i++)
                {
                        txtime=timeis+(double)((137 + 2 * i + Tsco)*SLOT_TIME)+
Offset;
                op_intrpt_schedule_self (txtime, POLL);
                }

        }
}
else if(endsWith(name,"6")==1)
{
        if(temp_active[6]==0 && respons[6]==-1)
        {
                Poll_interval=20;
                op_ima_obj_attr_set (my_node_id, "next_master", 0);
                timeis=op_sim_time();
                for(i=0; i<Number_of_slaves; i++)
                {
                        txtime=timeis+(double)((137 + 2 * i + Tsco)*SLOT_TIME)+
Offset;                 op_intrpt_schedule_self (txtime, POLL);
                }
        }
}


//Slave-Side
if (accept)
{

op_pk_nfd_get (bt_pkt, "ARQ", &arq);

if (arq)
s_Retx = OPC_FALSE;
/* get the slot modulo for the next timeslot */
next_slot_mod = (s_Current_slot + 1) % *s_Tsco;
s_Last_received_pk_type = pk_stats[TYPE];
Gb_overhead_tput[My_piconet] += pk_stats[1];
op_stat_write (Mac_packets_recv_stat, 1);
if (pk_stats[TYPE] == POLL_PKT)
{
                int next_master,my_address;
                        int check;

                op_ima_obj_attr_get(my_node_id, "Slave Address", &my_address);
                op_ima_obj_attr_get (my_node_id, "name", &name);
                op_ima_obj_attr_get (my_node_id, "Role_change", &check);
                op_pk_nfd_get(bt_pkt , "next_master", &next_master );
                sprintf(str,"Next master is %d and its slave sdd is
%d",next_master, my_address);
```

```
                if(next_master==7 && name[0]=='M')
                        {
                        op_ima_obj_attr_set(my_node_id, "next_master",0);
                        op_ima_obj_attr_set(my_node_id, "Role_change", 1);
                        op_intrpt_schedule_self (op_sim_time ()
+0.1025+(6+2*4)*0.000625, SLAVEMODE);
                        }
                else if(next_master==my_address && name[0]!='M' && check!=3)
                        {
        /// Switch back to Original Master   ////////////////
        if(endsWith(name,"6")==1)
                op_ima_obj_attr_set(my_node_id, "next_master",7);
        else
                op_ima_obj_attr_set(my_node_id, "next_master",(next_master+1));
                op_ima_obj_attr_set(my_node_id, "Role_change", 1);
                op_intrpt_schedule_self (op_sim_time () +0.1025, SLAVEMODE)}

op_stat_write (Poll_pkt_received_stat, 1);

}
```

B.4 The c-Language code in OPNET modeller for Fusion algorithm (using Kalman Filter)

```
/**
    UNILS SHEME IN OPNET
    Blue_WAP.prj
    Purpose: Implementing the fusion algorim using Kalman filter

    @author Halgurd S Maghdid
    @version 1.0 November 2014
*/
/////////     Inertial-Sensor Model     ////////////////////

double act[7];
int i,j;
double x,y,t, *delta_dis_angle;
double **A;
double ** B;
double **G;
double Kss = (1e-2);
double Kst = (1e-6);
double Ktt = (1e-4); //Need to be multiplied by a factor
double **PA, **AT, **PAP, **temp;
double ** BG;
double **BT;double **BGB;
double **Q;
double dr_x_co, dr_y_co, dr_heading_co, xr,yr, xb, yb, rm, dist, R, S,*H,
*del_q, *K;
double *PH, *KS, **KSK;
double er_x,er_y,impv_rms;
int jj;
double xpp=0,ypp=0,zpp=0;
heading_drift=heading_drift+(0.65*MyPI/180);
heading=heading+heading_drift+op_dist_uniform(4.0)*MyPI/180;
delta_dis_angle=(double *) malloc(2*sizeof(double));
if(mobility_count==1)
{
       q_pre[0]=dr_x;
       q_pre[1]=dr_y;
       q_pre[2]=heading;
       op_stat_write (x_estimation_dr,dr_x);
       op_stat_write (y_estimation_dr,dr_y);
}
else
{
///////////// integrate TOA & DR using Kalman filter ///////////////////////
///// Dead-reckonging Algorithm ////////////////////////////
x=dr_x+(0.6)*cos(heading);
y=dr_y-(0.6)*sin(heading);
dr_x=x;
dr_y=y;
op_stat_write (x_estimation_dr, x);
op_stat_write (y_estimation_dr, y);
```

```
delta_dis_angle[0]=0.6;
delta_dis_angle[1]=(heading-last_heading);

x = q_pre[0]; // previuos x_axis value
y = q_pre[1]; // previuos y_axis value
t = q_pre[2]; // heading
q_pre[0] = x +  delta_dis_angle[0]*cos(t);
q_pre[1] = y -  delta_dis_angle[0]*sin(t);
q_pre[2] = t + delta_dis_angle[1];
A = (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
A[i]=(double *) malloc(3*sizeof(double));
A[0][0]=1.0;        A[0][1]=0.0;        A[0][2]=-1*delta_dis_angle[0]*sin(t);
A[1][0]=0.0;        A[1][1]=1.0;        A[1][2]= delta_dis_angle[0]*cos(t);
A[2][0]=0.0;        A[2][1]=0.0;        A[2][2]=1.0;
B= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
B[i]=(double *) malloc(3*sizeof(double));
B[0][0]=cos(t); B[0][1]=0.0;
B[1][0]=sin(t);  B[1][1]=0.0;
B[2][0]=0;        B[2][1]=1;
G= (double **)malloc(2*sizeof(double *));
for( i=0;i<2;i++)
G[i]=(double *) malloc(2*sizeof(double));
G[0][0]=1e-5; G[0][1]=0;
G[1][0]=0; G[1][1]=1e-5;

PA= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
      PA[i]=(double *) malloc(3*sizeof(double));
AT= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
      AT[i]=(double *) malloc(3*sizeof(double));
PAP= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
      PAP[i]=(double *) malloc(3*sizeof(double));
temp= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
      temp[i]=(double *) malloc(3*sizeof(double));
for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
            temp[i][j]=P_pre[i][j];
}
multiplication3by3(PA,  A, temp,3, 3, 3, 3);
trans3by3(AT,A,3,3);
multiplication3by3(PAP, PA, AT, 3, 3, 3, 3);
BG= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
BG[i]=(double *) malloc(3*sizeof(double));
BG[0][0]=B[0][0]*G[0][0]+B[0][1]*G[1][0];
      BG[0][1]=B[0][0]*G[0][1]+B[0][1]*G[1][1];
BG[1][0]=B[1][0]*G[0][0]+B[1][1]*G[1][0];
      BG[1][1]=B[1][0]*G[0][1]+B[1][1]*G[1][1];
```

```
BG[2][0]=B[2][0]*G[0][0]+B[2][1]*G[1][0];
       BG[2][1]=B[2][0]*G[0][1]+B[2][1]*G[1][1];
BT= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
BT[i]=(double *) malloc(3*sizeof(double));
BT[0][0]=B[0][0];   BT[0][1]=B[1][0];         BT[0][2]=B[2][0];
BT[1][0]=B[0][1];   BT[1][1]=B[1][1];         BT[1][2]=B[2][1];
BGB= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
BGB[i]=(double *) malloc(3*sizeof(double));
BGB[0][0]=BG[0][0]*BT[0][0]+BG[0][1]*BT[1][0];
       BGB[0][1]=BG[0][0]*BT[0][1]+BG[0][1]*BT[1][1];
       BGB[0][2]=BG[0][0]*BT[0][2]+BG[0][1]*BT[1][2];
BGB[1][0]=BG[1][0]*BT[0][0]+BG[1][1]*BT[1][0];
       BGB[1][1]=BG[1][0]*BT[0][1]+BG[1][1]*BT[1][1];
       BGB[1][2]=BG[1][0]*BT[0][2]+BG[1][1]*BT[1][2];
BGB[2][0]=BG[2][0]*BT[0][0]+BG[2][1]*BT[1][0];
       BGB[2][1]=BG[2][0]*BT[0][1]+BG[2][1]*BT[1][1];
       BGB[2][2]=BG[2][0]*BT[0][2]+BG[2][1]*BT[1][2];
Q= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
Q[i]=(double *) malloc(3*sizeof(double));
Q[0][0]=Kss*Habs(delta_dis_angle[0]*cos(t)); Q[0][1]=0.0; Q[0][2]=0.0;
Q[1][0]=0.0; Q[1][1]=Kss*Habs(delta_dis_angle[0]*sin(t)); Q[1][2]=0.0;
Q[2][0]=0.0; Q[2][1]=0.0; Q[2][2]= Kst*Habs(delta_dis_angle[0]) +
Ktt*Habs(delta_dis_angle[1]);


P_pre[0][0]=PAP[0][0]+BGB[0][0]+Q[0][0];
P_pre[0][1]=PAP[0][1]+BGB[0][1]+Q[0][1];
P_pre[0][2]=PAP[0][2]+BGB[0][2]+Q[0][2];
P_pre[1][0]=PAP[1][0]+BGB[1][0]+Q[1][0];
P_pre[1][1]=PAP[1][1]+BGB[1][1]+Q[1][1];
P_pre[1][2]=PAP[1][2]+BGB[1][2]+Q[1][2];
P_pre[2][0]=PAP[2][0]+BGB[2][0]+Q[2][0];
P_pre[2][1]=PAP[2][1]+BGB[2][1]+Q[2][1];
P_pre[2][2]=PAP[2][2]+BGB[2][2]+Q[2][2];

PH=(double *) malloc(3*sizeof(double));
KS=(double *) malloc(3*sizeof(double));
K=(double *) malloc(3*sizeof(double));
H=(double *) malloc(3*sizeof(double));
del_q=(double *) malloc(3*sizeof(double));

KSK= (double **)malloc(3*sizeof(double *));
for( i=0;i<3;i++)
KSK[i]=(double *) malloc(3*sizeof(double));

if (flagchange==1)
{
for( jj=0;jj<Number_of_slaves;jj++)
act[jj]=p_active[jj];
}
else
{
```

```
for( jj=0;jj<Number_of_slaves;jj++)
act[jj]=active[jj];
}
for(i=0;i<Number_of_slaves;i++) //Number_of_slaves
{
	xr = q_pre[0];
	yr = q_pre[1];
	if(act[i]==1)
	{
		xb = x_posi[i];
		yb = y_posi[i];
		rm = pr_integration[i]; //range measurement
		dist = sqrt( (xr-xb)*(xr-xb)+(yr-yb)*(yr-yb) );




		H[0]=(xr-xb)/sqrt((xr-xb)*(xr-xb)+(yr-yb)*(yr-yb)); H[1]= (yr-
yb)/sqrt((xr-xb)*(xr-xb)+(yr-yb)*(yr-yb)); H[2]=0;
		R =MyPI;
		//MEASUREMENT UPDATE STEP

		PH[0]=H[0]*P_pre[0][0]+H[1]*P_pre[0][1]+H[2]*P_pre[0][2];
		PH[1]=H[0]*P_pre[1][0]+H[1]*P_pre[1][1]+H[2]*P_pre[1][2];
		PH[2]=H[0]*P_pre[2][0]+H[1]*P_pre[2][1]+H[2]*P_pre[2][2];
				S=(PH[0]*H[0]+PH[1]*H[1]+PH[2]*H[2]+R);

		K[0]=PH[0]/S; K[1]=PH[1]/S; K[2]=PH[2]/S;

del_q [0]= K[0]*(rm-dist)*10; del_q [1]= K[1]*(rm-dist)*10; del_q [2]=
K[2]*(rm-dist);
		q_pre[0]=q_pre[0]+del_q[0];
		q_pre[1]=q_pre[1]+del_q[1];
		q_pre[2]=q_pre[2]+del_q[2];

		KS[0]=K[0]*S; KS[1]=K[1]*S; KS[2]=K[2]*S;

		KSK[0][0]=KS[0]*K[0]; KSK[0][1]=KS[0]*K[1]; KSK[0][2]=KS[0]*K[2];
		KSK[1][0]=KS[1]*K[0]; KSK[1][1]=KS[1]*K[1]; KSK[1][2]=KS[1]*K[2];
		KSK[2][0]=KS[2]*K[0]; KSK[2][1]=KS[2]*K[1]; KSK[2][2]=KS[2]*K[2];

		P_pre[0][0]=P_pre[0][0]-KSK[0][0];  P_pre[0][1]=P_pre[0][1]-
KSK[0][1];  P_pre[0][2]=P_pre[0][2]-KSK[0][2];
		P_pre[1][0]=P_pre[1][0]-KSK[1][0];  P_pre[1][1]=P_pre[1][1]-
KSK[1][1]; P_pre[1][2]=P_pre[1][2]-KSK[1][2];
		P_pre[2][0]=P_pre[2][0]-KSK[2][0];  P_pre[2][1]=P_pre[2][1]-
KSK[2][1];  P_pre[2][2]=P_pre[2][2]-KSK[2][2];
		}
}
dr_x_co=q_pre[0];
dr_y_co=q_pre[1];
op_stat_write (x_estimation_dr_co,dr_x_co);
op_stat_write (y_estimation_dr_co, dr_y_co);
dr_heading_co=q_pre[2]*180.0/MyPI;
if (dr_heading_co<0)
```

```
{
       dr_heading_co=dr_heading_co+360;
}
op_stat_write(sensor_gyro_co,dr_heading_co);
my_node_id = op_topo_parent(op_id_self ());
op_ima_obj_attr_get (my_node_id, "x position", &old_xpp);
op_ima_obj_attr_get (my_node_id, "y position", &old_ypp);
er_x=myabs((dr_x_co-old_xpp));
er_y=myabs((dr_y_co-old_ypp));
impv_rms=sqrt(0.5* ( pow(er_x,2)+pow(er_y,2) ) );
op_stat_write(rms_p_error_filtered,impv_rms );
free(BGB); free(Q); free(PAP);  free(AT); free(H); free(K); free(KS);
free(KSK); free(del_q); free(HP); free(temp);
}

last_heading=heading;
heading=heading*180/MyPI;
if (heading<0)
{
       heading=heading+360;
}
op_stat_write(sensor_gyro,heading);
////////// end of Kalman filter //////////////////////////////

for(i=0;i<7;i++)
{
       p_active[i]=0;
}

////////               END Sensor         ///////////////
```

B.5 The SILS implementation in multi-Piconets scenario

In the following experiments, the SILS is tested in multi-Piconets scenario. Where 20 Piconets (each Piconet has 8 connected-SPs) start to run SILS in the outside of the buildings, as shown in Figure B-1.

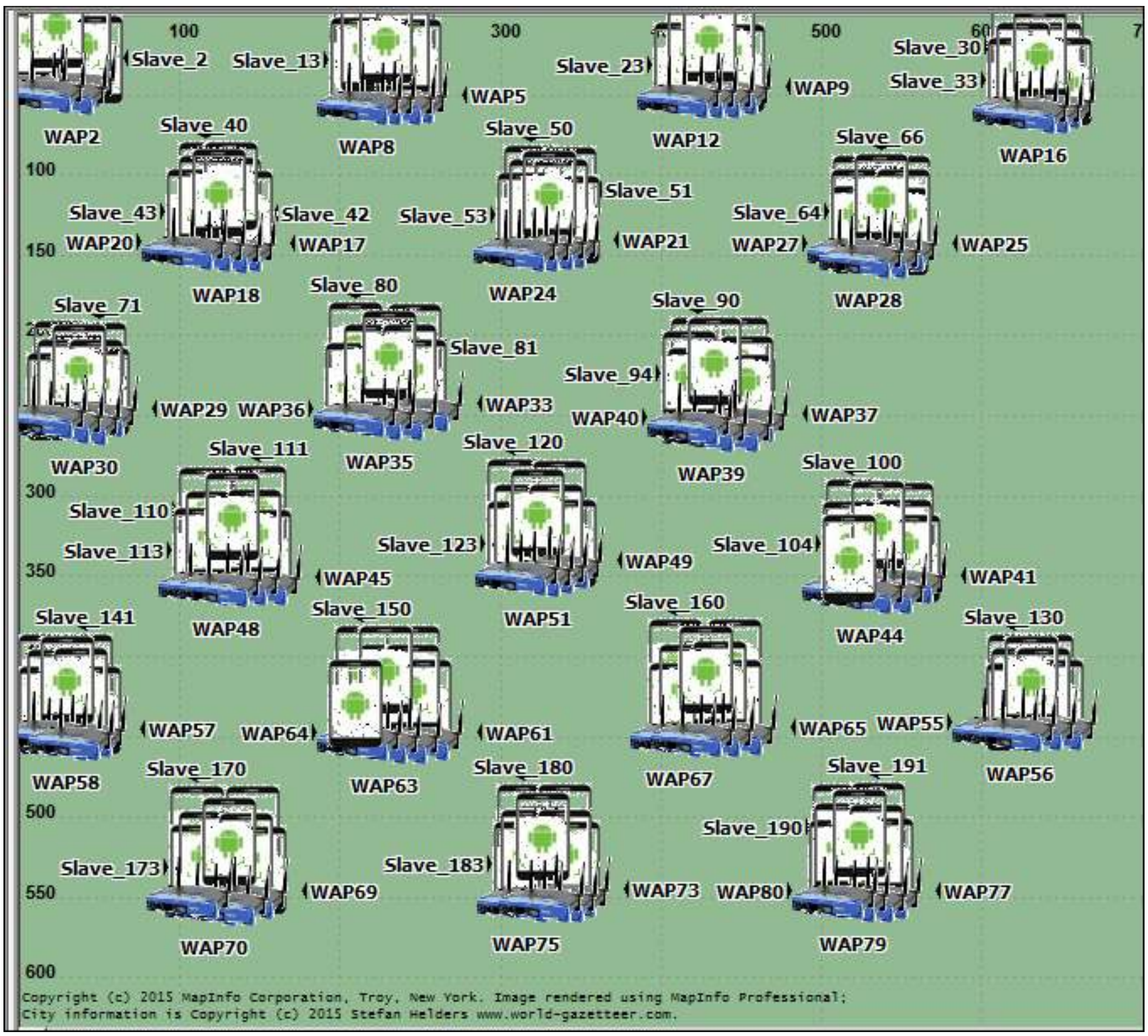


Figure B-1: Muli-Piconets when few numbers of SPs are moved from outdoors to indoors.

In each Piconet, one of the SPs moves indoors through light indoors area (signals are crossing 1 wall from the outside) and deep indoors (signals are crossing 3 walls deep inside the building). The movement of the SPs is illustrated in Figure B-2 by trajectory line (white lines.

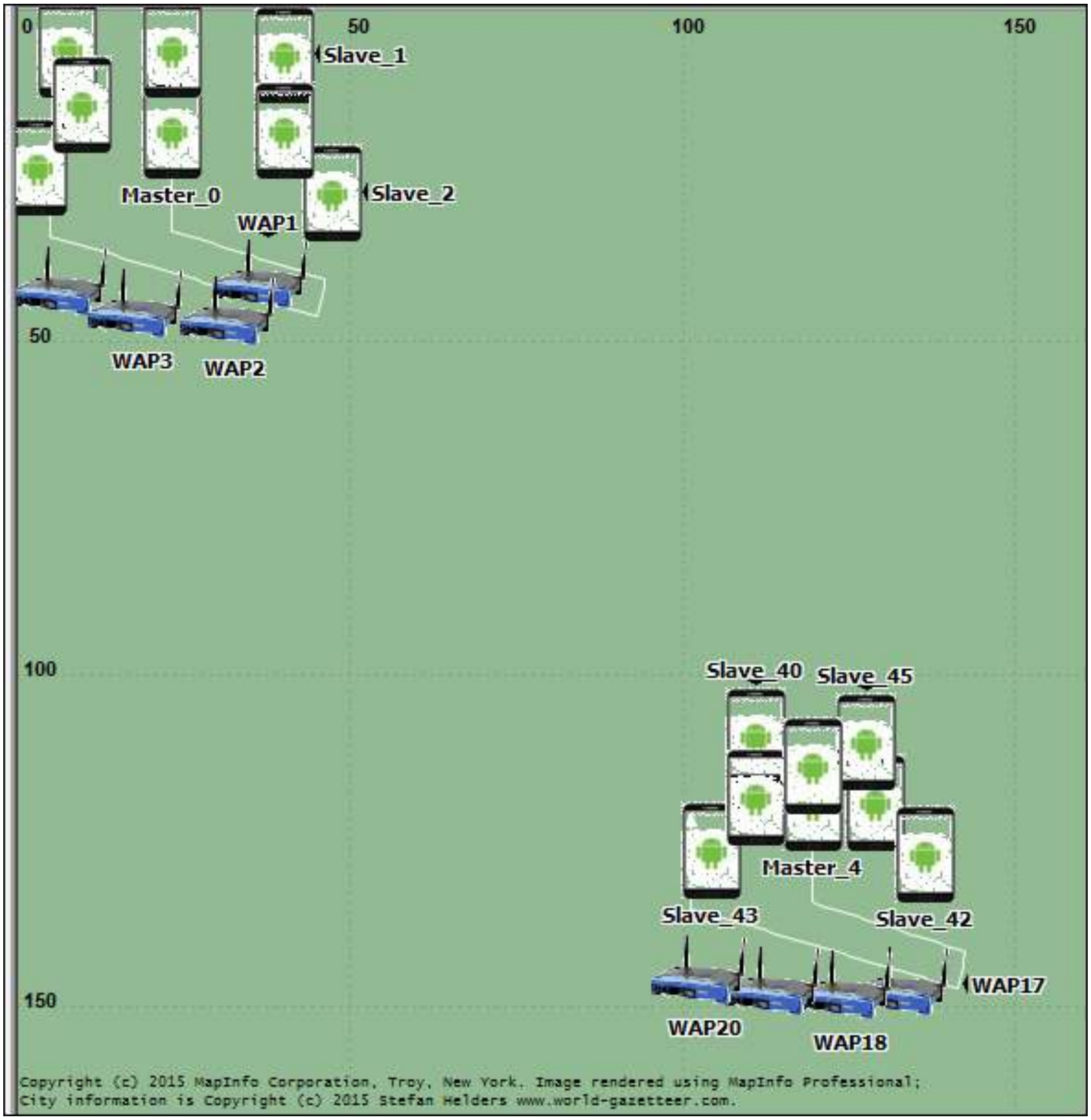


Figure B-2: The movements of SPs for two Piconets in the vicinity.

Figure B-3 shows the SNR obtained by the SPs that is labelled “master” as it travels from outdoors to deep indoors and comes back. Note: we put the SNR of only five Piconets so that the figure displays the result more clearly instead of the twenty Piconets.

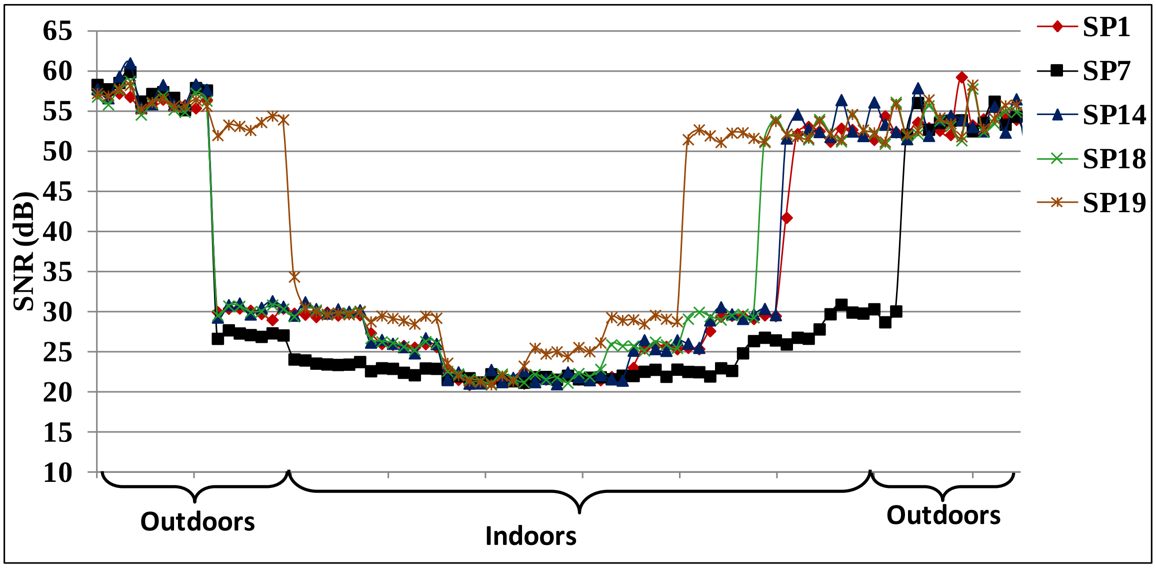


Figure B-3: SNR measurements from outdoor to indoor.

In addition, Figure B-4 shows the number of SPs connected in the five Piconets as the "master" node signals passed extra walls.

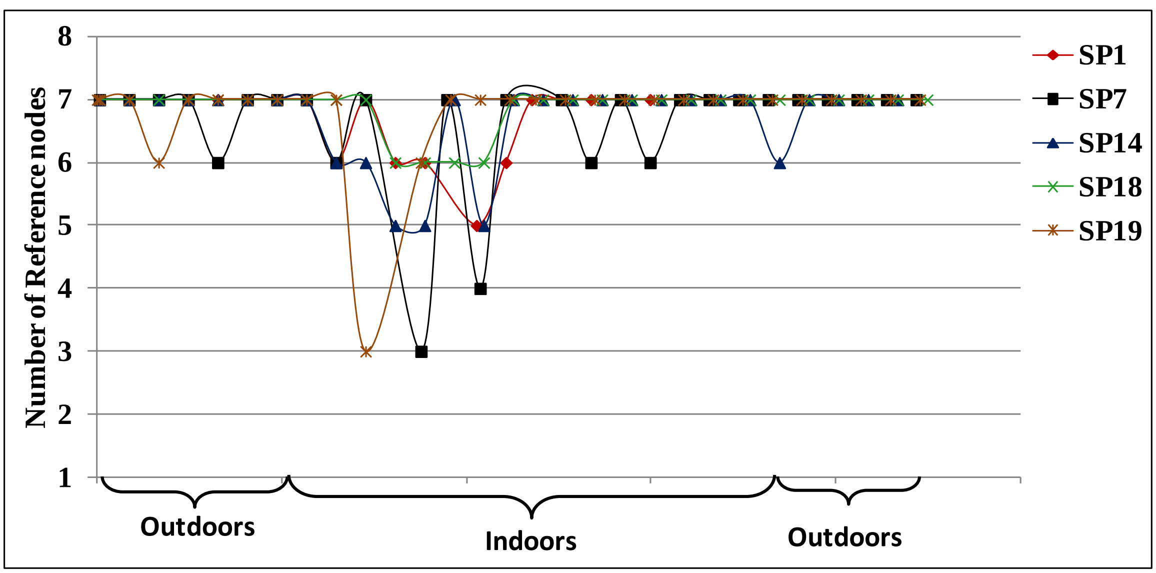


Figure B-4: Number of SPs connected in the Piconets.

Figure B-5 shows the obtained location error without adopting the SMSR algorithm. That is, by only using basic hop synchronisation to estimate pseudoranges between SPs.

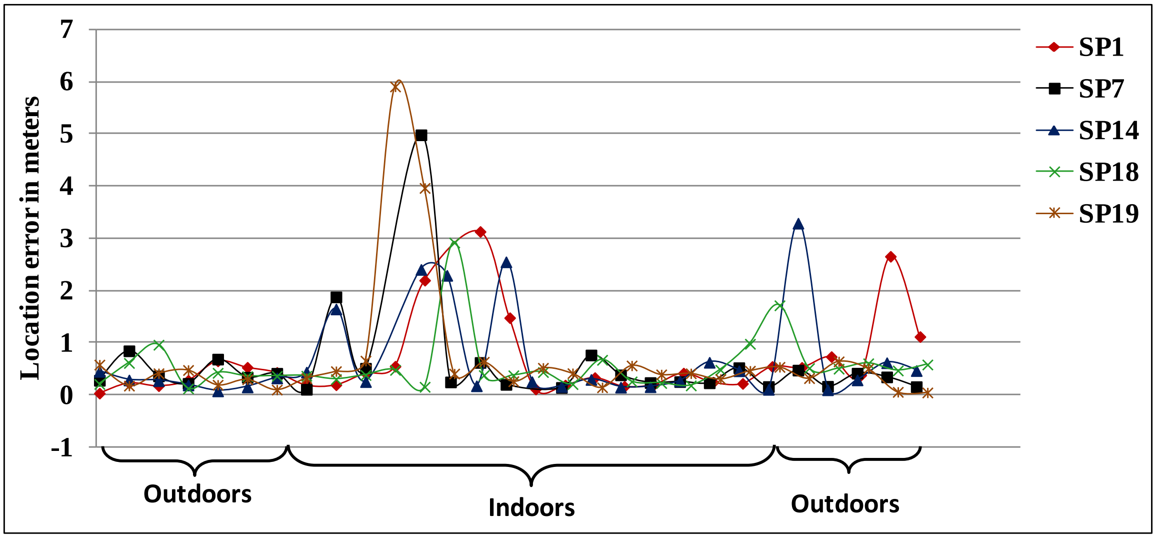


Figure B-5: Location errors from outdoors to indoors without SMSR.

Figure B-6 shows the SPm location error when the SILS works in the full-functionality mode with using beacon signals WAPs. It also demonstrates how the error of 3-walls deep indoors trial has now been improved by including the WAPs as reference positions.

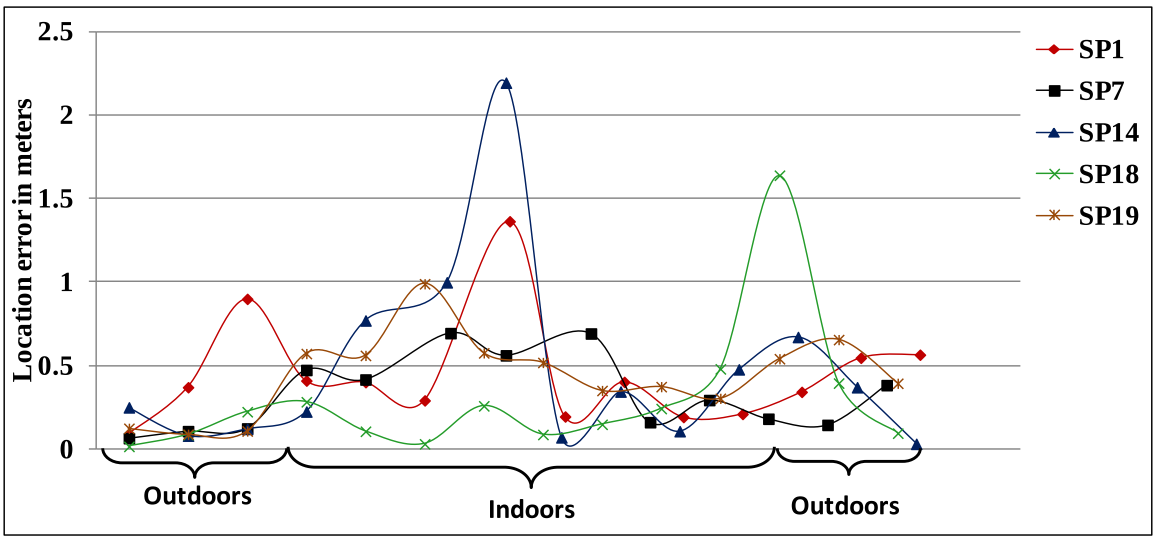


Figure B-6: Location error for full SILS functionality.

Moreover, Table B-1 displays statistics on the obtained location accuracy (based on RMSE) for all 20 Piconets. As it can be observed that when the SILS works in full functionality mode (SMSR+PRP), it provides accurate positioning than when SILS work in development mode.

Table B-1: Statistics on the obtained Location accuracy via SILS.

Master SPs in all Piconets	**STDEV**		**Mean**	
	SILS in Development	**SILS in Full-Functionalities**	**SILS in Development**	**SILS in Full-Functionalities**
SP1	0.392	0.356	0.497	0.431
SP2	0.597	0.306	0.593	0.377
SP3	0.953	0.192	0.784	0.326
SP4	1.020	0.367	0.967	0.404
SP5	2.900	0.167	1.097	0.287
SP6	0.172	0.036	0.361	0.233
SP7	0.393	0.317	0.411	0.328
SP8	0.232	0.161	0.334	0.255
SP9	0.256	0.147	0.321	0.254
SP10	0.547	0.272	0.485	0.399
SP11	0.992	0.569	0.649	0.598
SP12	0.486	0.289	0.482	0.443
SP13	0.396	0.492	0.469	0.468
SP14	0.859	0.216	0.717	0.270
SP15	1.060	0.355	0.800	0.343
SP16	0.247	0.342	0.334	0.316
SP17	0.259	0.348	0.362	0.316
SP18	0.461	0.175	0.616	0.239
SP19	0.385	0.435	0.489	0.386
SP20	0.973	0.225	0.649	0.356
MIN	**0.172**	**0.036**	**0.321**	**0.233**
MAX	**2.900**	**0.569**	**1.097**	**0.598**
Mean	**0.679**	**0.288**	**0.571**	**0.351**
STDEV	**0.604**	**0.128**	**0.217**	**0.091**

APPENDIX C.ANDROID-BASED SPS IMPLEMENTATIONS

C.1 The Java code for Android-based SPs of this constructing Piconet

```
/**
    SILS SHEME on Android-based SPs
    Blue_Network
    Purpose: Constructing network via Bluetooth technology

    @author Halgurd S Maghdid
    @version 1.0 November 2013
*/
```

BluetoothApp.java

```
package com.example.bluelocation;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Date;
import java.util.Iterator;
import java.util.UUID;
import com.example.bluelocation.BluetoothLocationService.ConnectedThread;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.GoogleMap.OnMyLocationChangeListener;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.Circle;
import com.google.android.gms.maps.model.CircleOptions;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.Dialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.location.Address;
import android.location.Criteria;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
```

```java
import android.os.Message;
import android.support.v4.app.FragmentActivity;
import android.util.FloatMath;
import android.util.Log;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.View.OnClickListener;
import android.view.inputmethod.EditorInfo;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
public class BluetoothApp extends FragmentActivity implements LocationListener
{
int doneIt;
File root;
File textFile;
FileWriter textWriter;
// GPS Location & Google Map
GoogleMap googleMap;
LocationManager mlocation = null;
double[] remote_device_lat;
double[] remote_device_acc;
double[] remote_device_log;
int[] remote_device_rssi;
double[] remote_device_prev_dis;
String[] remote_device;
LatLng[] latLng;
// Debugging
private static final String TAG = "BluetoothApp";
private static final boolean D = true;
// Message types sent from the BluetoothLocationService Handler
public static final int MESSAGE_STATE_CHANGE = 1;
public static final int MESSAGE_READ = 2;
public static final int MESSAGE_WRITE = 3;
public static final int MESSAGE_DEVICE_NAME = 4;     int count_devices;
public static final int MESSAGE_TOAST = 5;
// Key names received from the BluetoothLocationService Handler
public static final String DEVICE_NAME = "device_name";
public static final String TOAST = "toast";
// Intent request codes
private static final int REQUEST_CONNECT_DEVICE = 1;
private static final int REQUEST_ENABLE_BT = 2;
// Layout Views
private TextView mTitle;
private ListView mConversationView;
 // private EditText mOutEditText;
 // Name of the connected device
private String mConnectedDeviceName = null;
```

```java
// Array adapter for the conversation thread
private ArrayAdapter<String> mConversationArrayAdapter;
// String buffer for outgoing messages
private StringBuffer mOutStringBuffer;
// Local Bluetooth adapter
private BluetoothAdapter mBluetoothAdapter = null;
// Member object for the location services
private BluetoothLocationService mChatService = null;
TextView tvLocation;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if(D) Log.e(TAG, "+++ ON CREATE +++");
// Set up the window layout
requestWindowFeature(Window.FEATURE_CUSTOM_TITLE);
setContentView(R.layout.mainblue);
getWindow().setFeatureInt(Window.FEATURE_CUSTOM_TITLE, R.layout.custom_title);
/// for location
mlocation = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
// Set up the custom title
mTitle = (TextView) findViewById(R.id.title_left_text);
mTitle.setText(R.string.app_name);
mTitle = (TextView) findViewById(R.id.title_right_text);
tvLocation = (TextView) findViewById(R.id.tv_location);
String name="kaka",address="baba";
name=getDeviceName();
address=getDeviceAddr();
tvLocation.setText(name+"\t"+address);
remote_device_lat=new double[7];
remote_device_log=new double[7];
remote_device_acc=new double[7];
remote_device_prev_dis=new double[7];
remote_device=new String[7];
remote_device_rssi=new int[7];
latLng=new LatLng[7];
// Get local Bluetooth adapter
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
// If the adapter is null, then Bluetooth is not supported
if (mBluetoothAdapter == null) {
    Toast.makeText(this, "Bluetooth is not available",
Toast.LENGTH_LONG).show();
        finish();
        return;
    }
}
@Override
public void onStart() {
    super.onStart();
    if(D) Log.e(TAG, "++ ON START ++");

    // If BT is not on, request that it be enabled.
// setupChat() will then be called during onActivityResult
   if (!mBluetoothAdapter.isEnabled()) {
        Intent enableIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
```

```
        startActivityForResult(enableIntent, REQUEST_ENABLE_BT);
    // Otherwise, setup the chat session
    } else {
        if (mChatService == null)
        {
                    setupChat();
        }

    }

}

@Override
public synchronized void onResume() {
    super.onResume();
    if(D) Log.e(TAG, "+ ON RESUME +");
// Performing this check in onResume() covers the case in which BT was
// not enabled during onStart(), so we were paused to enable it...
// onResume() will be called when ACTION_REQUEST_ENABLE activity returns.
if (mChatService != null) {
    // Only if the state is STATE_NONE, do we know that we haven't started
already
if (mChatService.getState() == BluetoothLocationService.STATE_NONE) {
  // Start the Bluetooth chat services
          mChatService.start();
        }
    }
}
private void setupChat() {
    Log.d(TAG, "setupChat()");

// Initialize the array adapter for the conversation thread
mConversationArrayAdapter = new ArrayAdapter<String>(this, R.layout.message);
mConversationView = (ListView) findViewById(R.id.in);
mConversationView.setAdapter(mConversationArrayAdapter);
// Initialize the BluetoothChatService to perform bluetooth connections
mChatService = new BluetoothLocationService(this, mHandler);
// Initialize the buffer for outgoing messages
mOutStringBuffer = new StringBuffer("");
    int status =
GooglePlayServicesUtil.isGooglePlayServicesAvailable(getBaseContext());
 // Showing status
if(status!=ConnectionResult.SUCCESS){ // Google Play Services are not
available
      int requestCode = 10;
    Dialog dialog = GooglePlayServicesUtil.getErrorDialog(status, this,
requestCode);
    dialog.show();

}
else { // Google Play Services are available
// Getting reference to the SupportMapFragment of activity_main.xml
SupportMapFragment fm = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.mymap);
// Getting GoogleMap object from the fragment
```

```
googleMap = fm.getMap();
googleMap.setMyLocationEnabled(true);
googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.NO_REQUIREMENT);
criteria.setPowerRequirement(Criteria.NO_REQUIREMENT);
String best = mlocation.getBestProvider(criteria, true);
mlocation.requestLocationUpdates(best, 1000, 0, BluetoothApp.this);
    }
}

@Override
public void onDestroy() {
    super.onDestroy();
    // Stop the Bluetooth chat services
if (mChatService != null) mChatService.stop();
if(D) Log.e(TAG, "--- ON DESTROY ---");
}
private void ensureDiscoverable() {
    if(D) Log.d(TAG, "ensure discoverable");
    if (mBluetoothAdapter.getScanMode() !=
        BluetoothAdapter.SCAN_MODE_CONNECTABLE_DISCOVERABLE) {
        Intent discoverableIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);

discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION,
2000);
        startActivity(discoverableIntent);
    }
}
public void sendMessage(String message) {
// Check that there's actually something to send
if (message.length() > 0) {
    // Get the message bytes and tell the BluetoothChatService to write
byte[] send = message.getBytes();
mChatService.write(send);

// Reset out string buffer to zero and clear the edit text field
mOutStringBuffer.setLength(0);
//mOutEditText.setText(mOutStringBuffer);
    }
}
// The action listener for the EditText widget, to listen for the return key
private TextView.OnEditorActionListener mWriteListener =
    new TextView.OnEditorActionListener() {
    public boolean onEditorAction(TextView view, int actionId, KeyEvent event)
{
        // If the action is a key-up event on the return key, send the message
if (actionId == EditorInfo.IME_NULL && event.getAction() ==
KeyEvent.ACTION_UP) {
    String message = view.getText().toString();
    sendMessage(message);
}
if(D) Log.i(TAG, "END onEditorAction");
            return true;
```

```
        }
    };
boolean connected;
int countrssi;
    // The Handler that gets information back from the BluetoothChatService
private final Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        switch (msg.what)
        {
        case MESSAGE_STATE_CHANGE:
            if(D) Log.i(TAG, "MESSAGE_STATE_CHANGE: " + msg.arg1);
switch (msg.arg1)
{
    case BluetoothLocationService.STATE_CONNECTED:
       if(count_devices==1)
       {
              mTitle.setText(R.string.title_connected_to);
              mTitle.append(mConnectedDeviceName+";");
       connected=true;
}
else
{
       mTitle.append(mConnectedDeviceName+";");
       connected=true;

}
    break;
        case BluetoothLocationService.STATE_CONNECTING:
            mTitle.setText(R.string.title_connecting);
            break;
        case BluetoothLocationService.STATE_LISTEN:
        case BluetoothLocationService.STATE_NONE:
            mTitle.setText(R.string.title_not_connected);
            break;
    }
    break;
case MESSAGE_WRITE:
    byte[] writeBuf = (byte[]) msg.obj;
    // construct a string from the buffer
String writeMessage = new String(writeBuf);
String[] tokenMsgw=writeMessage.split("#");
mConversationArrayAdapter.add("Me:  " + tokenMsgw[1]);
    break;
case MESSAGE_READ:
    byte[] readBuf = (byte[]) msg.obj;
    // construct a string from the valid bytes in the buffer
String readMessage = new String(readBuf, 0, msg.arg1);
if (readMessage.length() > 0)
{
       countrssi++;
       countlocation++;
       String[] tokenMsg=readMessage.split("#");
String[] dlatlong=tokenMsg[1].split(",");
```

```
    for (int i=0;i<BluetoothLocationService.numberConnection;i++)
    {
      if(i==Integer.parseInt(dlatlong[0]))
      {
            remote_device[i]=tokenMsg[0];
            remote_device_lat[i]=Double.parseDouble(dlatlong[1]);
            remote_device_log[i]=Double.parseDouble(dlatlong[2]);
            remote_device_acc[i]=Double.parseDouble(dlatlong[3]);
            if(remote_device[i]!=null)
                  remote_device_rssi[i]=getRSSI(remote_device[i]);

      }
      latLng[i] = new LatLng(remote_device_lat[i], remote_device_log[i]);
    }

countloc++;
if(countloc==1)
{
      Date dt = new Date();
    int hours = dt.getHours();
    int minutes = dt.getMinutes();
    int seconds = dt.getSeconds();
    String curTime = hours + "_"+minutes+ "_"+ seconds+"_Blue.txt";
root = Environment.getExternalStorageDirectory();
textFile = new File(root, "/"+curTime );
}

tvLocation.setText(countrssi+";");
if (countrssi%30==0)
{
      googleMap.clear();
            try

            textWriter = new FileWriter(textFile, true);
            BufferedWriter out = new BufferedWriter(textWriter);
            if(BluetoothLocationService.numberConnection>2)
      {
                  for (int
i=0;i<BluetoothLocationService.numberConnection;i++)
           {

      tvLocation.append(remote_device[i]+";"+remote_device_rssi[i]+";"+Predict
_Dis(remote_device_rssi[i],remote_device_prev_dis[i])+"\n");

      out.write(remote_device[i]+";"+remote_device_rssi[i]+";"+remote_device_l
at[i]+";"+remote_device_log[i]+";");

            double[]
latlong=MyTrilateration(remote_device_lat[0],remote_device_log[0],remote_devic
e_rssi[0],remote_device_prev_dis[0],

      remote_device_lat[1],remote_device_log[1],remote_device_rssi[1],remote_d
evice_prev_dis[1],
```

```
        remote_device_lat[2],remote_device_log[2],remote_device_rssi[2],remote_d
evice_prev_dis[2]);
                LatLng xx = new LatLng(latlong[0],latlong[1]);
                googleMap.addMarker(new MarkerOptions()
               .position(xx)
               .title("Predicted Location")
               .snippet("This is my spot!")

.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE))
);
                tvLocation.append(""+xx.latitude+";"+xx.longitude);

        out.write(xx.latitude+";"+xx.longitude+";"+latitude+";"+longitude+"\n");
        }
        else if (BluetoothLocationService.numberConnection==1)
        {
                if (ShellInterface.isSuAvailable())
                        ShellInterface.runCommand("myping -c 4000 -s 15 -d 0.001
80:57:19:61:09:5F");

                out.write(latitude+";"+longitude+"\n");

        }
        else
                out.write(latitude+";"+longitude+"\n");

                out.close();
        } //end try
        catch (IOException e)
        {
                Log.v(getString(R.string.app_name), "We
cannot......"+e.getMessage());

        }// end catch
}// end if

if(countrssi%60==0)
        countrssi=0;
if (BluetoothLocationService.numberConnection>0)
{

        googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng[0]));
        googleMap.animateCamera(CameraUpdateFactory.zoomTo(18));

        for (int i=0;i<BluetoothLocationService.numberConnection;i++)
    {
                switch (i)
                {
                case 0:
                     googleMap.addCircle(new CircleOptions()
                    .center(latLng[i])

.radius(Predict_Dis(remote_device_rssi[i],remote_device_prev_dis[i]))
                    .strokeColor(Color.BLUE)
```

```
                    .strokeWidth(2)
                    .fillColor(Color.TRANSPARENT))
                     break;
              case 1:
                       googleMap.addCircle(new CircleOptions()
                    .center(latLng[i])

.radius(Predict_Dis(remote_device_rssi[i],remote_device_prev_dis[i]))
                    .strokeColor(Color.RED)
                    .strokeWidth(2)
                    .fillColor(Color.TRANSPARENT));
                     break;
              case 2:
                     googleMap.addCircle(new CircleOptions()
                    .center(latLng[i])

.radius(Predict_Dis(remote_device_rssi[i],remote_device_prev_dis[i]))
                    .strokeColor(Color.GREEN)
                    .strokeWidth(2)
                    .fillColor(Color.TRANSPARENT));
                     break;
              case 3:
                     googleMap.addCircle(new CircleOptions()
                    .center(latLng[i])

.radius(Predict_Dis(remote_device_rssi[i],remote_device_prev_dis[i]))
                    .strokeColor(Color.CYAN)
                    .strokeWidth(2)
                    .fillColor(Color.TRANSPARENT));
                     break;
              case 4:
                     googleMap.addMarker(new MarkerOptions()
             .position(latLng[i])
             .title(remote_device[i])
             .snippet("This is my spot!")

.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ORANGE
)));
                     break;
              case 5:
                     googleMap.addMarker(new MarkerOptions()
             .position(latLng[i])
             .title(remote_device[i])
             .snippet("This is my spot!")

.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED)))
;
                     break;
              case 6:
                     googleMap.addMarker(new MarkerOptions()
             .position(latLng[i])
             .title(remote_device[i])
             .snippet("This is my spot!")
```

```
.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ROSE))
);
                   break;
             default:
                   googleMap.addMarker(new MarkerOptions()
            .position(latLng[i])
            .title(remote_device[i])
            .snippet("This is my spot!")

.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE)
));
             } //end switch

    } //end for
}// if for number connection
//add
if(tokenMsg[1].length()>0 && tokenMsg[1].startsWith("Slave"))
{
      String[] str=tokenMsg[1].split(",");
      int uuidIndex=Integer.parseInt(str[1]);
      String address=str[2];
      connected=false;
      mConversationArrayAdapter.clear();
      mChatService.stop();
      try
    {
        Thread.sleep(7000+uuidIndex*1000);
    }
    catch (InterruptedException e)
    {
        Log.e(TAG, "InterruptedException in connect", e);
    }
      mChatService=null;
      mChatService = new BluetoothLocationService(BluetoothApp.this,
mHandler);
      mChatService.start();
      ensureDiscoverable();
      BluetoothDevice device = mBluetoothAdapter.getRemoteDevice(address);
    mChatService.connect(device,mChatService.mUuids.get(uuidIndex)); // halo
    mTitle.setText("Salve");
    connected=true;
}
if(tokenMsg[1].length()>0 && tokenMsg[1].startsWith("Master"))
{
      mChatService.stop();
      connected=false;
      try
    {
        Thread.sleep(1000);
    }
    catch (InterruptedException e)
    {
        Log.e(TAG, "InterruptedException in connect", e);
                    }
```

```
                    mChatService=null;

                    mChatService = new
BluetoothLocationService(BluetoothApp.this, mHandler);
                    mChatService.start();
                    ensureDiscoverable();
                    mTitle.setText("Master");
       //connected=true;
             }
    }
    break;
case MESSAGE_DEVICE_NAME:
       count_devices++;
    // save the connected device's name
mConnectedDeviceName = msg.getData().getString(DEVICE_NAME);
Toast.makeText(getApplicationContext(), "Connected to "
                   + mConnectedDeviceName, Toast.LENGTH_SHORT).show();
    break;
case MESSAGE_TOAST:
       Toast.makeText(getApplicationContext(), msg.getData().getString(TOAST),
        Toast.LENGTH_SHORT).show();
//}
             break;
        }
    }
};


public void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(D) Log.d(TAG, "onActivityResult " + resultCode);
switch (requestCode) {
case REQUEST_CONNECT_DEVICE:
    // When DeviceListActivity returns with a device to connect
if (resultCode == Activity.RESULT_OK) {
    // Get the device MAC address
String address = data.getExtras()
                     .getString(DeviceListActivity.EXTRA_DEVICE_ADDRESS);
// Get the BLuetoothDevice object
BluetoothDevice device = mBluetoothAdapter.getRemoteDevice(address);
// Attempt to connect to the device
   mChatService.connect(device,UUID.fromString("b7746a40-c758-4868-aa19-
7ac6b3475dfc")); //shno
    }
    break;
case REQUEST_ENABLE_BT:
    // When the request to enable Bluetooth returns
if (resultCode == Activity.RESULT_OK) {
    // Bluetooth is now enabled, so set up a chat session
    setupChat();
} else {
    // User did not enable Bluetooth or an error occured
Log.d(TAG, "BT not enabled");
             Toast.makeText(this, R.string.bt_not_enabled_leaving,
Toast.LENGTH_SHORT).show();
             finish();
```

```
        }
    }
}
public int countlocation;
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.option_menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case R.id.scan:
        // Launch the DeviceListActivity to see devices and do scan
    Intent serverIntent = new Intent(this, DeviceListActivity.class);
    startActivityForResult(serverIntent, REQUEST_CONNECT_DEVICE);
    return true;
case R.id.discoverable:
    // Ensure this device is discoverable by others
        ensureDiscoverable();
        return true;
    }
    return false;
}
Location LastLocation;
int countloc;
double dis;
public String getDevName()
{
      if (ShellInterface.isSuAvailable()) {
            return ShellInterface.getProcessOutput("hcitool dev");
  }
return ("Not available!");
}
double latitude,longitude;
@Override
public void onLocationChanged(Location location) {
      // TODO Auto-generated method stub
// Getting latitude of the current location
latitude = location.getLatitude();

// Getting longitude of the current location
longitude = location.getLongitude();

// Creating a LatLng object for the current location
LatLng latLng = new LatLng(latitude, longitude);
String locInfo = String.format("%f,%f,%f", location.getLatitude(),
location.getLongitude(),location.getAccuracy());
// Showing the current location in Google Map
googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
// Zoom in the Google Map
googleMap.animateCamera(CameraUpdateFactory.zoomTo(18));
```

```
// Setting latitude and longitude in the TextView tv_location

countloc++;
if(countloc==1)
{
        Date dt = new Date();
     int hours = dt.getHours();
     int minutes = dt.getMinutes();
     int seconds = dt.getSeconds();
     String curTime = hours + "_"+minutes+ "_"+ seconds+"_Blue.txt";
root = Environment.getExternalStorageDirectory();
textFile = new File(root, "/"+curTime );
}
if(countloc%30==0)
{
             if(NearestNode()) //add
{
        if(changeMaster(NextMasterAddress)) //add
        {
        mChatService.stop();
        connected=false;
        BluetoothLocationService.numberConnection=0;
        try
          {
              Thread.sleep(5000);
          }
          catch (InterruptedException e)
          {
              Log.e(TAG, "InterruptedException in connect", e);
          }
        mChatService=null;
        mChatService = new BluetoothLocationService(BluetoothApp.this,
mHandler);
        mChatService.start();
        ensureDiscoverable();
        BluetoothDevice device =
mBluetoothAdapter.getRemoteDevice(NextMasterAddress);
          mChatService.connect(device,mChatService.mUuids.get(0)); // halo
          connected=true;
               mTitle.setText("Slave");
               }
        }
}

if (countloc>1)
{
        LastLocation=location;
}

static String NextMasterAddress;
public double distance_points(double lat1,double lat2,double lon1,double lon2)
{
        int EARTH_RADIUS_KM = 6371;
        double lat1Rad,lat2Rad,deltaLonRad,dist;
        lat1Rad = Math.toRadians(lat1);
```

```
        lat2Rad = Math.toRadians(lat2);

        deltaLonRad = Math.toRadians(lon2 - lon1);

        dist = Math.acos(Math.sin(lat1Rad) * Math.sin(lat2Rad) +
Math.cos(lat1Rad) * Math.cos(lat2Rad) * Math.cos(deltaLonRad)) *
EARTH_RADIUS_KM;
        dist=dist*1000.000000;
        return dist;
}
public boolean NearestNode()
{
        if(BluetoothLocationService.numberConnection>7)
        {
                double SumCurrentMasterDis=0.0;
                double[] MasterDis=new double[7];
                for(int i=0;i<BluetoothLocationService.numberConnection;i++)
                {

        MasterDis[i]=distance_points(latitude,remote_device_lat[i],
                                longitude,remote_device_log[i]);
                        SumCurrentMasterDis=SumCurrentMasterDis+MasterDis[i];
                }
                double[][] distances=new double[7][7];
                double[] SumSlaveDis=new double[7];
                for(int i=0;i<BluetoothLocationService.numberConnection;i++)
                {
                        for(int
j=0;j<BluetoothLocationService.numberConnection;j++)
                        {
                                if(i==j)

        distances[i][j]=distance_points(remote_device_lat[i],latitude,
                                                remote_device_log[i],longitude);
                                else

        distances[i][j]=distance_points(remote_device_lat[i],remote_device_lat[j
],

        remote_device_log[i],remote_device_log[j]);
                                SumSlaveDis[i]=SumSlaveDis[i]+distances[i][j];
                        }
                }
                String nextMaster;
                double min=SumSlaveDis[0];
                NextMasterAddress=remote_device[0];
                for(int i=0;i<BluetoothLocationService.numberConnection;i++)
                {
                        if(SumSlaveDis[i]<min && SumSlaveDis[i]!=0.0)
                        {
                                NextMasterAddress=remote_device[i];
                                min=SumSlaveDis[i];
                        }
                }
```

```
		SumCurrentMasterDis=SumCurrentMasterDis-5; //due to GPS error
reduce 5 meters
		if(SumCurrentMasterDis>min)
			return true;
		else
			return false;
	}
	return false;
}
public boolean changeMaster(String MasterAddress)
{
	if (BluetoothLocationService.numberConnection>0)
	{
		int index=1;
		ConnectedThread r;
		for(int i=0;i<BluetoothLocationService.numberConnection;i++)
		{
			r=BluetoothLocationService.mConnThreads.get(i);

	if(BluetoothLocationService.mDeviceAddresses.get(i).equals(MasterAddress
))
			{

				String
message=mBluetoothAdapter.getAddress()+"#"+"Master";
	byte[] send = message.getBytes();
	r.write(send);
}
else
{
	String
message=mBluetoothAdapter.getAddress()+"#"+"Slave,"+index+","+MasterAddress;
		byte[] send = message.getBytes();
		r.write(send);
		index++;
	}

}// end for loop
		return true;
	}
	else
		return false;
}// end method
static double wight=0.995;
}
```

BluetoothLocationService.java

```
package com.example.bluelocation;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.UUID;
import android.annotation.SuppressLint;
import android.bluetooth.BluetoothAdapter;
```

```java
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothServerSocket;
import android.bluetooth.BluetoothSocket;
import android.content.Context;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;
public class BluetoothLocationService {
private static final String TAG = "BluetoothLocationService";
private static final boolean D = true;
public int maxConnections=7;
public static int numberConnection=0;
// Name for the SDP record when creating server socket
private static final String NAME = "BluetoothMulti";
// Member fields
private final BluetoothAdapter mAdapter;
private final Handler mHandler;
private AcceptThread mAcceptThread;
private ConnectThread mConnectThread;
private ConnectedThread mConnectedThread;
private int mState;
public static ArrayList<String> mDeviceAddresses;
public static ArrayList<ConnectedThread> mConnThreads;
public ArrayList<BluetoothSocket> mSockets;
public ArrayList<UUID> mUuids;
// Constants that indicate the current connection state
public static final int STATE_NONE = 0;       // we're doing nothing
public static final int STATE_LISTEN = 1;     // now listening for incoming
connections
public static final int STATE_CONNECTING = 2; // now initiating an outgoing
connection
public static final int STATE_CONNECTED = 3;  // now connected to a remote
device
public BluetoothLocationService(Context context, Handler handler) {
    mAdapter = BluetoothAdapter.getDefaultAdapter();
    mState = STATE_NONE;
    mHandler = handler;
    mDeviceAddresses = new ArrayList<String>();
    mConnThreads = new ArrayList<ConnectedThread>();
    mSockets = new ArrayList<BluetoothSocket>();
    mUuids = new ArrayList<UUID>();
    // 7 randomly-generated UUIDs. These must match on both server and client.
mUuids.add(UUID.fromString("b7746a40-c758-4868-aa19-7ac6b3475dfc"));
mUuids.add(UUID.fromString("2d64189d-5a2c-4511-a074-77f199fd0834"));
mUuids.add(UUID.fromString("e442e09a-51f3-4a7b-91cb-f638491d1412"));
mUuids.add(UUID.fromString("a81d6504-4536-49ee-a475-7d96d09439e4"));
mUuids.add(UUID.fromString("aa91eab1-d8ad-448e-abdb-95ebba4a9b55"));
mUuids.add(UUID.fromString("4d34da73-d0a4-4f40-ac38-917e0a9dee97"));
mUuids.add(UUID.fromString("5e14d4df-9c8a-4db7-81e4-c937564c86e0"));
}
private synchronized void setState(int state) {
    if (D) Log.d(TAG, "setState() " + mState + " -> " + state);
mState = state;
```

```
    mHandler.obtainMessage(BluetoothApp.MESSAGE_STATE_CHANGE, state, -
1).sendToTarget();
}
public synchronized int getState() {
    return mState;
}
public synchronized void start() {
    if (D) Log.d(TAG, "start");

// Cancel any thread attempting to make a connection
if (mConnectThread != null) {mConnectThread.cancel(); mConnectThread = null;}

// Cancel any thread currently running a connection
if (mConnectedThread != null) {mConnectedThread.cancel(); mConnectedThread =
null;}

// Start the thread to listen on a BluetoothServerSocket
    if (mAcceptThread == null) {
        mAcceptThread = new AcceptThread();
        mAcceptThread.start();
    }
    setState(STATE_LISTEN);
}
public synchronized void connect(BluetoothDevice device,UUID c_uuidToTry) {
    if (D) Log.d(TAG, "connect to: " + device);
BluetoothSocket myBSock = null;
if (mState == STATE_CONNECTING) {
    if (mConnectThread != null) {mConnectThread.cancel(); mConnectThread =
null;}
}

// Cancel any thread currently running a connection
    if (mConnectedThread != null) {mConnectedThread.cancel(); mConnectedThread
= null;}
for (int j = 0; j < 3 && myBSock == null; j++)
{
mConnectThread = new ConnectThread(device,c_uuidToTry);
mConnectThread.start();
setState(STATE_CONNECTING);
myBSock=mConnectThread.mmSocket;
if (myBSock == null)
{
    try
    {
        Thread.sleep(200);
    } catch (InterruptedException e)
    {
        Log.e(TAG, "InterruptedException in connect", e);
            }
        }
   // }
}
       synchronized (BluetoothLocationService.this)
        {
            mConnectThread = null;
```

```
        }
}
public synchronized void connected(BluetoothSocket socket, BluetoothDevice
device) {
    if (D) Log.d(TAG, "connected");

mConnectedThread = new ConnectedThread(socket);
mConnectedThread.start();
// Add each connected thread to an array
mConnThreads.add(mConnectedThread);
// Send the name of the connected device back to the UI Activity
    Message msg = mHandler.obtainMessage(BluetoothApp.MESSAGE_DEVICE_NAME);
    Bundle bundle = new Bundle();
    bundle.putString(BluetoothApp.DEVICE_NAME, device.getName());
    msg.setData(bundle);
    mHandler.sendMessage(msg);

    setState(STATE_CONNECTED);
}
public synchronized void stop() {
    if (D) Log.d(TAG, "stop");
    if (mConnectThread != null) {mConnectThread.cancel(); mConnectThread =
null;}
    if (mConnectedThread != null) {mConnectedThread.cancel(); mConnectedThread
= null;}
    if (mAcceptThread != null) {mAcceptThread.cancel(); mAcceptThread = null;}
    setState(STATE_NONE);
}
public void write(byte[] out) {
      // When writing, try to write out to all connected threads
for (int i = 0; i < mConnThreads.size(); i++) {
      try {
       // Create temporary object
ConnectedThread r;
// Synchronize a copy of the ConnectedThread
synchronized (this) {
    if (mState != STATE_CONNECTED) return;
    r = mConnThreads.get(i);
}
// Perform the write unsynchronized
            r.write(out);
             } catch (Exception e) {
             }
      }
}
private void connectionFailed() {
    setState(STATE_LISTEN);
}
private void connectionLost() {
    setState(STATE_LISTEN);

    // Send a failure message back to the Activity
Message msg = mHandler.obtainMessage(BluetoothApp.MESSAGE_TOAST);
Bundle bundle = new Bundle();
bundle.putString(BluetoothApp.TOAST, "Device connection was lost");
```

```
    msg.setData(bundle);
    mHandler.sendMessage(msg);
}
private class AcceptThread extends Thread {
      BluetoothServerSocket serverSocket = null;

    public AcceptThread() {
    }

      public void run() {
       if (D) Log.d(TAG, "BEGIN mAcceptThread" + this);
setName("AcceptThread");
BluetoothSocket socket = null;

try {
      // Listen for all 7 UUIDs

for (int i = 0; i < 7 && maxConnections > 0; i++)
{
             serverSocket = mAdapter.listenUsingRfcommWithServiceRecord(NAME,
mUuids.get(i));
        socket = serverSocket.accept();
        serverSocket.close();
        if (socket != null)
        {
             String address = socket.getRemoteDevice().getAddress();
            mSockets.add(socket);
            mDeviceAddresses.add(address);
            connected(socket, socket.getRemoteDevice());
            maxConnections=maxConnections-1;
            numberConnection=numberConnection+1;
            //break;
            }
      }
} catch (IOException e) {
    Log.e(TAG, "accept() failed", e);
}
if (D) Log.i(TAG, "END mAcceptThread");
}

public void cancel() {
    if (D) Log.d(TAG, "cancel " + this);
try {
    serverSocket.close();
} catch (IOException e) {
    Log.e(TAG, "close() of server failed", e);
        }
    }
}
private class ConnectThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final BluetoothDevice mmDevice;
    private UUID tempUuid;

    public ConnectThread(BluetoothDevice device, UUID uuidToTry) {
```

```
        mmDevice = device;
        BluetoothSocket tmp = null;
        tempUuid = uuidToTry;

        // Get a BluetoothSocket for a connection with the
// given BluetoothDevice
try {
    tmp = device.createRfcommSocketToServiceRecord(uuidToTry);
} catch (IOException e) {
    Log.e(TAG, "create() failed", e);
    }
    mmSocket = tmp;
}

public void run()
{
    Log.i(TAG, "BEGIN mConnectThread");
setName("ConnectThread");

// Always cancel discovery because it will slow down a connection
mAdapter.cancelDiscovery();

// Make a connection to the BluetoothSocket
try {
    // This is a blocking call and will only return on a
// successful connection or an exception
    mmSocket.connect();
}
catch (IOException e)
{
       if (tempUuid.toString().contentEquals(mUuids.get(6).toString()))
       {
        connectionFailed();
       }
    // Close the socket
try
{
    mmSocket.close();
}
catch (IOException e2)
{
    Log.e(TAG, "unable to close() socket during connection failure", e2);
}
// Start the service over to restart listening mode
        BluetoothLocationService.this.start();
        return;
    }
    // Start the connected thread
    connected(mmSocket, mmDevice);
}

public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) {
```

```
            Log.e(TAG, "close() of connect socket failed", e);
            }
        }
    }
    public class ConnectedThread extends Thread {
        private final BluetoothSocket mmSocket;
        private final InputStream mmInStream;
        private final OutputStream mmOutStream;

        public ConnectedThread(BluetoothSocket socket) {
            Log.d(TAG, "create ConnectedThread");
    mmSocket = socket;
    InputStream tmpIn = null;
    OutputStream tmpOut = null;

    // Get the BluetoothSocket input and output streams
    try {
        tmpIn = socket.getInputStream();
        tmpOut = socket.getOutputStream();
    } catch (IOException e) {
        Log.e(TAG, "temp sockets not created", e);
        }

        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }

    public void run() {
        Log.i(TAG, "BEGIN mConnectedThread");
    byte[] buffer = new byte[1024];
    int bytes;

    // Keep listening to the InputStream while connected
    while (true) {
        try {
            // Read from the InputStream
    bytes = mmInStream.read(buffer);

    // Send the obtained bytes to the UI Activity
        mHandler.obtainMessage(BluetoothApp.MESSAGE_READ, bytes, -1, buffer)
                .sendToTarget();
    } catch (IOException e) {
        Log.e(TAG, "disconnected", e);
                connectionLost();
                break;
            }
        }
    }
    public void write(byte[] buffer) {
        try {
            mmOutStream.write(buffer);

            // Share the sent message back to the UI Activity
        mHandler.obtainMessage(BluetoothApp.MESSAGE_WRITE, -1, -1, buffer)
                .sendToTarget();
```

```
} catch (IOException e) {
    Log.e(TAG, "Exception during write", e);
    }
}

public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of connect socket failed", e);
            }
        }
    }
}
```

DeviceListActivity.java

```
package com.example.bluelocation;
import java.util.Set;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.AdapterView.OnItemClickListener;
public class DeviceListActivity extends Activity {
    // Debugging
private static final String TAG = "DeviceListActivity";
private static final boolean D = true;
// Return Intent extra
public static String EXTRA_DEVICE_ADDRESS = "device_address";
// Member fields
private BluetoothAdapter mBtAdapter;
private ArrayAdapter<String> mPairedDevicesArrayAdapter;
private ArrayAdapter<String> mNewDevicesArrayAdapter;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
// Setup the window
requestWindowFeature(Window.FEATURE_INDETERMINATE_PROGRESS);
setContentView(R.layout.device_list);
// Set result CANCELED incase the user backs out
setResult(Activity.RESULT_CANCELED);
```

```
// Initialize the button to perform device discovery
Button scanButton = (Button) findViewById(R.id.button_scan);
scanButton.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        doDiscovery();
        v.setVisibility(View.GONE);
    }
});

// Initialize array adapters. One for already paired devices and
// one for newly discovered devices
mPairedDevicesArrayAdapter = new ArrayAdapter<String>(this,
R.layout.device_name);
mNewDevicesArrayAdapter = new ArrayAdapter<String>(this,
R.layout.device_name);
// Find and set up the ListView for paired devices
ListView pairedListView = (ListView) findViewById(R.id.paired_devices);
pairedListView.setAdapter(mPairedDevicesArrayAdapter);
pairedListView.setOnItemClickListener(mDeviceClickListener);
// Find and set up the ListView for newly discovered devices
ListView newDevicesListView = (ListView) findViewById(R.id.new_devices);
newDevicesListView.setAdapter(mNewDevicesArrayAdapter);
newDevicesListView.setOnItemClickListener(mDeviceClickListener);
// Register for broadcasts when a device is discovered
IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
this.registerReceiver(mReceiver, filter);
// Register for broadcasts when discovery has finished
filter = new IntentFilter(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);
this.registerReceiver(mReceiver, filter);
// Get the local Bluetooth adapter
mBtAdapter = BluetoothAdapter.getDefaultAdapter();
// Get a set of currently paired devices
Set<BluetoothDevice> pairedDevices = mBtAdapter.getBondedDevices();
// If there are paired devices, add each one to the ArrayAdapter
if (pairedDevices.size() > 0) {
    findViewById(R.id.title_paired_devices).setVisibility(View.VISIBLE);
    for (BluetoothDevice device : pairedDevices) {
        mPairedDevicesArrayAdapter.add(device.getName() + "\n" +
device.getAddress());
        }
    } else {
        String noDevices =
getResources().getText(R.string.none_paired).toString();
        mPairedDevicesArrayAdapter.add(noDevices);
    }
}
@Override
protected void onDestroy() {
    super.onDestroy();
    // Make sure we're not doing discovery anymore
if (mBtAdapter != null) {
    mBtAdapter.cancelDiscovery();
}
// Unregister broadcast listeners
    this.unregisterReceiver(mReceiver);
```

```
}
private void doDiscovery() {
    if (D) Log.d(TAG, "doDiscovery()");

// Indicate scanning in the title
setProgressBarIndeterminateVisibility(true);
setTitle(R.string.scanning);

// Turn on sub-title for new devices
findViewById(R.id.title_new_devices).setVisibility(View.VISIBLE);

// If we're already discovering, stop it
if (mBtAdapter.isDiscovering()) {
    mBtAdapter.cancelDiscovery();
}

// Request discover from BluetoothAdapter
    mBtAdapter.startDiscovery();
}

// The on-click listener for all devices in the ListViews
private OnItemClickListener mDeviceClickListener = new OnItemClickListener() {
    public void onItemClick(AdapterView<?> av, View v, int arg2, long arg3) {
        // Cancel discovery because it's costly and we're about to connect
mBtAdapter.cancelDiscovery();

// Get the device MAC address, which is the last 17 chars in the View
String info = ((TextView) v).getText().toString();
String address = info.substring(info.length() - 17);

// Create the result Intent and include the MAC address
Intent intent = new Intent();
intent.putExtra(EXTRA_DEVICE_ADDRESS, address);

// Set result and finish this Activity
        setResult(Activity.RESULT_OK, intent);
        finish();
    }
};

// The BroadcastReceiver that listens for discovered devices and
// changes the title when discovery is finished
private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();

        // When discovery finds a device
if (BluetoothDevice.ACTION_FOUND.equals(action)) {
    // Get the BluetoothDevice object from the Intent
BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
// If it's already paired, skip it, because it's been listed already
if (device.getBondState() != BluetoothDevice.BOND_BONDED) {
```

```
    mNewDevicesArrayAdapter.add(device.getName() + "\n" +
device.getAddress());
}
// When discovery is finished, change the Activity title
            } else if
(BluetoothAdapter.ACTION_DISCOVERY_FINISHED.equals(action)) {
                setProgressBarIndeterminateVisibility(false);
                setTitle(R.string.select_device);
                if (mNewDevicesArrayAdapter.getCount() == 0) {
                    String noDevices =
getResources().getText(R.string.none_found).toString();
                    mNewDevicesArrayAdapter.add(noDevices);
                }
            }
        }
    };

}
```

C.2 The Java code in Android-based SPs to detect the step and then to calculate step length

```java
/**
    UNILS SHEME on Android-based SPs
    Sensors_Project
    Purpose: Reading accelerometer sensor to estimate distance-displacement

    @author Halgurd S Maghdid
    @version 1.0 August 2014
*/
```

StepActivity.java

```java
   package com.example.Sensors;
   public class StepActivity {
   public double step_detect_length(double accx[],double accy[],double accz[])
   {
            double step_length=0.0;
            double[]acc=new double[accx.length];
            int i,j;
            for (i=0;i<accx.length;i++)
      acc[i]=Math.sqrt(Math.pow(accx[i],2)+Math.pow(accy[i],2)+Math.pow(accz[i
   ],2));
            //Applying high pass filter
             double gravity=9.81; int r=acc.length;
             double[] acc_high_filt=new double[acc.length];
             double[] acc_low_filt=new double[acc.length];
             for(i=0;i<r;i++)
             {
                acc_high_filt[i]=acc[i]-gravity;
                gravity=acc[i]*0.1+gravity*0.9;
             }
            //Applying low pass filter
            double sum=0.0;
            acc_low_filt[0]=acc[0]; acc_low_filt[1]=acc[1];
            for(i=2;i<r-2;i++)
            {
                  for(j=i-2;j<i+3;j++)
                        sum=sum+acc_high_filt[j];
                  acc_low_filt[i]=sum/5;
            }
            //Applying detecting steps using peak-detection algorithm
            int count=0;
            double th=0.0;
            double[] max_peaks=new double[r/10];
            double[] min_peaks=new double[r/10];
            int[] max_indx=new int[r/10];
            int[] min_indx=new int[r/10];
            double max=acc_low_filt[0], min=acc_low_filt[0];
            int max_in=0, min_in=0;
            for(i=1;i<r-10;i=i+10)
```

```
	{
		for(j=i;j<i+10;j++)
		{
			if(max<acc_low_filt[j])
			{
				max=acc_low_filt[j];
				max_in=j;
			}
			if(min>acc_low_filt[j])
			{
				min=acc_low_filt[j];
				min_in=j;
			}

		}
		max_peaks[count]=max;
		max_indx[count]=max_in;
		min_peaks[count]=min;
		min_indx[count]=min_in;
		th=th+(max_peaks[count]-min_peaks[count]);
		count=count+1;
	}
	th=th/count;
	count=0;
	for(i=1;i<r-10;i=i+10)
	{
		if(max_peaks[i]>(th/2-0.15))
		{
			max_peaks[count]=max_peaks[i];
			max_indx[count]=max_indx[i];
			count++;
		}
		if(max_peaks[i]<(-1*th/2+0.15))
		{
			min_peaks[count]=min_peaks[i];
			min_indx[count]=min_indx[i];
			count++;
		}
	}
	double k_scarlet=0.6, avg_acc=0.0;
	//Calculate step length by scarlet algorithm
	for(i=0;i<min_peaks.length-1;i++)
	{
		sum=0.0;
		if((max_peaks[i]-min_peaks[i])!=0)
		{
			for(j=max_indx[i];j<max_indx[i+1];j++)
				sum=sum+Math.abs(acc_low_filt[j]);
		    avg_acc=((sum/j)-min_peaks[i])/(max_peaks[i]-
min_peaks[i]);
			step_length=k_scarlet*avg_acc;
		}
	}
	return step_length;}}
```

C.3 The Java code in Android-based SPs for estimating SPs-heading

```
/**
    UNILS SHEME on Android-based SPs
    Sensors_Project
    Purpose: Reading gyroscope sensor to estimate Heading

    @author Halgurd S Maghdid
    @version 1.0 September 2014
*/
```

HeadingActivity.java

```
package com.example.Sensors;
import android.app.Activity;
import android.content.Context;
import android.hardware.GeomagneticField;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.util.Log;
import android.widget.RadioGroup;
import android.widget.TextView;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.math.RoundingMode;
import java.text.DecimalFormat;
import java.util.Date;
import java.util.Timer;
import java.util.TimerTask;
import com.thousandthoughts.tutorials.R;
public class HeadingActivity extends Activity
implements SensorEventListener, RadioGroup.OnCheckedChangeListener,
LocationListener {
      LocationManager location = null;
      File root;
      File textFile;
      FileWriter textWriter;
      String curTime;
      private SensorManager mSensorManager = null;
    // angular speeds from gyro
    private float[] gyro = new float[3];
    // rotation matrix from gyro data
    private float[] gyroMatrix = new float[9];
```

```
    // orientation angles from gyro matrix
    private float[] gyroOrientation = new float[3];
    // magnetic field vector
    private float[] magnet = new float[3];
    // accelerometer vector
    private float[] accel = new float[3];
    // orientation angles from accel and magnet
    private float[] accMagOrientation = new float[3];

    // final orientation angles from sensor fusion
    private float[] fusedOrientation = new float[3];

    // accelerometer and magnetometer based rotation matrix
private float[] rotationMatrix = new float[9];

public static final float EPSILON = 0.000000001f;
private static final float NS2S = 1.0f / 1000000000.0f;
private float timestamp;
private boolean initState = true;

public static final int TIME_CONSTANT = 30;
public static final float FILTER_COEFFICIENT = 0.98f;
private Timer fuseTimer = new Timer();

// The following members are only for displaying the sensor output.
public Handler mHandler;
private RadioGroup mRadioGroup;
private TextView mAzimuthView;
private TextView GPSView;
private TextView mPitchView;
private TextView mRollView;
private int radioSelection;
DecimalFormat d = new DecimalFormat("#.##");


@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
gyroOrientation[0] = 0.0f;
gyroOrientation[1] = 0.0f;
gyroOrientation[2] = 0.0f;
gyroMatrix[0] = 1.0f; gyroMatrix[1] = 0.0f; gyroMatrix[2] = 0.0f;
gyroMatrix[3] = 0.0f; gyroMatrix[4] = 1.0f; gyroMatrix[5] = 0.0f;
gyroMatrix[6] = 0.0f; gyroMatrix[7] = 0.0f; gyroMatrix[8] = 1.0f;
// get Location Manager
location = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
// get sensorManager and initialise sensor listeners
mSensorManager = (SensorManager) this.getSystemService(SENSOR_SERVICE);
initListeners();
fuseTimer.scheduleAtFixedRate(new calculateFusedOrientationTask(),
                              1000, TIME_CONSTANT);
// GUI stuff
mHandler = new Handler();
radioSelection = 0;
```

```
d.setRoundingMode(RoundingMode.HALF_UP);
d.setMaximumFractionDigits(3);
d.setMinimumFractionDigits(3);
mRadioGroup = (RadioGroup)findViewById(R.id.radioGroup1);
mAzimuthView = (TextView)findViewById(R.id.textView4);
mPitchView = (TextView)findViewById(R.id.textView5);
mRollView = (TextView)findViewById(R.id.textView6);
GPSView=(TextView)findViewById(R.id.textView8);
mRadioGroup.setOnCheckedChangeListener(this);
Date dt = new Date();
int hours = dt.getHours();
int minutes = dt.getMinutes();
int seconds = dt.getSeconds();
curTime = hours + "_"+minutes+ "_"+ seconds+"_Blue.txt";
    root = Environment.getExternalStorageDirectory();
}
@Override
public void onStop() {
       super.onStop();
       mSensorManager.unregisterListener(this);
}
@Override
public void onResume() {
       super.onResume();
       // restore the sensor listeners when user resumes the application.
       initListeners();
}
public void initListeners(){
    mSensorManager.registerListener(this,
        mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
        50000);
    mSensorManager.registerListener(this,
        mSensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE),
        50000);
    mSensorManager.registerListener(this,
        mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD),
        50000);

    location.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000, 0,
HeadingActivity.this);
}
@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
}
@Override
public void onSensorChanged(SensorEvent event) {
       switch(event.sensor.getType()) {
    case Sensor.TYPE_ACCELEROMETER:
        // copy new accelerometer data into accel array and calculate
orientation
        System.arraycopy(event.values, 0, accel, 0, 3);
        calculateAccMagOrientation();
        break;

    case Sensor.TYPE_GYROSCOPE:
```

```
            // process gyro data
            gyroFunction(event);
            break;

        case Sensor.TYPE_MAGNETIC_FIELD:
            // copy new magnetometer data into magnet array
            System.arraycopy(event.values, 0, magnet, 0, 3);
            break;
        }
    }
    // calculates orientation angles from accelerometer and magnetometer output
    public void calculateAccMagOrientation() {
        if(SensorManager.getRotationMatrix(rotationMatrix, null, accel, magnet)) {
            SensorManager.getOrientation(rotationMatrix, accMagOrientation);
        }
    }
    private void getRotationVectorFromGyro(float[] gyroValues,float[]
    deltaRotationVector,
            float timeFactor)
    {
          float[] normValues = new float[3];

          // Calculate the angular speed of the sample
    float omegaMagnitude =
    (float)Math.sqrt(gyroValues[0] * gyroValues[0] +
    gyroValues[1] * gyroValues[1] +
    gyroValues[2] * gyroValues[2]);

    // Normalize the rotation vector if it's big enough to get the axis
    if(omegaMagnitude > EPSILON) {
    normValues[0] = gyroValues[0] / omegaMagnitude;
    normValues[1] = gyroValues[1] / omegaMagnitude;
    normValues[2] = gyroValues[2] / omegaMagnitude;
    }

          float thetaOverTwo = omegaMagnitude * timeFactor;
          float sinThetaOverTwo = (float)Math.sin(thetaOverTwo);
          float cosThetaOverTwo = (float)Math.cos(thetaOverTwo);
          deltaRotationVector[0] = sinThetaOverTwo * normValues[0];
          deltaRotationVector[1] = sinThetaOverTwo * normValues[1];
          deltaRotationVector[2] = sinThetaOverTwo * normValues[2];
          deltaRotationVector[3] = cosThetaOverTwo;
    }
    public void gyroFunction(SensorEvent event)
    {
         if (accMagOrientation == null)
            return;

        // initialisation of the gyroscope based rotation matrix
        if(initState) {
            float[] initMatrix = new float[9];
            initMatrix = getRotationMatrixFromOrientation(accMagOrientation);
            float[] test = new float[3];
            SensorManager.getOrientation(initMatrix, test);
            gyroMatrix = matrixMultiplication(gyroMatrix, initMatrix);
```

```
        initState = false;
    }

    float[] deltaVector = new float[4];
    if(timestamp != 0) {
        final float dT = (event.timestamp - timestamp) * NS2S;
    System.arraycopy(event.values, 0, gyro, 0, 3);
    getRotationVectorFromGyro(gyro, deltaVector, dT / 2.0f);
    }
    // measurement done, save current time for next interval
    timestamp = event.timestamp;
    // convert rotation vector into rotation matrix
    float[] deltaMatrix = new float[9];
    SensorManager.getRotationMatrixFromVector(deltaMatrix, deltaVector);
    // apply the new rotation interval on the gyroscope based rotation matrix
    gyroMatrix = matrixMultiplication(gyroMatrix, deltaMatrix);
    // get the gyroscope based orientation from the rotation matrix
    SensorManager.getOrientation(gyroMatrix, gyroOrientation);
}

private float[] getRotationMatrixFromOrientation(float[] o) {
    float[] xM = new float[9];
    float[] yM = new float[9];
    float[] zM = new float[9];
    float sinX = (float)Math.sin(o[1]);
    float cosX = (float)Math.cos(o[1]);
    float sinY = (float)Math.sin(o[2]);
    float cosY = (float)Math.cos(o[2]);
    float sinZ = (float)Math.sin(o[0]);
    float cosZ = (float)Math.cos(o[0]);
    // rotation about x-axis (pitch)
    xM[0] = 1.0f; xM[1] = 0.0f; xM[2] = 0.0f;
    xM[3] = 0.0f; xM[4] = cosX; xM[5] = sinX;
    xM[6] = 0.0f; xM[7] = -sinX; xM[8] = cosX;
    // rotation about y-axis (roll)
    yM[0] = cosY; yM[1] = 0.0f; yM[2] = sinY;
    yM[3] = 0.0f; yM[4] = 1.0f; yM[5] = 0.0f;
    yM[6] = -sinY; yM[7] = 0.0f; yM[8] = cosY;
    // rotation about z-axis (azimuth)
    zM[0] = cosZ; zM[1] = sinZ; zM[2] = 0.0f;
    zM[3] = -sinZ; zM[4] = cosZ; zM[5] = 0.0f;
    zM[6] = 0.0f; zM[7] = 0.0f; zM[8] = 1.0f;
    // rotation order is y, x, z (roll, pitch, azimuth)
    float[] resultMatrix = matrixMultiplication(xM, yM);
    resultMatrix = matrixMultiplication(zM, resultMatrix);
    return resultMatrix;
}

private float[] matrixMultiplication(float[] A, float[] B) {
    float[] result = new float[9];
    result[0] = A[0] * B[0] + A[1] * B[3] + A[2] * B[6];
    result[1] = A[0] * B[1] + A[1] * B[4] + A[2] * B[7];
    result[2] = A[0] * B[2] + A[1] * B[5] + A[2] * B[8];
    result[3] = A[3] * B[0] + A[4] * B[3] + A[5] * B[6];
    result[4] = A[3] * B[1] + A[4] * B[4] + A[5] * B[7];
```

```
    result[5] = A[3] * B[2] + A[4] * B[5] + A[5] * B[8];
    result[6] = A[6] * B[0] + A[7] * B[3] + A[8] * B[6];
    result[7] = A[6] * B[1] + A[7] * B[4] + A[8] * B[7];
    result[8] = A[6] * B[2] + A[7] * B[5] + A[8] * B[8];
    return result;
}
public void updateOreintationDisplay()
{

        try
        {
                textFile = new File(root, "/"+curTime );
                textWriter = new FileWriter(textFile, true);
                BufferedWriter out = new BufferedWriter(textWriter);
                out.write((timestamp*NS2S)+";"+accel[0]+";"+accel[1]+";"+accel[2]
                +";"+gyro[0]+";"+gyro[1]+";"+gyro[2]+";"+magnet[0]+";"+magnet[1]+
                ";"+magnet[2]+";"+myBearing+";"+dec+"\n");
                out.close();
        }
        catch (IOException e)
        {
                Log.v(getString(R.string.app_name), "We
        cannot......"+e.getMessage());
        }// end catch
        mAzimuthView.setText(d.format(gyroOrientation[0] * 180/Math.PI) + 'd');
        mPitchView.setText(d.format(gyroOrientation[1] * 180/Math.PI) + 'd');
        mRollView.setText(d.format(gyroOrientation[2] * 180/Math.PI) + 'd');
}

private Runnable updateOreintationDisplayTask = new Runnable()
        {
                public void run() {
                        updateOreintationDisplay();
                }
        };
}
```