# DISTRIBUTED OBJECT-LOCALIZATION USING RFID CROWDSOURCING

by

Lobna Eslim

A thesis submitted to the

School of Computing

in conformity with the requirements for

the degree of Doctor of Philosophy

Queen’s University

Kingston, Ontario, Canada

August, 2015

# Abstract

Internet of Things (IoT) refers to an evolution of the current Internet in which a large number of “smart objects” sense their surroundings and communicate amongst themselves and to data analytic servers. IoT applications are rooted in our physical world to offer users more convenient context and location-aware services, for which a common requirement is the ability to locate objects. Two approaches are proposed to localize IoT objects based on Radio Frequency IDentification (RFID) technology: localize mobile/stationary tagged objects through a set of coordinated readers that report to a central server and localize mobile reader based on connectivity information with a set of tags deployed at known locations. The former is based on a centralized and fixed infrastructure which provides limited scalability while the latter is not cost effective for IoT settings as a large number of objects have to be equipped with RFID readers. In a typical IoT environment, there are considerable RFID crowdsourcing resources in terms of a large number of tags attached to objects and a considerable group of ad hoc mobile readers which are possibly heterogeneous and un-coordinated and can be used for locating objects.

We investigate this promising direction and devise distributed localization schemes that leverage heterogeneous and independent mobile RFID readers along with RFID tags’ residual memories to cooperatively localize passive-tagged objects, while maintaining high scalability. In estimating object location, Multilateration is a commonly used technique that estimates object location based on the intersection of all plausible areas where the object is expected to exist. This technique requires at least three concurrent readings about an object to estimate its location which is a challenge under IoT settings. We address ways to overcome this challenge and provide better location accuracy in the absence of sufficient concurrent readings. We propose location information dissemination strategies that work on providing high location information availability with low overhead. We validate our schemes via extensive simulation and field experiments and show that our approach has the potential to provide localization service in typical IoT environments.

# Acknowledgements

First of all I thank Allah (my God), the most gracious and merciful as he gave me the ability and the power to finalize this research. I am grateful to my wonderful supervisor Dr. Hossam S. Hassanein for his encouragement, support, guidance and continued enthusiasm and patience throughout my research.

My thankfulness to my beautiful family, my husband Ashraf Shalaby and my children Ahmed and Omar, for supporting me and pushing me forward especially during tough times, without them my graduate degree would not have been achieved. I owe every success I ever had to my dear parents, although they have passed away, I remember their words to me, words that give me the spirit and the motivation to achieve my goals and dreams.

I would like to thank my colleagues at the Queen’s Telecommunications Research Lab. Walid Ibrahim, Sharief Oteafy, Abdallah Alma'aitah, Mahmoud Qutqut, Sherin Abdel Hamid, Khaled Elgazzar, Abdulmonem Rashwan, Yaser Al Mtawa, and everyone else for their friendship and the many fruitful discussions and collaborations. Special thanks to Walid Ibrahim for teaching me the network simulation.

I am in debt to Basia Palmer our research assistant at the Queen’s TRL, as she always represents the hidden force behind producing a well written article and thesis, no words could describe how supportive and helpful she is.

Thanks as well to my supervisory committee, Dr. Nick Graham and Dr. Mohammad Zulkernine, for their help and valuable advice during my PhD. program. I would also like to thank the members of the School of Computing at Queen’s University who continuously work to provide us with an enjoyable and productive research environment.

# Statement of Originality

I hereby certify that all of the work described within this Ph.D. thesis is original and that all ideas and/or techniques from the work of others have been properly referenced in accordance with the standard referencing practices.

Lobna Eslim

June, 2015

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| **AoA** | Angle of Arrival |
| **Bluetooth** | Short range wireless connectivity standard |
| **DSSS** | Direct Sequence Spread Spectrum |
| **DTN** | Delay Tolerant Network |
| **GBMM** | Graph-Based Mobility Model |
| **GPS** | Global Positioning System |
| **HF** | High Frequency |
| **ID** | Identifier |
| **INS** | Inertial Navigation System |
| **IoT** | Internet of Things |
| **IPS** | Indoor Positioning System |
| **ISO** | International Organization for Standardization |
| **IR** | Infra-Red |
| **LF** | Low Frequency |
| **LOS** | Line Of Sight |
| **LPS** | Local Positioning System |
| **M2M** | Machine to Machine communication |
| **MW** | Micro Wave |
| **RF** | Radio Frequency |
| **RFID** | Radio Frequency IDentification |
| **RSS** | Received Signal Strength |
| **RSSI** | Received Signal Strength Indicator |
| **RToA** | Roundtrip Time of Flight |
| **RWMM** | Random Way Mobility Model |

| | |
|---|---|
| **SNR** | Signal to Noise Ratio |
| **TDoA** | Time Difference of Arrival |
| **ToA** | Time of Arrival |
| **UHF** | Ultra High Frequency |
| **UWB** | Ultra Wide Band |
| **Wi-Fi** | a worldwide standard for high-speed local area networking [50] |
| **WLAN** | Wireless Local Area Network |
| **WSN(s)** | Wireless Sensor Network(s) |
| **XML** | eXtensible Markup Language |
| **ZigBee** | a low-cost, low-power wireless connectivity standard [51] |

# Chapter 1

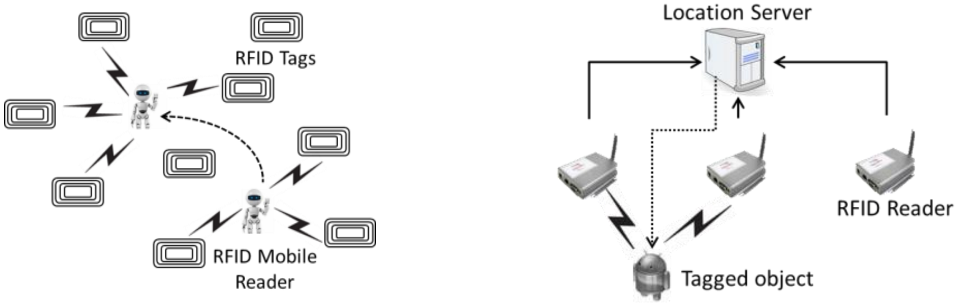
# Introduction

The ability to embed intelligence into everyday objects, by augmenting objects with different sensors and actuators, creates autonomous "smart objects" with the ability to interact with their environment. Smart objects sense and interpret their surroundings and, by means of unique addressing schemes, communicate amongst themselves and to data analytic servers forming pervasive computing environments [1] [2]. The popularity of these smart objects in our daily life along with the possibility of having millions or billions of them sprouts the idea of the Internet of Things (IoT). The term IoT was originally used by the Radio Frequency IDentification (RFID) development community, and can be broadly defined as a decentralized system of smart objects which are seamlessly integrated into an information network; continuously providing a variety of smart services via the Internet with support of a set of enabling technologies and communication solutions. The realization of IoT faces a number of challenges, including heterogeneity of objects, resource efficiency, scalability, interoperability and data interpretation, localization and tracking, as well as security and privacy [3] [4].

IoT applications span a wide and diverse range of domains such as: transportation, healthcare, smart environments, environmental monitoring, inventory and product management, and security and surveillance [5]. These applications are rooted in the physical realm typically to offer users more convenient context and location-aware services, each of which has different requirements. For context information to be useful, and for enabling location-based services, the ability to locate objects is essential. This problem is known as "object localization" [6] - [13], which generally refers to the process of determining where the object of interest is be it static or mobile. In the case of a static object, its location needs to be determined once. However, location sampling is required periodically to determine the location of a mobile object.

RFID, a key technology in IoT, stands at the forefront for the purpose of object identification and tracking [2] [14] [15]. In the past few years, RFID development has achieved unprecedented technical progress in addition to cost reductions and standardization [16]; resulting in unconventional utilizations and massive deployments beyond mere identification (e.g., asset tracking, manufacturing, payment systems, security and access control) [17]. RFID can provide localization service in an inexpensive, reliable, flexible and scalable manner, which are key requirements in IoT applications.

A typical RFID system is composed of a set of tags (passive, semi-passive or active according to the source of power and self-capabilities) and a set of readers, the passive tag is the least and the reader is the most expensive. Accordingly, several RFID localization systems have been proposed to localize mobile/stationary tagged objects or mobile readers [13] [18] [19]. In reader localization, objects are equipped with RFID readers and are localized based on connectivity information with a set of active or passive tags deployed at known locations. Whereas in tag localization, objects are embedded with RFID tags and are localized through a set of coordinated RFID readers. Readers report tags' spatial information (i.e., tag ID, time of detection and tag to reader distance) to a central server for location estimation as shown in Figure 1.1. Reader localization is well suited for autonomous mobile robot applications but it is not cost effective for IoT settings as a large number of objects have to be equipped with RFID readers. While the tag localization approach has a lower cost than the reader localization approach, the former is based on a centralized and fixed infrastructure, which provides limited scalability and may not be practical in IoT settings.



(a) Reader localization

(b) Tag localization

**Figure 1.1: Classification of RFID-based localization systems.**

## 1.1 Motivation

IoT applications have become very popular in our daily life offering users more convenient services through a typically large number of spatially disseminated smart objects in surrounding environments (e.g., homes, buildings, transportation). In such applications, each smart object plays a role in a distributed network of heterogeneous context-aware devices. Thus, IoT applications depend on the location on smart objects as a key functionality to take advantage of their context. Object localization solutions for such applications should address the challenges resulting from the IoT characteristics (i.e., scalability, heterogeneity of objects and dynamicity).

Smart objects are typically embedded with RFID tags or readers. Embedded RFID readers in mobile devices are being rapidly adopted due to the great interest of RFID manufacturers, along with the fast advancements in antenna design for handheld RFID readers [20] [21]. Moreover, tags have residual memories which are capable of storing data in addition to their unique identifiers, and can be utilized and shared in a given area to enhance data exchange [16] [22]. Considering this unceasing progress and proliferation, it is typical to have an environment that is comprised of a large number of RFID-tagged objects and a considerable group of ad hoc mobile readers, which are possibly heterogeneous and un-coordinated. Examples of such environments are shopping malls, airports and attractions. Participants in these environments are typically interested in localizing some objects depending on their changing needs and context, typically there are ad hoc RFID resources that can be utilized in order to provide object localization based on crowdsourcing.

The available RFID readers are distributed, heterogeneous, and may not identify and/or communicate with one another. In addition, due to their ad hoc mobility, concurrent spatial information about surrounding tagged objects may not be available or sufficient to estimate the objects' position. Existing RFID systems are not designed to leverage such heterogeneous distributed and dynamic ad hoc RFID resources for the purpose of object localization. In addition, in existing localization systems using RFID technology, there is a lack of a distributed information sharing approach, which ensures timely dissemination of location information among a system's participants. From these the following research questions arise:

**R.1** Can we devise a scheme to estimate the location of RFID tags in a distributed manner?

**R.2** Can we estimate a mobile tags' location in the absence of sufficiently concurrent proximity detection information?

**R.3** How is location information disseminated amongst ad hoc mobile devices/readers?

## 1.2 Thesis Objectives

The goal of the proposed research is to localize and track RFID passive-tagged objects which may represent a person, product, or an animal in dynamic and mobile IoT settings and ensure timely dissemination of location information based on a distributed approach. Our proposed research attempts to answer the aforementioned research questions and devise appropriate solutions, as follows:

A.1 Estimating and keeping track of RFID tags' locations using an inexpensive, reliable, flexible, and distributed approach. To this end, the proposed work investigates utilizing RFID technology in particular to localize objects, studies leveraging the available RFID-based crowdsourcing and proposes innovative distributed localization schemes that overcome IoT challenges (e.g., scalability, object heterogeneity and mobility).

A.2 Enhancing location accuracy when sufficient concurrent spatial information is not available to accurately localize tags, which is common in dynamic and mobile environments. To this end, the proposed work devises a new technique: Time-Shifted Multilateration (TSM). TSM utilizes the available asynchronous spatial information, and based on the estimated tags' speed and time differences, each spatial information is shifted to reflect the expected current location of the tag, providing better accuracy.

A.3 Maintaining availability of location information among a system's participants without causing significant overhead and/or delay. To this end, the proposed work investigates different information dissemination techniques and proposes both proactive and reactive protocols to ensure timely dissemination of location information. In addition, the work studies the possibility of enhancing the performance of distributed dissemination approaches through a distributed infrastructure.

Location estimation refers to the process of determining the most accurate position of an object which can be a stationary and/or mobile (i.e., a person, product, or an animal). The term "tracking" used in this research refers to maintaining location sampling over time. The accuracy of location estimation is measured based on location error, which is the deviation between the actual location of an object and the estimated location. When a query about an object location is initiated, the localization delay is measured as the time the system takes to localize the object of interest and reply back to the query initiator, while the overhead represents the amount of messages generated to disseminate location information or to carry out a location query. In IoT scenarios, applications may have different requirements in terms of localization accuracy and localization delay. Thus, the proposed research work considers the adaptability of both localization and location information dissemination strategies.

## 1.3 Thesis Contributions

In this thesis we first propose two distributed cooperative localization schemes that leverage the available RFID crowdsources to localize surrounding mobile tags. We then devise a TSM technique, which utilizes the available asynchronous spatial information about tags along with the estimated tags' speed to estimate tags' current location. Following, we propose two distributed dissemination techniques to ensure the availability of location information.

The main contributions of this research work can be summarized as follows:

- Leveraging the available RFID crowdsourcing (i.e., heterogeneous and independent mobile RFID readers along with RFID tags' residual memories) in a typical IoT environment to estimate and keep track of locations of passive-tagged objects based on a distributed approach. In this regard, the research proposes two different innovative distributed cooperative localization schemes. The first is through direct cooperation amongst readers in a one-hop neighborhood, which share spatial information about surrounding tags. The second is through indirect cooperation amongst passing readers which utilize tags' residual memory as a focal point to store spatial information about the tag.

- Enhancing location accuracy when sufficient synchronous spatial information is not available to accurately localize passive-tagged objects. The research overcomes this challenge, which is common in dynamic and mobile environments, by devising the new technique named Time-

Shifted Multilateration (TSM). TSM utilizes the available asynchronous spatial information, the estimated tag speed and time differences to provide better accuracy.

- Maintaining availability of location information through different distributed information dissemination strategies along with proactive and reactive protocols. The dissemination strategies work on providing high location information availability among system's participants. The research also introduces a simple, less expensive and flexible infrastructure (i.e., memory spots) that are distributed in the area of interest to be used to disseminate location information and to exchange location queries; providing high location information availability with lower overhead and delay.

## 1.4 Thesis Outline

The reminder of this thesis is organized as follows. Chapter 2 introduces the preliminary background on IoT, RFID technology along with its applications, and object localization problem. Chapter 3 presents two distributed cooperative localization schemes based on RFID crowdsourcing followed by a discussion on some practical implications. Chapter 4 highlights challenges that may affect localization accuracy in dynamic and mobile environments and proposes TSM, which overcomes these challenges and provides better accuracy under different dynamicity settings. Chapter 5 presents and evaluates two distributed location information dissemination strategies: GOSSIPY Pull strategy which requires direct communication amongst RFID readers, and Dissemination Using Memory Spots strategy which disseminates location information with no direct communication amongst RFID readers. Chapter 6 concludes this document by highlighting the main contributions to knowledge proposed in this thesis, showing its limitations and assumptions and outlining potential future research directions. The relationship among the thesis's chapters is further explained in Figure 1.2.

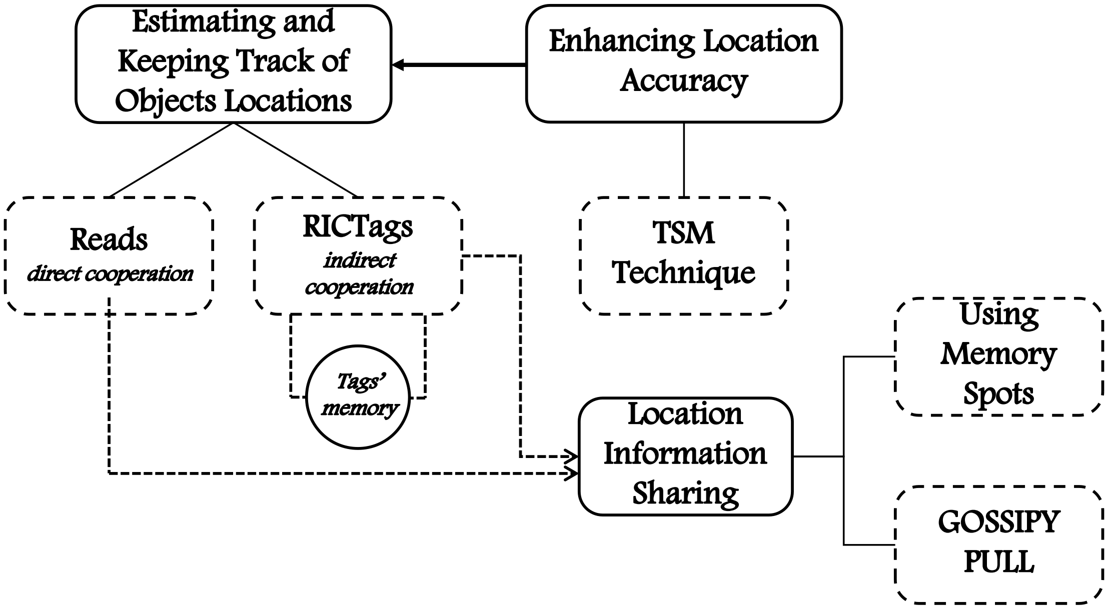


**Figure 1.2: Thesis Components**

# Chapter 2

# Background

In this chapter, we present background material related to the work in this thesis. We start by defining the term IoT in Section 2.1, describe its enabling technologies, and discuss IoT applications. In Section 2.2, we focus on RFID technology as one of the key technologies for IoT, explain RFID system components, and give some examples of RFID applications in different domains. Section 2.3 discusses object localization problem, explains different measurement and location estimation techniques, and shows how mobile anchors can be a solution for dynamic environments when fixed anchors are infeasible.

## 2.1 The Internet of Things: Concept, Technologies, and Applications

The term IoT originated within the RFID development community and pioneered by the Auto ID Labs[1] more than 15 years ago, fosters continued development in academia and industry. Several research efforts have been made to define the IoT concept, predict its future and address its expected challenges [23] - [27]. For example, the work in [23] looks at the IoT as a vast and unexplored research area with no borders in which all current technologies participate with specific roles to provide solutions that are normally ad hoc and distributed. While the work in [24] defines IoT as a world-wide network of a huge number of uniquely addressable objects that are interconnected based on standard communication protocols. The authors of the work in [25] invoke three different IoT definitions defined by the RFID group, the Cluster of Europe

[1] http://autoidlabs.org/

research projects[2], and Forrester Research[3] to give a more user centric definition of the IoT. They defined IoT as an interconnection of sensing and actuating devices which provides the ability to share information across different platforms using a unified framework; creating a common operating environment for enabling innovative applications. In our approach, we define IoT as "*a decentralized system of smart objects which are seamlessly integrated into an information network; continuously providing a variety of smart services over the Internet with support of a set of enabling technologies and communications solutions*". These smart objects are autonomous objects with the ability to interact with their environment, sense and interpret their surroundings and, by means of unique addressing schemes, communicate amongst themselves and to data analytics servers forming pervasive computing environments.

IoT is built upon many technologies which can be broadly grouped into identification, sensing and communication, and middleware technologies [26]. RFID technology is most prominent for the purpose of object identification in IoT where an object, a person and even an animal, can be tagged by a small lightweight tag which holds a unique identifier for that object. RFID is further supported by WSN technology to augment the awareness of the surrounding environments; opening the door for abundant of new context-aware applications [27]. In addition, communication technologies such as WLAN, ZigBee, Bluetooth, and M2M are commonly used in IoT; taking into consideration the different power and communication capabilities of the heterogeneous objects. The middleware is a software layer used to integrate the aforementioned technologies with the application layer; enabling developers to seamlessly develop new services irrespective of the underlying technologies along with the prospective data formats. In the literature, there are several middleware solutions proposed such as Hydra, Ubiware, and Wherex,

[2] http://www.internet-of-things-research.eu/

[3] https://www.forrester.com/home/

which are comprehensively surveyed in reference [28] w.r.t. some functional requirements (i.e., device management, interoperation, platform portability, security and privacy as well as context-awareness). The study in [28] shows that almost all of the middleware solutions focus on device management and do not provide context-awareness functionality which is a core requirement for pervasive and ubiquitous computing.

Applications under the umbrella of IoT span a wide and diverse range of domains which can be broadly classified into five different categories: industrial or enterprise, healthcare, smart infrastructure, social, and security and surveillance [24] - [27]. Figure 2.1 illustrates some application scenarios under each category while more application scenarios can be found in [29] and [30]. Each application has its own challenges and technical issues however there are some common IoT specific challenges that need to be addressed for the vitality of those applications. These challenges include: security, privacy, data integrity and analytics, mobility support, heterogeneity of objects, and scalability. In addition to these challenges, there are technology specific challenges such as architecture, energy efficiency and quality of service.

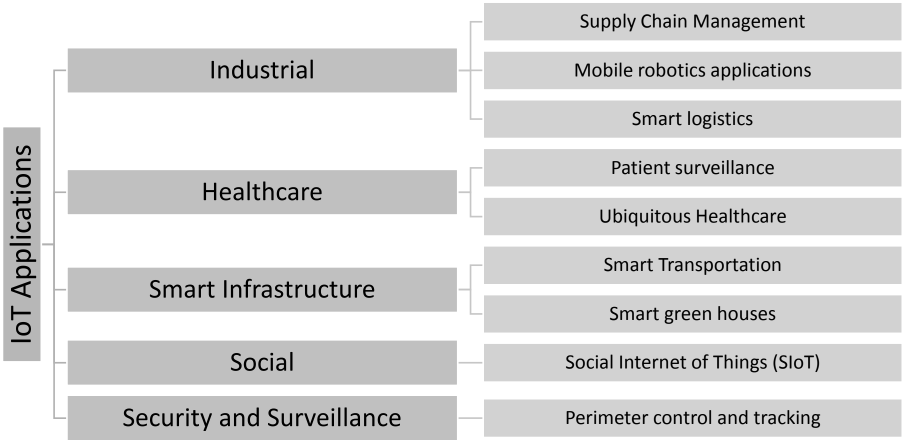


**Figure 2.1: IoT Application domains and relevant scenarios.**

## 2.2 RFID Technology and Applications

RFID technology stands at the forefront of IoT enabling technologies through which “things” are identified from a distance by means of small, lightweight, and inexpensive transponders called tags [14] [15]. Originally RFID was grouped under Automatic Identification (AutoID) technologies, developed to overcome the limitations of the traditional bar code technology [31]. However in recent years, we have witnessed considerable development and technical process in RFID coupled with cost reductions and standardization [16]; resulting in mainstream applications beyond mere identification [17]. In a typical scenario, an RFID system consists of three components: tag, reader/writer device, and application server which interact with each other, as shown in Figure 2.2. As illustrated in the figure, the reader/writer interrogates a tag by broadcasting RF signals through its antenna, the signal is received by the tags’ antenna hence the tag is powered and is able to reply back to the reader/writer by its identifier and possibly by data stored in its memory. This is known in the literature as the backscattering modulation technique [32] which is the standard communication protocol in passive RFID systems that are adopted in our approach. The other class of RFID systems is the active one, where tags are self-powered by means of internal batteries which affect their cost, size, and lifetime. We next explain

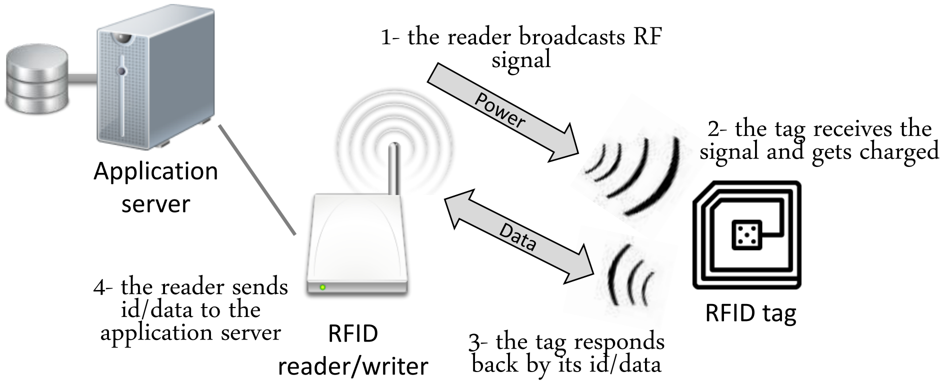

1- the reader broadcasts RF signal
Power
2- the tag receives the signal and gets charged
Application server
Data
RFID tag
4- the reader sends id/data to the application server
RFID reader/writer
3- the tag responds back by its id/data

**Figure 2.2: Typical RFID system components and operation processes.**

the characteristics of passive and active RFID systems along with their unique features, present some applications, and discuss several of the expected challenges and open issues.

### 2.2.1 Current RFID Systems

RFID systems can operate in four different frequencies: Low Frequency (LF, 125-134 KHz), High Frequency (HF, 13.56 MHz), Ultra-High Frequency (UHF, 860-960 MHz), and Microwave (MW, 2.4 GHz and 5.8 GHz) [33]. System operations in LF and HF are based on inductive coupling between the reader and the tag antennas through a magnetic field which results in relatively short reading ranges (less than a meter). These systems are less expensive compared with UHF and MW and usually are used for applications where reading information from short distances is required. In UHF and MW, systems use electromagnetic waves propagating between reader and tag antennas and accordingly have long reading ranges (technically up to 100 meters), which suits a wider range of applications [15] [34]. Under these specifications, however, passive and active systems have their own characteristics in terms of: size, cost, reading ranges, memory capacity, and lifetime which are mainly centered on the type of tags used  which can be either: passive, semi-passive, or active [35] according to its source of power.

**Passive RFID tags**

The passive tag is the least expensive as it harvests energy from the reader based on backscattering modulation. It operates under the constraints that it is within the RF field of the reader, and the power received from the reader is sufficient to power its microchip and to send back information on the same wave. Thus, passive tags typically provide short reading ranges and have less data storage compared to active tags (up to 10 meters) however,  it is inexpensive, tiny, and has almost endless life time; making it the most prominent type of RFID tags today.

**Semi-passive RFID tags**

This type of tags, also named Battery-Assisted Passive (BAP) tag, is powered by a reader for communication (like passive tags), and has an internal battery to power its internal circuits. It has a relatively long reading range compared with a typical passive RFID tag, on-board processor along with off-board sensors such as a thermal sensor, and larger data storage. Since their cost is much higher than passive tags, semi-passive tags are used mainly for costly items that are read over longer distances (up to 30 meters).

**Active RFID tags**

In contrast to passive tags, an active tag has an on-board power supply (i.e., battery) and on-board electronics. The built-in power supply allows the tag to automatically broadcast their signal and work as a Beacon; providing up to 100 meter reading range. The on-board electronics may consist of battery-powered sensors and microprocessors, which allows longer range of communications and higher processing capabilities. Active tags are used in applications that require data collection and processing such as medical equipment, supply chain management of high value products, and animal tracking as part of environmental monitoring.

### 2.2.2 RFID Applications

RFID applications span a range of diversified domains such as logistic, retail, toll system, security, ticketing, location-based services, conferences, exhibitions, and healthcare [14], [36] - [38]. For instance, in supply chain management, goods are identified and tracked from manufacture to their point of sales by means of RFID readers and passive tags. RFID also influences most product delivery applications in which products are tracked from pick up to delivery; relieving incorrect delivery due to human error. In retail, RFID helps in enhancing customer shopping experiences through analyzing which products are picked up frequently by the customer and provide such customers special offers and advertisements according to their preferences. One of the most important payment systems to be automated are the toll systems specially in highways and car parking spots where toll collection should be facilitated to relieve the traffic jam problems occurred in human manned toll stations. In these systems, RFID enables vehicles to automatically do check-in and check-out, and possibly pay the charged fee, under contactless, fast and secure environment. In healthcare RFID is used in health equipment management and in enabling e-health. RFID can aid physicians and support staff in performing their duties such as automating patient admission process, screening and treating processes, and communicating with caregiver teams.

### 2.2.3 RFID Challenges and Open Issues

Although RFID technology influences many domains and is considered a key factor of an abundant number of applications, it has limitations and challenges. One big challenge is the production cost of RFID components which is still not competitive to the traditional labeling technology, hence, even in its cheapest form, RFID cannot be used for low cost products. Another issue in using RFID systems is security especially in passive systems due to the lack of processing capability. RFID, as a wireless technology, is susceptible to security threads such as sneaking, scanning, and authorized reading or writing data on tags which may violate privacy. Thus, authentication protocols and encryption methods must be considered when data is transmitted between a reader and a tag or between a reader and the backend server [39]. Another important issue in using RFID is the standardization protocols and frequencies. Until now, there are no universal standard protocols or formalized frequency use which hinders the global diffusion of RFID technology. In most situations, RFID is implemented internally and it is the responsibility of the manufacturers to design their standards with support of the two main overlapping RFID standardization efforts: ISO and EPC Global. Other technical challenges in RFID arena are: the reading ranges, tag memory capacity, and reader-to-tag and reader-to-reader collision. For the latter, anti-collision algorithms are proposed which reduce the overall reading time and maximize number of simultaneously interrogated tags.

## 2.3 Localization Problem

Localization in general refers to the process of determining the location (physical, symbolic or relative) of an object, which might be stationary or mobile. In the case of a stationary object, its location needs to be determined once. Otherwise, location sampling is required to periodically determine the location of a mobile object which defines the term “Tracking” that might be used in this research. In the following subsections we will give an exhaustive definition of the localization problem, present measuring and positioning techniques found in the literature, along with their characteristics, briefly discuss expected challenges, and show how mobile anchors can be a solution for dynamic environments when fixed anchors are not feasible.

### 2.3.1 Definition and Exhaustive View

Localization is the process of identifying and estimating the location or position of an object based on spatial information or measurements (i.e., distance or angle information) with respect to nodes which have known positions (reference nodes). It can be either network-centric localization or self-localization. In the former, a central unit, named localization server, is used to estimate objects’ position based on spatial information collected from those reference nodes. On the other hand, in self-localization, the position is estimated via the object itself which is responsible of collecting spatial information with respect to reference nodes deployed in the environment and estimates its position accordingly [40] - [42]. The reference nodes, which are called anchors or beacons, and their placement, can significantly affect the localization process. Anchors know their positions by means of either manual configuration (hard coding) or the Global Positioning System (GPS) by fitting them with a GPS receiver, and also they can be mobile [43] - [45]. Objects are called unknown nodes and they are required to know their 2D or 3D positions and possibly their velocity, orientation, etc. in both indoor and outdoor environments. According to application's demand an objects' location can be one of the following types [40] [46][47]:

- ***Physical location***: which identifies a point in 2D (or 3D) by giving its *x*, *y* and *z* coordinates.
- ***Symbolic location***: which expresses a location using a natural-language (e.g., in the office or in the third-floor bedroom.).
- ***Absolute location*** (global): when a common reference grid is used by all objects.
- ***Relative location*** (local): when location is estimated based on proximity to anchors that are not common for all objects. However, if the absolute locations of some anchors are known; relative locations can be transformed into absolute ones [45].

Accordingly, localization and tracking systems are those concerned with localizing multiple objects and keeping track of their locations over time using a wireless technology such as Infrared, Ultrasound, RFID, Ultra Wide Band or Bluetooth [48] - [51]. GPS technology, despite its popularity in outdoor localization, cannot provide accurate localization indoors or dense urban environments due to line-of-sight requirements, presence of obstacles, power consumption, production cost, and object size constraints [41], [44] and [52] - [54]. In fact, this challenging problem attracts significant research interest especially in typical IoT scenarios where objects may scale from millions to billions, large numbers of objects are mobile, and objects are heterogeneous and vary in terms of capabilities and communication characteristics.

In a general, localization schemes can be considered as measuring techniques used to measure some location metrics between unknown nodes and some anchors, followed by a positioning technique. The latter uses the measured metrics to compute the location of the unknown nodes, and optionally refine the nodes' positions to reduce positioning errors. In the following two sections, we explain measuring techniques and positioning techniques and highlight the characteristics of each.

### 2.3.2 Measuring Techniques

The first step in the localization process is to measure some metrics for the node that needs to be localized. In wireless network, these metrics may be distance, angle, or connectivity information. Accordingly, measuring techniques can be broadly categorized as: distance based, angle (or direction) based and connectivity based [40] - [45], [55].

#### 2.3.2.1 Distance-based measuring techniques

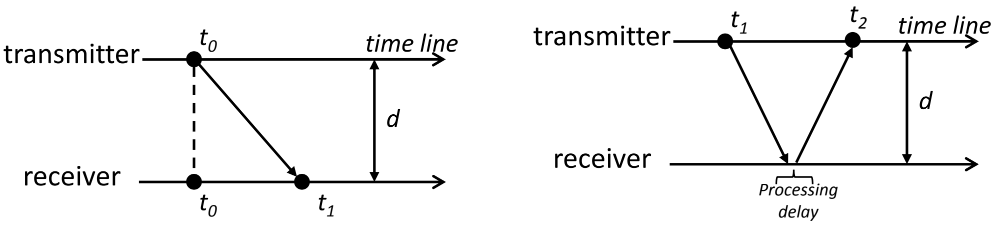
Distance based techniques calculate the distance between a node and anchor(s) which might be highly affected by noise, interference and multipath.

- **Received Signal Strength Indicator (RSSI)**

The foundation of these techniques is based on the existence of RSSI as a standard feature of most wireless devices, which indicates the relative power level of the signal received by a node w.r.t. a certain anchor. RSSI-based techniques use such indicators to estimate the distance between two nodes by relying on the fact that radio signals diminish with the square of the distance from the signals' source [56] [57]. Actually, the signals' propagation is inversely affected by environmental dependent factors such as diffraction, reflection and scattering. Understanding the characteristics of signal attenuation may help in accurately mapping the RSSI to an actual distance. Two mapping models are used for this purpose: analytical model which uses a path-loss propagation model (e.g., electromagnetic wave propagation into space) to map the RSSI to a distance [58], and empirical model in which an RSSI profile is created during the deployment phase and then used to map the RSSI to a distance [59]; giving better distance estimation. Such RSSI profile is created through carrying out experiments as a training and accordingly generate a database of vectors of signal strengths at given sample points (sniffing devices) in the coverage area [60] [61]. This profile, however, is subject to modification in the deployment environment; rendering it a complex and expensive solution. RSSI-based techniques are attractive as they do not require additional hardware nor consume significant amounts of the nodes' battery power.

- **Time of Arrival (ToA)**

ToA-based techniques depend on the one way propagation time of a signal (e.g., Radio Frequency (RF), acoustic, ultrasound, or others) between an unknown node (*receiver*) and an anchor node (*transmitter*) to estimate the distance in between as explained in Figure 2.3 (a). Assuming that the two nodes are highly synchronized, ToA is measured by adding the time of a signal transmission to the time a signal takes to reach the anchor node. This can simply be calculated as the difference between the sending time of a signal at the transmitter and its receiving time at the receiver. Based on the knowledge of signal propagation speed and the time difference, the distance can be calculated as: $d = c_r \times (t_1 - t_0)$ where $c_r$ is the propagation speed of the transmitted signal. The key issues here are the time synchronization and time stamp information, which allow the receiver to accurately estimate the signal arrival time but makes



(a) Time of Arrival (ToA)

(b) Round-trip Time of Arrival (RToA)

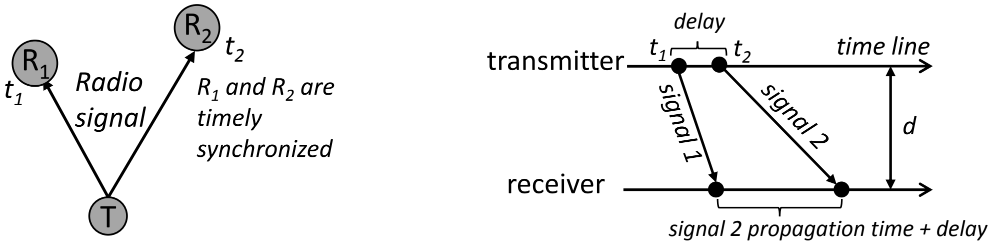
**Figure 2.3: ToA and RToA measuring techniques.**

ToA less attractive and impossible in asynchronous wireless networks [62]. In addition, the type of signal used strongly affects the accuracy of distance calculation. For instance, a signal with low propagation speed and/or large bandwidth such as Ultra Wide Band provides more accuracy than higher propagation speed signals such as Direct Sequence Spread Spectrum (DSSS) [63].

- **Round-trip Time of Arrival (RToA)**

RToA-based techniques avoid the drawback of time synchronization constraints in ToA-based techniques by considering a two way (round-trip) propagation time measured only at the transmitter side as explained in Figure 2.3 (b) [64] [65]. As shown in the figure, time is calculated based on how long it takes to send a signal from a transmitter to a receiver and receiving a reply back. The measured time between the transmission and the reception of the reply at the transmitter is twice the propagation delay plus a reply delay for handling the signal at the receiver which is typically ignored. Thus, the distance can be calculated as: $d = c_r \times \frac{(t_2 - t_1)}{2}$ where $c_r$ is the propagation speed of the transmitted signal. RToA gives better accuracy compared with ToA however the ignored signal processing time at the receiver is considered a major error source in addition to noise, interference and multipath [66].

- **Time Difference of Arrival (TDoA)**



(a) Same signal, 2 synchronized receivers. (b) Two different signals, same receiver.

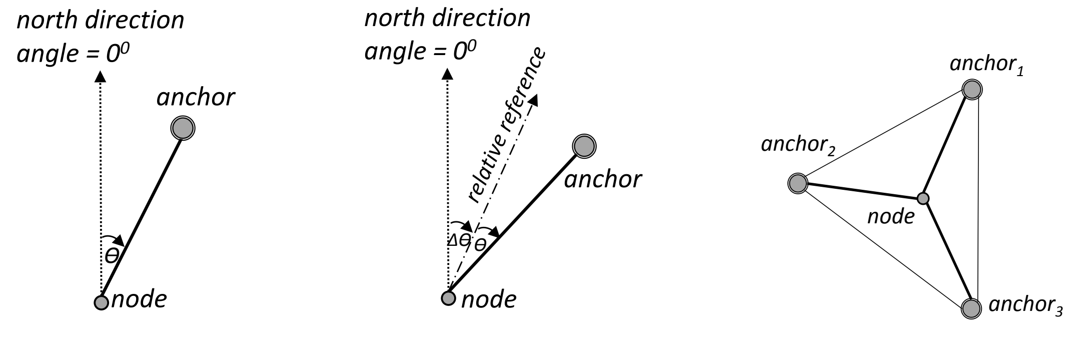
**Figure 2.4: TDoA scenarios.**

Named so because it depends on the difference between the arrival times of the same signal at two time synchronized receivers as shown in Figure 2.4 (a) or between arrival times of two signals with different propagation models at the same receiver as depicted in Figure 2.4 (b) [67]. In the case of multiple signals, a node has to be equipped with a speaker and a microphone which generate signals with different propagation speeds (e.g., ultrasound/acoustic and radio signals) reflecting extra cost. While in the case of multiple receivers (i.e., anchors), no time synchronization between nodes is required as in ToA-based techniques but between the anchors with a tradeoff between anchors' separation and the accuracy. TDoA despite its accuracy suffers from high cost and difficultly in meeting the line-of-sight requirements [68].

2.3.2.2 Angle-based measuring techniques

- **Angle of Arrival (AoA)**

These techniques determine the propagation direction of the received signals with reference to a given orientation or direction [69]. The reference orientation or direction can be either: absolute, relative, or unknown, as shown in Figure 2.5 (a), (b) and (c), respectively. For the first two types of reference orientation, two anchors are enough to estimate the node position based on



(a) Absolute direction (b) Relative direction (c) Unknown direction

**Figure 2.5: AoA: three different reference orientations.**

AoA, however for the latter type, three anchors are required in non-collinear locations. The common approach to determine the angle of arrival is to use multiple antennas where the AoA is computed by analyzing the phase or time difference for the transmitted signals at different array elements. Another approach uses directional antennas and defines AoA by computing the RSS ratio between several well-placed directional antennas such that their main beams overlap. Due to direct line-of-sight constraint of the AoA technique the direction of the antennas along with shadowing and multipath reflections significantly affects the AoA accuracy. Additionally, a major disadvantage of the AoA approach is that it requires additional hardware, which increases the cost and size of nodes to be localized [70][71].

2.3.2.3 Connectivity-based measuring techniques

- **Radio hop count**

This technique relies on the fact that nodes, with communication capabilities, can communicate with one another if the distance between them is less than their radio ranges. By using RSS as a built in connectivity indicator between nodes, a graph of vertices as nodes and edges as connectivity, can be drawn. The hop count between two nodes can be considered as the length of the shortest path between their correspondent vertices in the graph. Hop-count based techniques depend highly on the communication capabilities of nodes as well as their density.

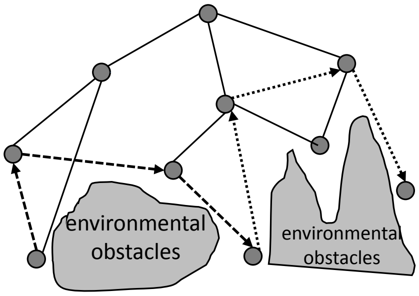


**Figure 2.6: Radio Hop Count approach and anisotropic network situations.**

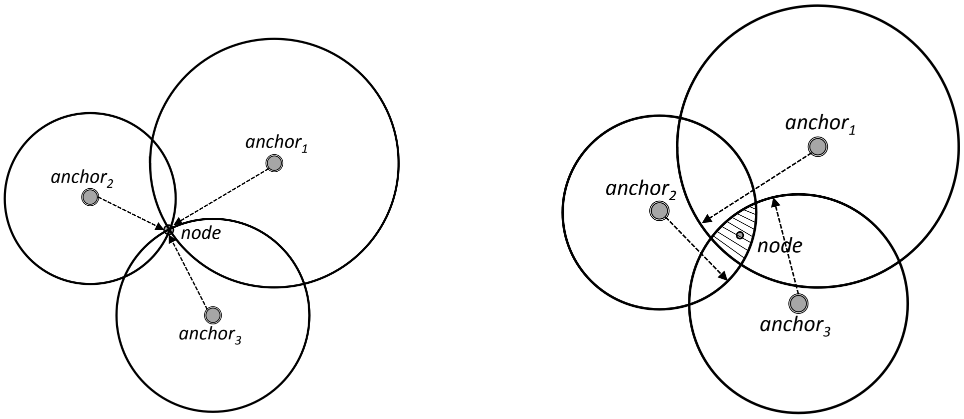
Therefore, these techniques are not applicable for passive nodes and likewise they are not suitable for anisotropic network topologies that contain holes which are unfortunately more likely to exist in practice. The two main demerits of such techniques are: (a) the distance between nodes is always integral multiples of the maximum range of their radios, and (b) the lack of a solution to overcome the problem of the precluded edges in the graph due to environmental obstacles as depicted in Figure 2.6 [44][55].

### 2.3.3 Positioning Techniques

Positioning techniques, also known as localization techniques, estimate the position of an object to be localized based on the measured metrics. In this section, we discuss the main localization techniques that are considered as a base to more advanced techniques: Multilateration using Least Square and Bounding Box which are based on distance metrics, and angulation using Linear Least Square which depends on angle metrics.

#### 2.3.3.1 Multilateration using Least Square

Multilateration technique estimates the position of the object to be localized based on



(a) Ideal case, error free distance measurements. (b) Distance measurements with error.

**Figure 2.7: Multilateration concept in error and error-free distance measurements.**

distance metrics measured by means of RSSI, ToA, or TDoA. In an ideal case where all distance measurements are error free, the position of the object is the intersection point of the circles centered at the anchor position with radius equal to the distance between the object and such anchor as shown in Figure 2.7 (a). Minimum of 3 circles are required for 2D localization and 4 for 3D localization, thereafter in our explanation we consider the 2D localization. The ideal case, however, is not realistic due to the effect of surrounding noise in distance measurement; resulting in noisy measurements as shown in Figure 2.7 (b) which makes the estimation process more challenging.

This problem can be presented as a set of linear equations of the form:

$$(x_i - x)^2 + (y_i - y)^2 = d_i^2$$

Where $x_i$ and $y_i$ are the coordinate of $i^{th}$ anchor, $i \geq 3$, $x$ and $y$ are the coordinate of the object to be localized and $d_i$ is the measured distance between such an object and $i^{th}$ anchor. The resulting linear equation system can be solved using least squares optimization technique [72] [73].

### 2.3.3.2 Bounding Box

One of the distance-based and computationally efficient alternatives to Multilateration is the

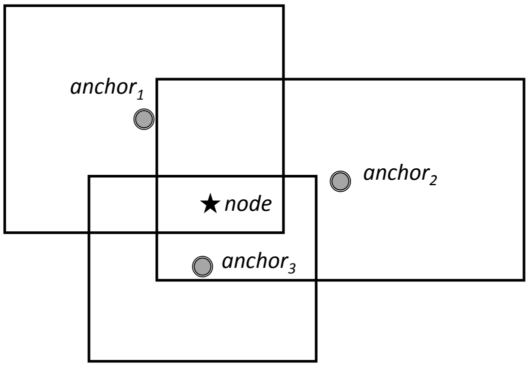


**Figure 2.8: Example of the intersection of three bounding boxes.**

Bounding Box localization technique which is also known as “minmax” algorithm in the literature [74]. Instead of manipulating circles, Bounding Box relies on the intersection of rectangles to estimate the location of an object where the height and width of the $i^{th}$ rectangle (or box) is double the distance measured between the object and $i^{th}$ anchor, the center point of their intersecting bounding box represents the object location as shown in Figure 2.8. Thus, computing the intersection of those bounding boxes is carried out, without use of floating point operations, by taking the maximum of all minimum coordinates and the minimum of all maximums as following:

$$[\max(x_i - d_i), \max(y_i - d_i)] \times [\min(x_i + d_i), \min(y_i + d_i)] \ \forall i = 1,2,..,n$$

It can be noted that the accuracy of Bounding Box technique is less than the Multilateration but on account of the computation cost; rendering such technique more suitable for powerless objects.

### 2.3.3.3 Angulation using Linear Least Square

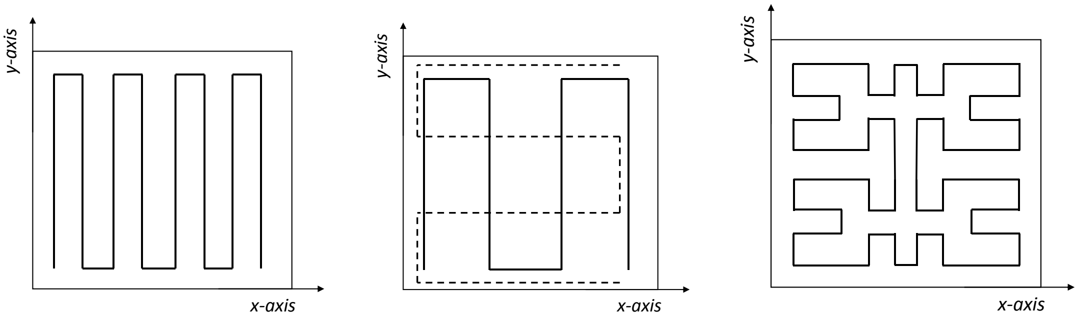
This technique estimates the 2-D position of an object by using at least two angles (or directions) relative to 2 anchors along with their positions instead of the distance in between. Using the angles information and the anchors’ positions, trigonometry laws of sines and cosine are used to calculate the nodes’ position. Arguably that one anchor is used to estimate the position and the second is to confirm. The advantage of this method is that a node position can be estimated using as few as 2 anchors for 2-D and 3 anchors for 3-D with no time synchronization which is not the case as in lateration. However, the disadvantages are that extra and complex hardware is required and the accuracy is degraded as the object moves farther from the anchors.

### 2.3.3.4 Scene Analysis and Proximity Techniques

Some other positioning techniques are scene analysis (or radio map) and proximity [75] [76]. Scene analysis algorithms such as probabilistic, neural networks and support vector machine firstly collect features of a scene (fingerprints) offline, as a training phase, to create a database and then use such database online to locate an object by matching the calculated measurements with the closest fingerprint. RSS is the commonly used fingerprinting method in scene analysis algorithms. The challenge with such techniques is that the database may become unreliable, and requires frequent updates due to changes in the channel and environment. On the other hand, proximity techniques provide a symbolic location of an object based on a dense grid of well-known anchors. If the object is detected by only one anchor, its position will be considered the anchors' location. Otherwise, the object location will be the location of the anchor that receives the strongest signal from such object. Selecting a localization technique is application dependent. For instance, techniques based on angles typically achieve better accuracy than techniques based on distance but at the expense of equipment cost which is a major issue in large scale environments such as IoT.

### 2.3.4 Localization Using Mobile Anchors

The aforementioned localization techniques depend on fixed anchors that are sufficiently deployed in the area of interest in order to have the minimum number of measurements for the objects to be localized. Although this approach is robust and can assure a certain level of accuracy, it results in more expensive infrastructure and less scalability as well. To reduce the number of required anchors while releasing the systems' scalability, the concept of mobile anchors is proposed in the literature and its earlier application was in WSNs [53] and is extended to RFID-based localization techniques [77]. In these techniques, mobile anchors move in the area of interest and periodically broadcast time-stamped beacon packets which contain their IDs and their current locations. Accordingly, objects can localize themselves based on the received beacon packets using one of the localization techniques designed for mobile anchors [78][79].



(a) Scan (b) Double scan (c) Hilbert

**Figure 2.9: Three different mobile anchor trajectories.**

For example, the authors of the work in reference [80] allow mobile anchors to move in random paths based on Random Way Mobility Model (RWMM) [81]. Each object (i.e., sensor in their work) maintains a visitor list containing received beacon packets and their associate lifetime and localizes itself accordingly. However, the random mobility may result in poor performance in terms of time and accuracy as shown in reference [82] which fostered researchers to introduce the concept of predefined trajectory and to send beacon packets based on specific frequencies. Important remarks regarding the characteristics of the predefined trajectories and how frequent mobile anchors should broadcast beacon packets are discussed in [83]. The authors suggest planning the mobile anchor trajectory such that all objects are covered by at least three anchors per a time unit and the trajectories are as tight as possible for accurate localization. Based on their study, three predefined trajectory techniques (Figure 2.9 (a) – (c)) are proposed in reference [84]. In the first, the mobile anchor moves in parallel to either x-axis or y-axis as explained in Figure 2.9 (a) where the distance between any two parallel paths is double the communication range of the mobile anchor. This technique is simple and easy to implement however as mobile anchors move in straight lines, the collinearity problem raised which strongly affects the localization accuracy. This problem is avoided in the second technique by allowing mobile anchors to move in both directions as shown in Figure 2.9 (b), because of their travel distances more energy will be consumed. In the third technique (Figure 2.9 (c)), the area to be covered is divided into 4 squares where a mobile anchor connects the 4 sub-areas using $4^n$ as explained in the figure; providing better results in terms of localization accuracy compared with the other two techniques. Other techniques are proposed based on the aforementioned techniques such as circle and s-circle [85] which show better accuracy with same mobile anchor travel distance while s-curves [86] achieves same accuracy as of circles and s-circles but with less travel distance. Other researches depend on the ability of objects to forward messages received from mobile anchors to their neighbor objects; allowing them to localize themselves even if they are few hops away of mobile anchors. However, it is noted that all these techniques assume that objects to be localized are deployed and remain static over time and those objects are not passive and have some processing capabilities. These assumptions are valid and common in WSN applications, not in IoT dynamic environments. In IoT environments, however, objects typically are passive objects with very limited communication and computation capabilities.

The trend of using mobile anchors instead of using fixed and expensive infrastructure is also exported to RFID-based localization techniques [87][77]. In these techniques, mobile RFID readers move and detect surrounding tagged-objects and localize them with support of well-known deployed reference points named landmarks, where the localization process typically takes place on a backend server. These techniques still depend on a kind of fixed infrastructure (i.e., landmarks) but it is less expensive compared with a full-cover network of fixed readers. Despite the cost reduction achieved by such techniques, they still follow a centralized approach and strongly depend on landmarks which are active or passive tags that know their locations by manual configuration in addition to other operation constraints such as number of concurrent mobile readers. We remark that scalable RFID-based localization techniques using mobile readers is still an open research area.

## 2.4 Summary

In this chapter, we give an introduction about IoT and its main components. We follow by giving an overview about IoT enabling technologies, highlighting some of the IoT applications, and discussing the challenges that result from the ad hoc nature of IoT environments. The chapter provides background about RFID technology as a key player in IoT, presents different RFID systems along with their strengths and limitations, overviews some of current applications, challenges, and open issues. In addition, we take the advantage of this chapter to give an extensive background about localization problem, illustrate its two main building blocks: measuring techniques and positioning techniques, and explain each block by presenting their most dominant techniques. We give a glimpse on using mobile anchors for localization for cost reductions and flexibility and illustrate how this concept, as yet, is a hot of research area especially in RFID-based localization.

# Chapter 3

## Estimating and Keeping Track of Objects Locations

As mentioned earlier RFID technology has unique features: , non-line-of-sight communication, cost and power efficiency, high data rate and security which sparked the extensive use of RFID systems for object localization and tracking [77], [88] - [91], [94] - [101]. Existing RFID systems follow a centralized and coordinated approach in which readers communicate with a central server. In this chapter, we advocate the use of RFID crowdsourcing through the heterogeneous and independent mobile RFID readers along with RFID tags' residual memories, in object localization.

We propose two innovative distributed cooperative localization schemes to estimate and keep track of locations of passive-tagged objects based on crowdsourcing in indoor/outdoor environments. First, we propose Reader Direct cooperation System (ReaDS) [92], in which localization service is provided through direct cooperation amongst readers in a one-hop neighborhood which share spatial information about surrounding tags under time constraints. Second, we propose Reader Indirect Cooperation through Tags memory (RICTags) system [93], in which the readers do not have to directly communicate with one another but indirectly cooperate through utilizing the tags' residual memory as a focal point to store spatial information. RICTags releases communication overhead in large crowdsourcing environments (e.g., sport activities, conferences and fairs) where users, represented by mobile readers, do not have to communicate or know of each other. We remark that this approach is fundamentally different from existing tag localization techniques as it leverages RFID crowdsourcing in a distributed scalable approach, as opposed to the existing fixed centralized infrastructure.

The remainder of this chapter is organized as follows: Section 4.1 presents the related work and shows the motivation behind devising ReaDS and RICTags systems. Systems' components, notations and assumptions are given in Section 4.2. The exchanged information between system entities is explained in Section 4.3. Section 3.4 and 3.5 illustrate in details the operation of ReaDS and RICTags respectively, while Section 3.5.3 discusses practical implication along with user authentication and privacy issues and concludes the chapter.

## 3.1 Related Work and Motivation

The localization principles discussed in Chapter 2 can be applied to RFID systems [18]. Accordingly, several RFID-based localization systems have been proposed in the literature, which can be broadly categorized into reader localization and tag localization [13][19].

In reader localization systems [88], [94] - [97], typically a large number of active and/or passive tags are deployed at known locations in the area of interest to represent landmarks for mobile objects. Each mobile object, which is equipped with an RFID reader, estimates its location based on the connectivity information with those landmarks. For instance, in [88], the authors attach reference tags to the floor and ceiling into a square pattern to localize a mobile reader using the weighted average method and a weighting function. While in [94], the same approach is followed but the accuracy of localization is enhanced by rearranging the reference tags into a triangular pattern. However, in such systems, the required number of reference tags is relatively high. To avoid the dense deployment of reference tags, SLAC-RF [97] proposes specialized tags named supertags. Each supertag is an array of RFID tags which are arranged to simulate a virtual antenna array. A mobile reader which navigates the area estimates its position using the phase difference of received signals with respect to supertags along with inertial navigation system (INS) measurements. As well, the authors of [95] and [96] propose localization methods based on the geometric knowledge of the identification region in 3D space to provide a finer degree of localization. However, the work in [96] considers the fault frequency in localization and proposed a quality index to measure the quality of localization results.

In tag localization systems [77], [89] - [91], [98] - [101], an infrastructure of RFID readers, which detect tags and report detection information to a central server for location estimation, is used. For example, based on the inter-tag distances at multiple fixed readers, systems in [90] and [91] have been proposed where SpotON [90] is the pioneer. SpotON estimates tag location using an aggregation algorithm; ignoring measurement uncertainty caused by the environment dynamics. LANDMARC [98] uses an RFID reader infrastructure along with reference tags to calibrate the environment dynamics. By comparing the Received Signal Strength (RSS) from the targeted tag with those of reference tags, the server estimates the tag location based on the locations of the k-nearest reference tags. Improvements to LANDMARC were proposed in [99] for reference tags placement and their contribution to tag location estimation. VIRE [100] and LVIRT [101] use virtual reference tags instead of a dense deployment of reference tags. In such systems, the RSS readings of each virtual reference tag for each reader are calculated using those of surrounding real reference tags. For instance, VIRE calculates the RSS of each virtual reference tag using the RSS of the surrounding reference tags and a linear interpolation algorithm. Then, it compares a tags' RSS to that of reference tags either real or virtual, identifies all plausible locations and filters them using an elimination algorithm. An attempt to localize tagged objects using mobile readers is proposed in [77] with support of landmarks which are active or passive tags distributed randomly and know their locations by manual configuration. The reader-tag distance and tag-landmark distance are then used to estimate the tag location based on analytical method. The system depends on the availability of two concurrent readers, landmarks and a probabilistic RFID map-based technique which handicap the systems' scalability.

Reader localization systems are inherently distributed and provide good accuracy through a cost effective infrastructure. However, they suffer from the high cost of associating an RFID reader with every object, rendering such an approach infeasible for IoT settings. On the other hand, although tag localization systems cater to a wide range of applications, the centralized and fixed infrastructure-based systems provide limited scalability and may not be a practical solution for IoT settings especially in outdoor environments.

In a parallel scope, Tile[1] which is a Bluetooth tracker is introduced to localize personal items by attaching it to everything that needs to be localized and is supported with an app designed for this purpose. Although Tile's objective is to localize everyday items and keep track of them, it has some limitations as a Bluetooth-based device. First, it costs $25USD for each one, compared with just pennies apiece for RFID passive tags which hinder its usage in IoT applications. Second, Bluetooth consumes more energy for communications and its devices use less life time batteries compared with RFID passive tags which work with no battery. Last, Tile is larger in size and weight compared with RFID passive tags which come with a variety of forms to suit different applications.

In typical IoT environments there is abundance of distributed RFID resources that can be utilized in order to provide object localization based on crowdsourcing which, as yet, are not utilized. Towards this end, in this chapter we devise an RFID-based scheme to estimate the location of RFID passive tagged-objects in a distributed manner based on RFID crowdsourcing. For this purpose, we devise two alternative distributed cooperative localization schemes based on RFID crowdsourcing. The schemes take advantage of the following: (1) objects can be easily identified by passive RFID tags, which are inexpensive and widely available, (2) embedded RFID readers in mobile devices are being rapidly adopted due to the great interest of RFID

[1] https://www.thetileapp.com/

manufacturers, along with the rapid advancements in antenna design for handheld RFID readers [20] and (3) RFID tags are capable of storing data in addition to their unique identifiers [16] [22]. The first scheme uses direct cooperation amongst readers in a one-hop neighborhood which share spatial information about surrounding tags and the second deploys indirect cooperation amongst passing readers, which utilize the tags' residual memory as a focal point to store spatial information about such tag. The goal of the proposed systems is to provide a localization service indoors and outdoors within large scale dynamic environments based on crowdsourcing where deploying and maintaining a fixed central infrastructure for localization is expensive or infeasible. The following sections describe the schemes' components, notations, assumptions, and explain in detail the operation of the two proposed localization schemes.

## 3.2 Components, Notations and Assumptions

Given an RFID system of *n* passive tagged-objects and *m* mobile readers, we consider a two-dimensional localization problem of the passive tagged-objects using the mobile readers in an indoor/outdoor area. Accordingly, we define the following components, which are partially or fully used based on the operation of each proposed scheme:

***Tags*** – representing the objects to be localized. These objects can be either stationary or mobile and are identified by passive RFID tags. The number of *Tags* is much larger than the *Readers* in the scenarios under study.

***Readers*** – representing the mobile RFID readers in the area, which are predominantly dynamic, heterogeneous, and uncoordinated. In addition to object identification, readers are also used in localizing objects of interest in the environment. Such *Readers* may be the smartphones or handheld RFID readers.

***Detection table*** – represents a table containing temporal and spatial information about interrogated *Tags* with respect to the *Readers*.

***Location table***: represents a table containing time-stamped estimated locations of interrogated *Tags* based on the available detection information.

The location and manipulation of *Detection table* and *Location table* depend on the localization scheme in use and their attributes are given in Section 3.3

Thereafter in the chapter, we use the following notations to refer to the aforementioned components as well as other design parameters in explaining each schemes' operation:

- $T = \{t_1, t_2, t_3, \ldots., t_n\}$ is the set of *n* mobile or stationary *Tags*.
- $R = \{r_1, r_2, \ldots., r_m\}$ is the set of *m* mobile *Readers*.
- $CR_i = \{cr_{i1}, cr_{i2}, \ldots., cr_{ik}\} \subseteq R$ is a subset of the mobile *Readers* that cover a tag $t_i$ at time $\tau$.

- $NRi = \{nr_{i1}, nr_{i2}, \ldots, nr_{il}\} \subseteq R$ is a subset of the mobile *Readers* that are in neighborhood of reader $r_i$ at time $\tau$.
- *D_int* is a time interval every which each $r_i \in R$ detects *Tags* in its vicinity and updates the *Detection table* accordingly.
- *Loc_int* is a time interval every which each $r_i \in R$ estimates the locations of the detected *Tags* based on spatial information in hand (i.e., the available valid records in *Detection table*).

*Typically, D_int < Loc_int.*

- $D(t_i) = \{d_1(t_i), d_2(t_i), \ldots, d_k(t_i)\}$ is the set of $k$ records in *Detection table* w.r.t. a tag $t_i$; representing the spatial information measured by a subset of $R$ within a specific time interval. Each element $d_k$ *in* $D(t_i)$ is represented by $d_k.t$, $(d_k.x, d_k.y)$ *and* $d_k.r$, which are interrogation time, $x$ and $y$ coordinate of the reader position at time of interrogation, and the tag to reader distance respectively (see Table 3-1).
- $Loc(t_i) = \{loc_1(t_i), loc_2(t_i), \ldots, loc_l(t_i)\}$ is the location set of a tag $t_i$; representing the location history of $t_i$. Each element $loc_l$ in $Loc(t_i)$ is represented by $loc_l.t$, $(loc_l.x, loc_l.y)$ and $loc_l.LAI$ which respectively are time of location estimation, $x$ and $y$ coordinate of the estimated location and number of detections used in location estimation (see Table 3-2).

*The set $D(t_i)$ and $Loc(t_i)$ are ordered chronologically.*

The *Readers* are assumed to be capable of acquiring their absolute positions at any given time using one of the positioning systems for mobile readers (e.g., GPS, WiFi, anchors, etc.), and they are authorized to interrogate all *Tags* in the given environment. In our first scheme, ReaDS, we assume that *Readers* can reach neighboring *Readers* to share *Tags*' spatial information. In the second scheme, RICTags, we assume that *Readers* do not directly communicate with one another, but are authorized to update *Tags*' memory.

## 3.3 Exchanged Information

During the operation, two types of information are created: Detection information, which is created by *Readers* w.r.t. *Tags* and is filled in *Detection table* with attributes illustrated in Table 3-1. Detection information is built during the tags identification process which is carried out every *D_int* and used later in location estimation. Location information, represented in *Location table* contains the estimated locations of *Tags* and is updates by *Readers* every *Loc_int* (see Table 3-2). Each location is identified by its estimation time and a Location Accuracy Indicator (LAI) which represents the number of detections positively contributing to the tag location estimation and enhances the location accuracy. The table size of *Detection table* and *Location table* is of order (*mn*).

**Table 3-1: DETECTION TABLE**

| *Field* | *Description* | *Notation* |
|---|---|---|
| time | The time at which a reader *r* detects tag $t_i$ and creates a detection record. | $d_k.t$ |
| tag ID | The interrogated tag ID | $d_k.TID$ |
| position | The 2D position of the reader *r* at time of interrogation, it is represented by *x*, *y* coordinates. | $(d_k.x, d_k.y)$ |
| distance | The tag to reader distance, measured by means of RSS, time difference of arrival , angle of arrival,  etc. | $d_k.r$ |

**Table 3-2: LOCATION TABLE**

| *Field* | *Description* | *Notation* |
|---|---|---|
| time | The time at which a reader *r* estimates the location of tag $t_i$ based on the available detection information. | $loc_l.t$ |
| location | The estimate location of $t_i$, it is represented by *x*, *y* coordinates. | $(loc_l.x, loc_l.y)$ |
| LAI | Number of detections used by *r* to estimate the location of $t_i$. | $loc_l.LAI$ |

## 3.4 ReaDS System

Considering a snapshot of the distributed dynamic environment, typically we have a subset of *Readers CRi* covering a tag $t_i \in T$ and *Readers* can reach neighboring *Readers* for the purpose of information sharing. Based on direct cooperation amongst *Readers*, ReaDS operates in two interleaving processes. Each mobile reader periodically: (1) interrogates *Tags* in its vicinity and exchanges detection information with *Readers* in the one-hop neighborhood, (2) estimates *Tags* locations based on collected and exchanged detection information and maintains time-stamped location information about surrounding *Tags*. Exchanging detection information between the one-hop neighbors is proposed to improve location estimation from only proximity to more accurate location information. The following subsections elaborate the system's interleaving processes.

### 3.4.1 Detection Information Collection and Sharing

In this process, *Readers* maintain their detection table while they are moving. At every *D_int*, each mobile reader $r_i$ in *R* interrogates tags in its proximity, creates a detection record in its *Detection table* for each detected tag, and shares such records with all readers in the one-hop neighborhood $NR_i$ as explained in Figure 3.1

Each detection record is marked with a *flag*. A value of 0 indicates first hand detection information to limit its sharing among the one-hop neighboring readers only as illustrated in Figure 3.2. The number of detection records is limited to a time window which is equal to the *Loc_int* after which replacement takes place. Given that *D_int* < *Loc_int*; each Loc_int may hold multiple time-stamped detection records for the same tag either from a reader itself or from a reader in *CRi*. This in turn provides more accuracy and decreases the probability of collinear detections [102]. To accommodate mobile tags, we can acquire more detection records within each localization interval by decreasing the *D_int* with respect to *Loc_int*.

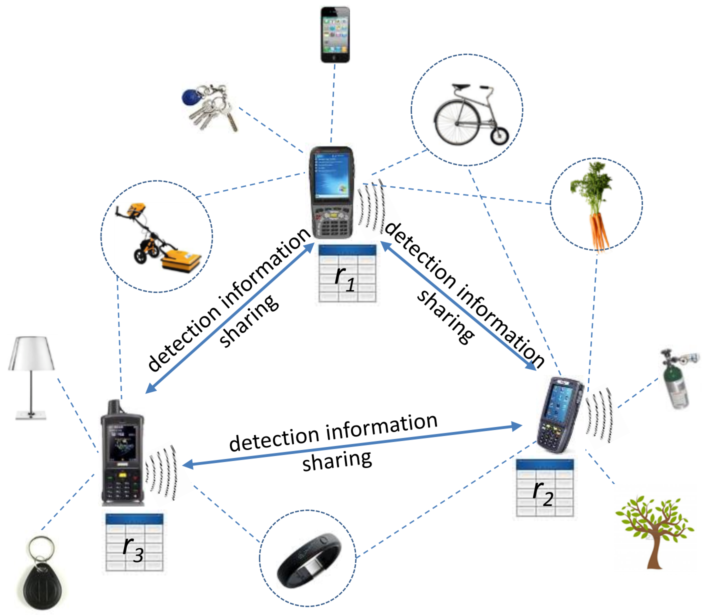
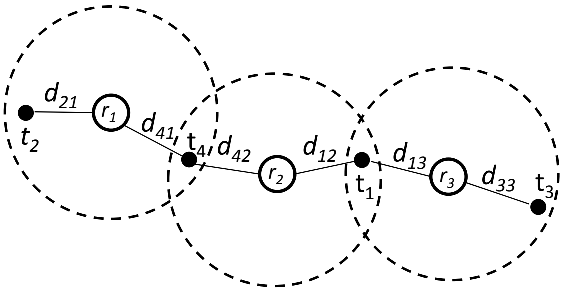


**Figure 3.1: Detection information collection and sharing.**



Detection information at reader $r_2$

| Time | Tag ID | Reader ID | Reader position | T-R distance | Flag |
|---|---|---|---|---|---|
| .. | $t_4$ | $r_2$ | .. | $d_{42}$ | 0 |
| .. | $t_1$ | $r_2$ | .. | $d_{12}$ | 0 |
| .. | $t_4$ | $r_1$ | .. | $d_{41}$ | 1 |
| .. | $t_1$ | $r_3$ | .. | $d_{13}$ | 1 |
| .. | $t_2$ | $r_1$ | .. | $d_{21}$ | 1 |
| .. | $t_3$ | $r_3$ | .. | $d_{33}$ | 1 |

**Figure 3.2: snapshot explains detection information sharing.**

### 3.4.2 Location Estimation

In case of stationary tags, typically most detection records positively contribute to localization accuracy. However, due to the probability of tag mobility, not all detection records are valid for localization. As a preprocessing step for location estimation, detection records should be filtered to exclude those that may negatively affect the localization accuracy, taking detection time into consideration. To do so, we consider each detection record as a circle, which is centered at reader position and has radius equal to the distance from the reader to the tag of interest. Starting with the most recent circle and follow with others one by one, we exclude circles that do not contribute to the intersection area of all previous circles. This eliminates the case where an incorrect/outdated detection is included, resulting in less accurate location. Otherwise, the common lateration or multilateration technique is used according to the number of valid detections. The location estimation process is illustrated in Algorithm 3.1, and the filtering steps

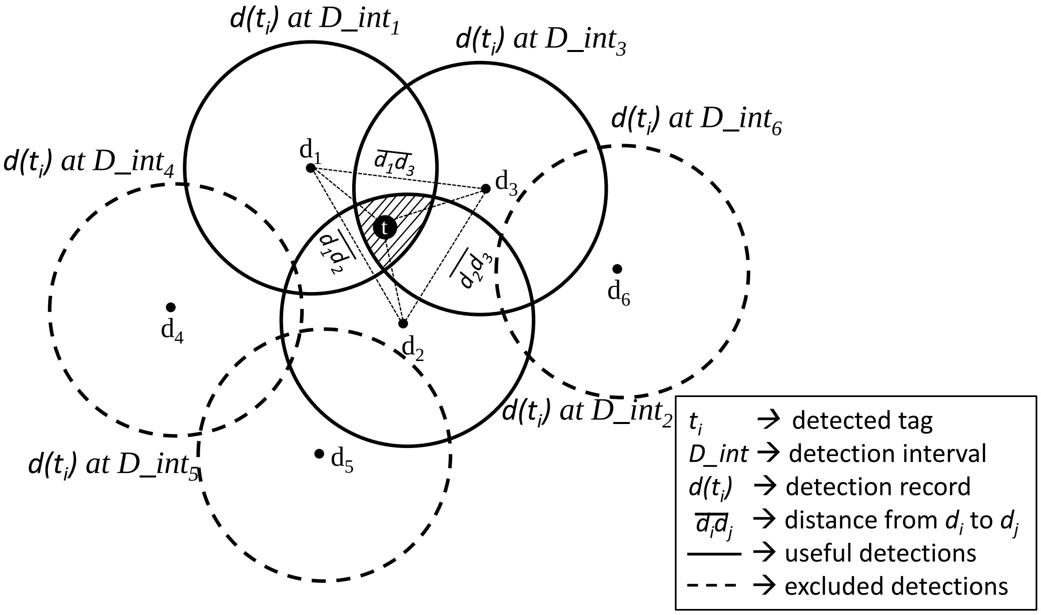


**Figure 3.3: Example of detections filtering.**

(3-18) are depicted in Figure 3.3 for only one tag for simplicity. In the figure, the dotted circles are excluded since they negatively contribute to the location accuracy.

At the update *Location table* step in Algorithm 3.1, the number of valid detections used in localizing a tag is added to the location information record as a location accuracy indicator *LAI*, which is used to decide which location is more accurate within the same *Loc_int* in the event of conflicting results.

**Algorithm 3.1**: location estimation

**Input:** *Detection table* **Output:** *Location table*

```
for each Loc_int do
  for each tag $t_i$ in Detection table do
    for each detection $d_j$ in $D(t_i)$ do      //filter negatively contributed detections
      if j = 1 then
        filtered_detect_info_list.add ($d_j$)
      else
        to_add_flag = True
        for each $d_k$ in filtered_detect_info_list do
          if distance between (($d_j.x$, $d_j.y$), ($d_k.x$, $d_k.y$)) > ($d_j.r + d_k.r$) then
              to_add_flag = False
              break
          end if
        end for
        if (to_add_flag = True)
         filtered_detect_info_list.add ($d_j$)
        end if
      end if
    end for
    $t_i$ .position = Estimate_Loc (filtered_detect_info_list) //using nonlinear least squares method
    Update Loc($t_i$) → (Get (current time), $t_i$.position, filtered_detect_info_list.size))
  end for
end for
```

## 3.5 RICTags System

*Readers* may not know or communicate with one another. To accommodate such cases, we propose using the *Tags* as the focal point for storing and exchanging detection and location information; allowing an indirect cooperation amongst *Readers* through the *Tags*' residual memory. In RICTags, the *Readers* periodically: (1) detect *Tags* in their interrogation zones and write detection information on the interrogated *Tags*' memory and (2) retrieve detection information obtained from passing *Readers*, estimate *Tags*' locations accordingly and update the *Tags* location information. Figure 3.4 shows the general framework of the RICTags system. We next explain the tags notification and tags localization processes.

### 3.5.1 Tags notification

*Tags*' memories are updated by passing *Readers*. The objectives are: (1) maintain detection records on the tag memory to be used by other passing *Readers* for tag localization and (2) allow a tag to know its estimated position at every *Loc_int*. For the former, in every *D_int* each *Reader* $r_j \in$ R interrogates the *Tags* in its proximity. For each successfully identified tag $t_i$, $r_j$ creates a

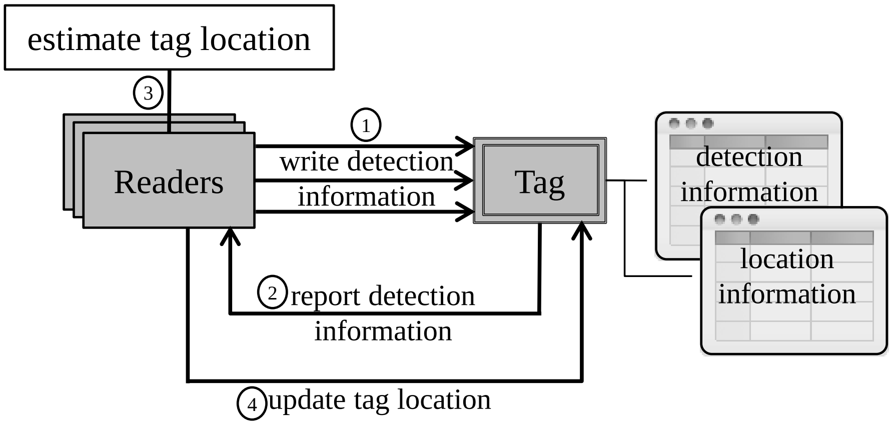
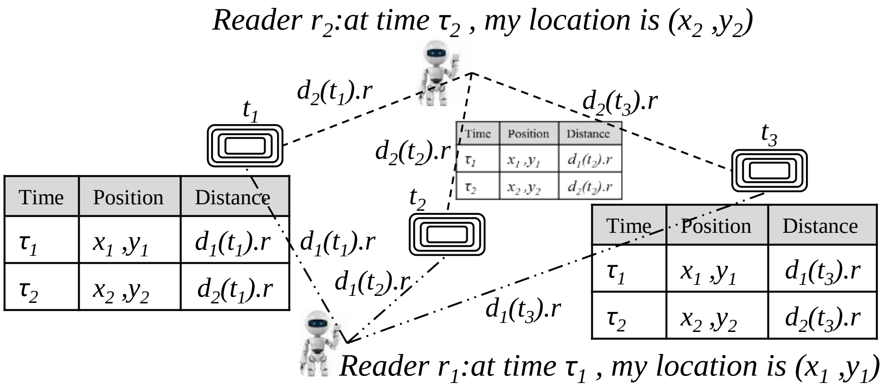


**Figure 3.4: General framework of RICTags system.**


Reader $r_2$:at time $\tau_2$ , my location is ($x_2$ ,$y_2$)
$d_2(t_1).r$
$t_1$
$d_2(t_3).r$
$t_3$
$d_2(t_2).r$
$t_2$
$d_1(t_1).r$
$d_1(t_2).r$
$d_1(t_3).r$
Reader $r_1$:at time $\tau_1$ , my location is ($x_1$ ,$y_1$)

| Time | Position | Distance |
|---|---|---|
| $\tau_1$ | $x_1$ ,$y_1$ | $d_1(t_1).r$ |
| $\tau_2$ | $x_2$ ,$y_2$ | $d_2(t_1).r$ |

| Time | Position | Distance |
|---|---|---|
| $\tau_1$ | $x_1$ ,$y_1$ | $d_1(t_2).r$ |
| $\tau_2$ | $x_2$ ,$y_2$ | $d_2(t_2).r$ |

| Time | Position | Distance |
|---|---|---|
| $\tau_1$ | $x_1$ ,$y_1$ | $d_1(t_3).r$ |
| $\tau_2$ | $x_2$ ,$y_2$ | $d_2(t_3).r$ |

**Figure 3.5: Snapshot of tags notification process**

detection record and writes such record into the memory of $t_i$ (see Figure 3.5). As shown in the figure, updating the tags' memory by a subset of *R* allows the tag to hold multiple time-stamped detection records which are limited to either the tags' memory or the time window of the *Loc_int*. If the tag is static, most of these detection records positively contribute to localization accuracy. However, in case of a mobile tag, a time constraint should be considered when localizing the tag, to effectively ignore outdated records with respect to the *Loc_int*, every *Loc_int Readers* update tags location information with information resulted from running the tags localization process.

### 3.5.2 Tags localization

The Tag localization process takes place every *Loc_int*, *Readers* interrogate surrounding tags, fetch their detection information, then estimate, and update the tags' location information accordingly. Algorithm 3.2 lists the tag localization process, where it is assumed that *Tags* and *Readers* are stationary. This may also correspond to a snapshot of the dynamic *Readers* and *Tags* case. In Algorithm 3.2, the detection records are processed first to filter out the outdated records with respect to *Loc_int* (lines 5-9). Then the remaining detection records are filtered to exclude detections that do not positively contribute to the intersection area of the more recent detections similar to Algorithm 3.1 (lines 10-20). The two sequenced filtrations may result in only one detection record, resulting in less localization accuracy. Otherwise, the Multilateration technique is applied and the number of used detections is added to the location record as the *LAI;* the *LAI* can be used to decide which location is more accurate in case multiple locations are estimated within a same *Loc_int*. The tag location information is then updated.

**Algorithm 3.2**: Tags localization Algorithm

**Input:** detection information **Output:** location information

```
for each Loc_int do
    for each $t_i$ in my proximity do
        set Detect_info($t_i$) = get $t_i$ .D($t_i$)
        set filtered_info($t_i$)
        for each record $d_j$ in Detect_info($t_i$) do    //filter outdated detections
            if $d_j$.time < current time – Loc_int then
                Detect_info($t_i$).delete($d_j$)
            end if
        end for
        for each record $d_j$ in Detect_info ($t_i$) do    //filter negatively contributed detections
            if $j = 1$ then    filtered_info($t_i$).add($d_j$)
            else
                for each $d_k$ in filtered_info($t_i$) do
                    if $d_j$ do not intersect with $d_k$ then
                        Detect_info($t_i$).delete($d_j$) and break
                    end if
                end for
                filtered_info($t_i$).add ($d_j$)
            end if
        end for
        set LAI = filtered _info($t_i$).size
        $t_i$ .position = Estimate_Loc (filtered_info($t_i$)) //using nonlinear least squares method
        Update Loc($t_i$) → (Get (current time), $t_i$.position, filtered_info($t_i$).size)
    end for
end for
```

The Tag notification process can result in accumulated detection information, which may be outdated after the *Loc_int*. To release *Tags* resources, *Readers* periodically delete this outdated detection information along with location information and maintain only the *S* most recent locations. The parameter *S* is application dependent and determines the location history maintained in each tag for other purposes such as tracking.

### 3.5.3 Use Case Scenario

We assume that mobile devices adopted a request/reply strategy via apps designed for localization service. In this strategy, a wireless device broadcasts a query asking for the location of tag(s) of interest. Each *Reader* receiving this query interrogates such tag(s) in its vicinity, retrieves its location if it exists and replies back to the requestor. If the tag does not exist, the *Reader* ignores the query. If the interested wireless device does not receive a response within a certain timeout, it initiates another location query.

Suppose that Tom plans to attend a fair that came to town with his active young son Max. Upon his arrival, he receives a notification on his mobile device indicating that he has the option to contribute to a participatory localization service at the Fairgrounds. Tom likes the idea as he is interested in keeping track of Max. So he accepts the notification, hence, an app is installed on his mobile device along with supportive quick help. Also, he is instructed to pick up a wristband RFID tag from the site administration for Max. A considerable number of participants have the same interest as Tom, thus they participate in the localization service as well. For the sake of illustration, we define the following potential types of actions that take place in the system:

- **Action A**: A mobile RFID reader interrogates a tag and writes such detection into the tags' memory.
- **Action B**: A mobile RFID reader interrogates a tag, fetches detection information from the tag memory, localizes the tag accordingly and writes the estimated location into the tags' memory.

- **Action C**: A mobile device broadcasts a query asking about location of certain tag(s).
- **Action D**: A mobile RFID reader receives a location query, triggers Action B with respect to the tag of interest and replies to the requestor.

Figure 3.6 depicts several locations and events over a time window of Tom's activities. Within this time window, there are 7 mobile RFID readers contributing to localization service including Tom's mobile device. At location 1, $r_1$ and $r_3$ executed a type A action in relation to Max's tag. At location 2, another Action A was taken by $r_4$; consequently Max's tag holds three detection records. When Tom and Max were at location 2, a science show attracted Max so he moved to location 3 to enjoy it without Tom.

While Max was enjoying the science show, $r_2$ conducted Action A while $r_5$ conducted Actions A and B. When performing Action B, $r_5$ uses the detection records to localize Max (at location 3) based on detection records created by itself, $r_2$, $r_3$ and $r_4$ (by now the detection from $r_1$

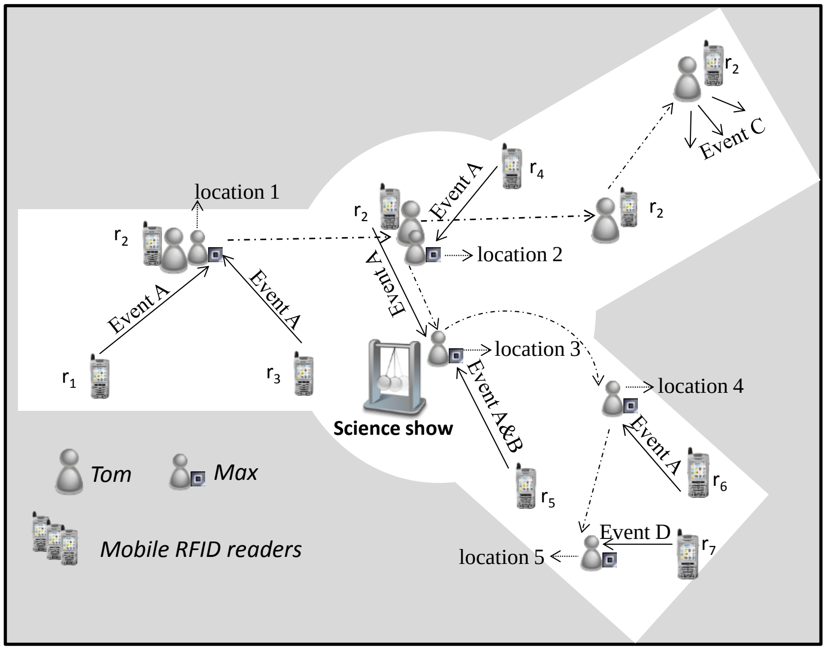


**Figure 3.6: Fair use case scenario.**

is outdated.) After a while, Max discovered that he was lost so he started running toward where he thought his father would be, but unfortunately, it was in the wrong direction. Tom did not realize that, thus he followed his path as shown in Figure 3.6. When Max reached location 4, Action A was taken by $r_6$. At the same time, Tom realized that his son was not around; he used his mobile device and carried out Action C with respect to Max's tag. During this time Max moved from location 4 to location 5. $r_7$ carried out Action D, which includes Action B as well. In Action B, $r_7$ uses the available detection records to localize Max (at location 5). Tom received a message from $r_7$ indicating that Max was now at location 5. Very relieved, Tom then rushed to this location for Max.

At the end of the day, Tom and Max decided to go home. At the exit gate, he received a message indicating that his mobile device was unregistered from the localization service and the app may then be uninstalled, releasing any resources on Tom's mobile device.

## 3.6 Practical Implications and Conclusion

This chapter proposes leveraging the available RFID crowdsources in typical IoT settings for the purpose of object localization through two different distributed cooperative schemes. Those RFID crowdsources are represented in the passive tags used to identify objects and the ad hoc heterogeneous and distributed mobile RFID readers in a given IoT environment. Both schemes depend on the ability of the mobile RFID readers to interrogate surrounding tags while being able to acquire their locations. The first requires the direct cooperation amongst the mobile readers assuming that readers can reach neighboring readers for information sharing, while the second scheme carries out the cooperation indirectly by means of the tags' residual memory assuming that readers are authorized to update a tags' memory. From the practical perspective, three points arise: (1) the availability and prevalence of such mobile RFID reader, (2) the capability of passive

RFID tags to hold information in addition to their identifiers, and (3) the authority of the mobile RFID readers to interrogate and update tags' memory while maintaining privacy. Four categories of RFID technology exist: low frequency (LF), high frequency (HF), ultra-high frequency (UHF), and microwave frequency (MW). Of the four passive RFID technologies the preferred is UHF, due to the advantages of long-range read and write, rapid identification, and high memory capability. We next address the aforementioned three points with respect to the UHF RFID systems, justify our assumptions and highlight potential challenges.

### 3.6.1 Mobile RFID Reader Considerations

Due to the great interest by RFID manufacturers  to compete in this rapidly growing market and the rapid advancements in antenna design for handheld RFID readers [20] and [21], there are abundant models of mobile readers in the market which are compatible with the passive ultra-high frequency EPC Class1 Gen 2 protocol [16]. From a technical point of view, these mobile readers can be categorized into two types: small and lightweight UHF RFID reader module to be embedded into mobile devices such as smartphones or PDA's; turning them into mobile RFID readers[2'3], and a portable handheld UHF RFID reader with different memory, communication, and computation capabilities[4]. Both types are supported with Wi-Fi, Bluetooth, and possibly GPS receivers which allow them to acquire their location outdoors using GPS-based positioning with typical accuracies of 1-3 meters and indoors using other wireless technologies such as Wi-Fi or Ultra Wide Band (UWB) with meter-level accuracy [103][104]. Typically, these readers are enriched with an operating system such as Microsoft Windows Mobile 6.5; allowing console and/or mobile applications to be developed to provide different RFID-related services including

[2] IDBLUE product datasheet accessed online http://idblue.com/rfid-readers/uhf-rfid-reader
[3] Nokia 5140 RFID Reader - RFID Journal accessed online http://www.rfidjournal.com/articles/view?5509
[4] product datasheet accessed online http://www.imsinc.ca/wp-content/uploads/Specification-Sheet-MC9190-Z-10_121.pdf

localization. In addition, readers have large memory capabilities that can be extended up to 32 GB in some models, which is enough to hold spatial and location information about interrogated tags. These readers are also capable to communicate with each other through a variety of built in communication capabilities such as WLAN and Bluetooth.

The reading ranges of RFID readers on mobile devices are relatively shorter than those of handheld readers since the antenna for such mobile devices are typically small and lightweight. However, the continued penetration of mobile devices along with the rising demand for RFID reader products for those devices fosters researchers and RFID manufacturers to invest more into developing antennas with reasonable size and weight that increase reading ranges [105] - [108].

### 3.6.2 Passive RFID tag Consideration

Passive RFID tags are battery-free devices which are energized by RFID readers and respond by reflecting energy back to the reader through backscattering modulation technique. According to this technique, a tags' antenna used to collect power and transmits or receives signals along with the sensitivity of the tags' Integrated Circuits (IC) strongly affects the reading and writing ranges. For instance, passive tags operating in UHF (856 MHz to 960 MHz frequency) with the latest generation of silicon IC have a reading range up to 10 meters while they cost just pennies apiece[5]'[6]. In addition, there is on-going research in the literature [109] - [113] towards designing antennas for RFID UHF passive tags, which is a crucial element for better performance (e.g., the tag antenna design discussed in [112] demonstrates up to 25 meter line of sight reading range).

A key factor in RICTags system is the availability of a relatively large rewritable memory on passive tags. As specified in EPC Global Class-1 Gen-2 standard [16], UHF RFID passive tags'

[5] http://www.aliexpress.com/item/Gen2-Passive-RFID-UHF-Tags-with-read-range-Max-10meters/1977823602.html

[6] product datasheet accessed online http://www.xtremerfid.com/spec-sheets

memory consists of four different areas named banks (*bank0, bank1, bank2, bank3*). Table 3-3 shows each memory area along with its description and utilization. As seen from the table the rewritable user memory can hold information up to 32KB in some tags[7]; making such tags capable of decentralized data storage for RFID distributed systems [114].

In the tag interrogation process, readers identify surrounding tags based on their EPC tag ID's stored in their memory, while in the writing process, a reader uses the tag ID to write data singly into the user memory of such tag. There is no standard to date to store data on the user memory however two data formats are suggested: Comma Separated Values (CSV) and Extended Markup Language (XML) data format [115] which are suitable even for tags with limited user memory. Each spatial information or location information can be represented as a set of attributes separated

**Table 3-3: UHF RFID Passive Tags Memory**

| *Bank#* | *Description* | *Utilization* |
|---|---|---|
| *bank0* | Reserved memory | It stores kill password used to disable the tag and access password used to lock and unlock the tags' read/write capabilities. It uses *32* bit per password and it cannot store information besides these two passwords |
| *bank1* | EPC memory | It is a minimum of 96 bits of writable memory used to store the Electronic Product Code and optionally can be extended only on account of user memory |
| *bank2* | TID memory | It store the unique tag ID number generated by the manufacturer and cannot be changed |
| *bank3* | User memory | It exists in certain tags to store user information for different purposes. No standards for number of bits used however it is usually 512 bits and some high memory tags have up to 32K bytes of memory |

[7] Bacheldor, B. (2009). "Tego Launches 32-Kilobyte EPC RFID Tag". Accessed online http://www.rfidjournal.com/articles/view?4578

by commas while using a special flag to distinguish between the two types of information For example, if the tag is located at time 8:02 in the position (1.006, 2.402) using 3 detections, the piece of information written to its user memory using CSV data format is: [*1, 08:02, 1.006, 2.402, 3*] where the first character indicates that this tuple is location information. By considering only the most recent 20 records of location information, it takes few hundreds bytes from the tags' memory.

### 3.6.3 User Authentication and Privacy Considerations

In our approach, we assume that mobile RFID readers are allowed to interrogate and localize surrounding tags (i.e., identified objects and people) and possibly update their residual memories in a distributed manner, which may raise some security issues such as user authentication and privacy and data integrity [116]. In the context of security and privacy in UHF passive RFID systems, a variety of protocol-based solutions are surveyed in [117] which follow one of two directions: constructing an RFID security protocol that is compatible with tags' constraints [118] [119], or defining privacy models for RFID systems [120]. As a protocol-based solution for our approach, scalable light-weight tree-based category of privacy preserving authentication (PPA) protocols [121] or advanced encryption standard (AES) protocols [122] can be adopted. In such protocols, authenticated keys and their hashed values can be stored in the memories of both tags and readers, under secure channel, during user's registration for the localization service. During the system operation, both readers and tags authenticate each other by matching the received hashed value of the key to the one stored at their memories. Although the implementation of such protocols on passive tags is challenging due to the power consumption, the study in [123] shows that this solution is practical even on low powered passive tags. The issues of user authentication and privacy are the subject of further research.

# Chapter 4

# Enhancing Location Accuracy in Dynamic and Mobile Environments

Location estimation refers to the process of determining the most accurate position of an object using the available spatial information measured by anchors or by the object itself. Location's accuracy is measured based on location error, which is the deviation between the actual location of an object and the estimated location. In both the ReaDS and RICTags localization systems proposed in Chapter 3, we adopted the Multilateration method as the technique for object location estimation. Multilateration is a commonly used technique that estimates object location based on the intersection of all plausible areas (i.e., spatial information) where the object is expected to exist. This technique along with other lateration-based localization techniques assumes that the measured spatial information, even those from mobile anchors, is sufficient and obtained simultaneously for each object. This assumption, however, may not be reliable in a typical dynamic environment where the anchors are mobile ad hoc RFID readers typically with short reading ranges. Three challenges arise in this case: (a) insufficient spatial information, (b) non-intersecting spatial information, or (c) the intersection may not reflect the object's real location. As a result, the difference between the actual and the estimated location may be significant. This chapter addresses ways to overcome these challenges and provide better location accuracy in the absence of sufficient concurrent readings.

In this chapter, we propose the technique Time-Shifted Multilateration (TSM), which utilizes the available asynchronous spatial information to enhance objects' location estimation. In TSM, each entry of spatial information is shifted, based on the estimated speed and time differences of the object, to reflect the expected current spatial information of the object; providing better accuracy. The strength of TSM technique is that it can be combined with any other distance-based localization techniques to enhance mobile object location estimation when there is not sufficient spatial information at a specific time. We analyze the properties of TSM technique and investigate its performance through extensive simulations using ns-3 with respect to *average speed error* and *average location error* metrics, which respectively represent the variance between the object actual speed and the estimated speed and the average of Euclidean distance between the actual location of an object and its estimated location (i.e., actual error).

The remainder of this chapter is organized as follows: Section 4.1 reviews related work and provides the motivation behind devising the TSM technique. Notations and assumptions are given in Section 4.2. TSM technique is detailed in Section 4.3 which gives an overview of TSM's operation, illustrates the object's speed estimation method and explains the time-shifting process with Multilateration. The TSM properties, including upper bounds on location accuracy are studied in Section 4.4, followed by simulation results and analysis in Section 4.5. The chapter is concluded by Section 4.6.

## 4.1 Related Work and Motivation

As explained in Chapter 2, localization process starts with measuring some spatial metrics, which we refer to as spatial information, for the object which needs to be localized. This spatial information might be distance, angle or network connectivity information with regard to some anchors which know their positions, and is used to estimate the object(s) location via positioning techniques such as lateration, angulation and DV-hop [124] [125]. These positioning techniques use the measured spatial information to compute the location of an object and optionally refine the object's location to enhance location accuracy. In this approach, we focus on distance-based spatial information along with Multilateration as a positioning technique, anchors are mobile RFID readers and objects are RFID passive tagged-objects.

Hereinafter, each record of spatial information is considered as a circle in 2D centered at the reader position at interrogation time with the radius equal to tag to reader distance, measured by means of RSS, time difference of arrival or angle of arrival according to readers' capabilities. To localize an object in *d*-dimensional space using Multilateration, at least *d+1* anchors are needed concurrently to estimate the object location which is a challenge in large scale ad hoc mobile networks.

Research in similar areas such as WSNs focus on how to reduce the number of required anchors to localize nodes (i.e., objects) for such networks. For example, in [126] and [127], the authors propose using iterative multilateration in which localized nodes in one iteration act as anchors in the following iteration; resulting in error accumulation due to the uncertainty in upgraded anchors positions. To decrease such an error, the authors in [128] propose filtering upgraded anchors based on their location uncertainty and exclude those of large uncertainty and range error. Another solution to the challenge of anchor less availability is range-free multilateration-based localization (RFML) techniques such as [129] - [133]. RFML techniques estimate distance between a node and faraway anchors based on hop-count instead of depending on absolute point-to-point distance estimation; releasing the requirement for a relatively large number of anchors. These solutions are typically proposed to localize stationary nodes and when extended to mobile nodes, the location estimation accuracy is aggressively affected. Prediction-based localization techniques [134] - [139] are mostly used for mobile object tracking. In this thesis, however, we tackle the problem of passive object localization based on actual and estimated measurements.

Iterative- and range-free multilateration-based localization techniques depend on the communication capabilities of nodes to capture network connectivity information with surrounding neighbors; rendering them unviable if we take into consideration passive nodes such as passive RFID tagged-objects. On the other hand, prediction-based localization techniques may impose significant storage and computational requirements on nodes to be localized which rule out their application in less powerful or passive nodes which are typical in IoT environments.

In this chapter we address the problem of accurately localizing passive mobile object when the available concurrent spatial information about objects is not sufficient. To this end, we devise the TSM technique to enhance location accuracy in dynamic and mobile environments specifically when: (1) anchors are relatively few with no guarantee to have enough anchors concurrently covering each object, (2) objects to be localized are passive (do not actively engage in or initiate communication) and (3) computational complexity is a concern.

## 4.2 Notations and Assumptions

We consider a group of passive RFID tagged-objects and a group of dynamic RFID readers (both are mobile[1]). While they move, readers are allowed to interrogate surrounding tags hence, generate time-stamped spatial information about interrogated tags. The time-stamped spatial information generated by a subgroup of readers is then used to estimate the tags' location and update tags' location history accordingly. The tags' location history is used to estimate a tags' speed which in turn is used to enhance tags' location estimation over time through a proposed time-shifting process.

For consistency, we use the following notations (as defined in Chapter 3):

[1] Mobility can be based on one of mobility models for mobile ad hoc networks such as Graph-Based Mobility Model (GBMM) [140] or Random Way Mobility Model (RWMM) [81].

- $T = \{t_1, t_2, t_3, ....., t_n\}$ is the set of *n* passive tags; representing objects to be localized.
- $R = \{r_1, r_2, ....., r_m\}$ is the set of *m* readers; representing the ad hoc mobile readers (i.e., anchors).
- $D(t_i) = \{d_1(t_i), d_2(t_i), ..., d_k(t_i)\}$ is the detection set of a tag $t_i$; representing the spatial information measured by a subset of *R* within a specific time interval. Each element $d_k$ *in* $D(t_i)$ is represented by $d_k.t$, $(d_k.x, d_k.y)$ *and* $d_k.r$, which are interrogation time, *x* and *y* coordinate of the reader position at time of interrogation, and the tag to reader distance respectively. The set $D(t_i)$ is ordered chronologically.
- $Loc(t_i) = \{loc_1(t_i), loc_2(t_i), ..., loc_l(t_i)\}$ is the location set of a tag $t_i$; representing the location history of $t_i$. Each element $loc_l$ in $Loc(t_i)$ is represented by $loc_l.t$, $(loc_l.x, loc_l.y)$ and $loc_l.LAI$ which respectively are time of location estimation, *x* and *y* coordinate of the estimated location and number of detections used in location estimation denoted by $|D_{used}(loc_l)|$. The set $Loc(t_i)$ is ordered chronologically.

In our approach, we consider time to be discrete and assume that readers are synchronized time wise and can acquire their own locations at any given time. Readers can access the detection set $D(t_i)$ and location history $Loc(t_i)$ of the interrogated tag $t_i$ through information sharing or via utilizing the tags' residual memory as explained in Chapter 3. As in typical localization schemes, each element $d_k$ in $D(t_i)$ is prone to two sources of errors: reader position and tag to reader distance errors. If the error in distance measurements for $d_k$ is negative, the object real location is outside the circle defined by $d_k$. and the object is not guaranteed to be in the intersection of all $d_k \in D(t_i)$. However, the study in [141] shows that within a set of distance measurements, the negative errors cannot be arbitrarily large negative. This allowed the authors of the work in [142] to propose a technique that converts all errors in distance measurements to be positive. In our approach, we assume that tag to reader distance error, denoted $\varepsilon_k$, is positive. We next explain the TSM technique and detail the tag speed estimation and the time-shifting processes.

## 4.3 Time-Shifted Multilateration (TSM) Technique

### 4.3.1 TSM Operation Overview

The TSM technique takes two inputs: asynchronous spatial information during a specific time window, $D(t_i)$, and a tag location history, $Loc(t_i)$, and works as follows (see Figure 4.1). First, if the tag has no previous estimated locations, TSM considers an initial tag speed based on the attributes of the mobile object it is attached to (e.g., walking speed for pedestrians). Otherwise, TSM uses $Loc(t_i)$ to estimate the tag speed using an exponentially weighted moving average. Second, TSM performs a time-shifting process, TSM enlarges each detection, $d_k$ in $D(t_i)$, based on the both the tag speed and the time difference between the detection and the time of location estimation; resulting in a synchronized detection set, denoted $D_{sync}(t_i)$. Last, TSM applies Multilateration to $D_{sync}(t_i)$ to estimate tag location. Figure 4.2 illustrates an instance of the TSM technique and shows how the time-shifting process takes place for 4 detections. The shifted

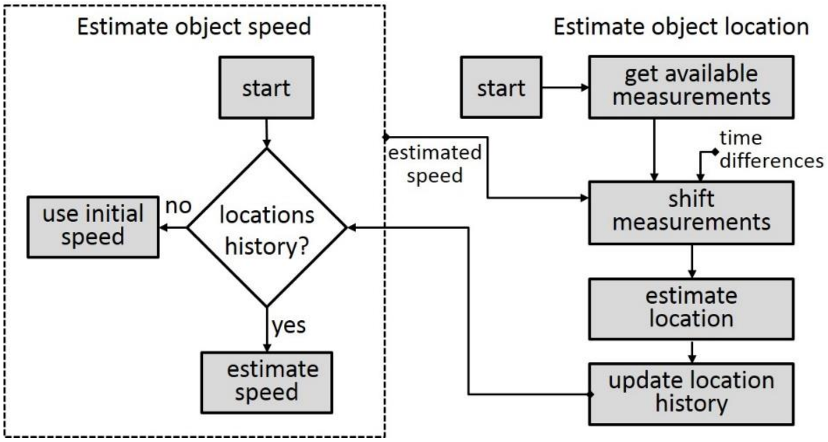
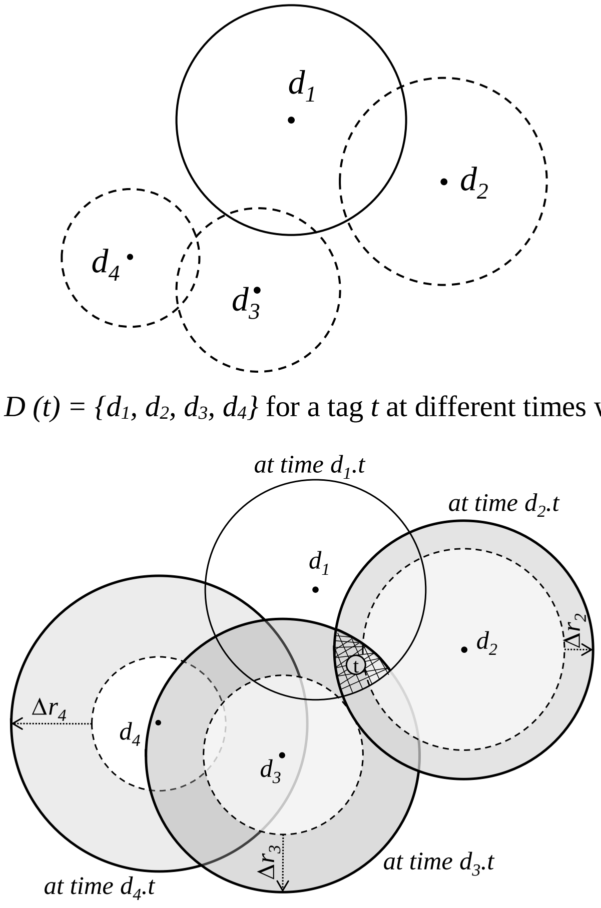


**Figure 4.1: TSM operational framework.**

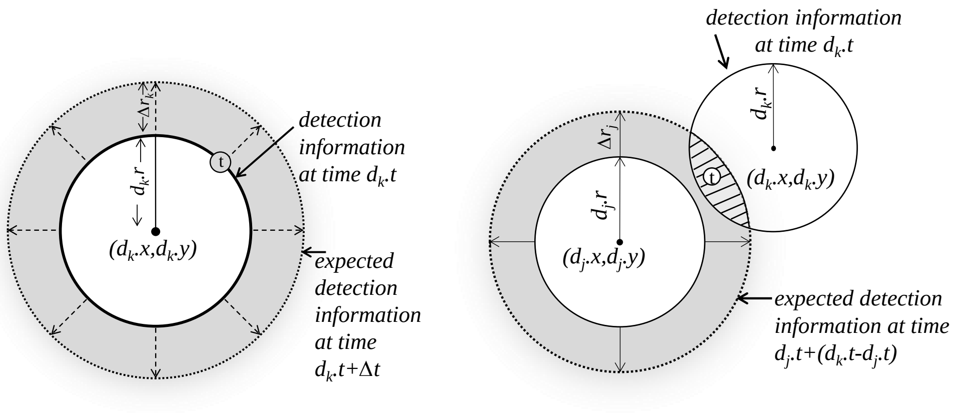
distance $\Delta r_i$ for each detection represents the maximum distance a mobile tag can travel when it moves using the estimated speed during its time difference. We next validate the positive effects of time-shifting on location accuracy. Knowing the speed of a mobile tag, the tag can be localized at time *t* using detection information from time *t*-Δ*t*. Accordingly, we can establish the following theorem.



(a) The detection set $D\ (t) = \{d_1, d_2, d_3, d_4\}$ for a tag *t* at different times where $d_1$ is the recent.

(b) The set of shifted detections $D_{sync}(t)$ after expanding $d_2$, $d_3$, $d_4$ by $\Delta r_2$, $\Delta r_3$ and $\Delta r_4$ respectively, the tag *t* is expected to be in the shaded area.

**Figure 4.2: TSM technique operation: illustrative example.**


$\Delta r_k$
$d_k.r$
t
detection information at time $d_k.t$
$(d_k.x,d_k.y)$
expected detection information at time $d_k.t+\Delta t$
detection information at time $d_k.t$
$d_k.r$
$\Delta r_j$
t
$(d_k.x,d_k.y)$
$d_j.r$
$(d_j.x,d_j.y)$
expected detection information at time $d_j.t+(d_k.t-d_j.t)$

(a) time-shifting of one detection after *Δt* (b) How time-shifting affects location accuracy.

**Figure 4.3: The concept of time-shifting process.**

*Theorem1:* A mobile tag $t_i$, which is localized by a detection $d_k(t_i)$ = { $d_k.t$, $(d_k.x,d_k.y)$, $d_k.r$ }, can be localized after time $\Delta t$ by a detection $d'_k$ = { $d_k.t+\Delta t$, $(d_k.x,d_k.y)$, $d_k.r+(s * \Delta t)$}, given its speed $s$.

*Proof:* Given the mobile tag speed $s$, the maximum distance a tag can travel during a period $\Delta t$ is $\Delta r = (s * \Delta t)$. So if the tag is localized by the detection $d_k$ as shown in Figure 4.3 (a); the worst case is when the tag is located at a point on the circumference of the circle at time $d_k.t$ and moves perpendicularly outside the circle. Considering the maximum distance $\Delta r$, if the tag is detected in a circle centered at $(d_k.x,d_k.y)$ and has a radius $d_k.r$; after the period $\Delta t$, the tag cannot reach a point outside the circle centered at $(d_k.x,d_k.y)$ and has a radius $d_k.r+\Delta r$. ■

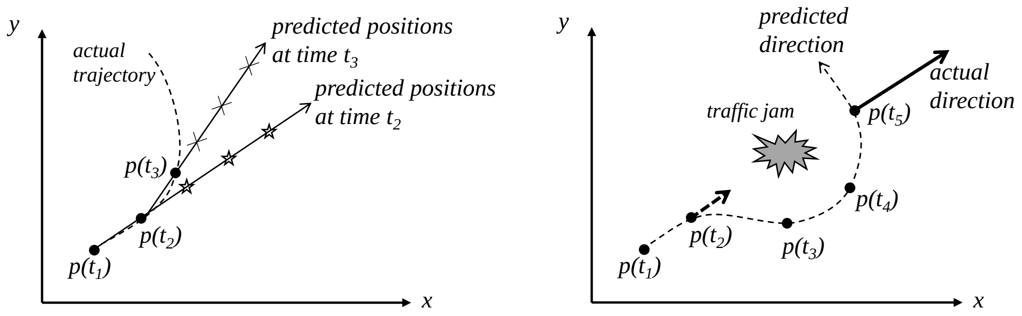
*Theorem2:* A mobile tag, which is localized by detections: $d_k$ = { $d_k.t$, $(d_k.x,d_k.y)$, $d_k.r$ } and $d_j$ = { $d_j.t$, $(d_j.x,d_j.y)$, $d_j.r$ } such that $d_k.t$ is more recent than $d_j.t$, is expected to be located in the area of intersection between the circle centered at $(d_k.x,d_k.y)$ with a radius $d_k.r$ and the circle centered at $(d_j.x,d_j.y)$ with a radius $(d_j.r + s*(d_k.t-d_j.t))$, given its speed is $s$.

*Proof:* If the tag is localized by the detection $d_k$ as shown in Figure 4.3 (b) then at time $d_k.t$, the tag is located at an arbitrary point in the circle centered at $(d_k.x,d_k.y)$ with a radius $d_k.r$. According to *theorem 1*, the tag is also located at an arbitrary point in the circle centered at $(d_j.x,d_j.y)$ with a radius $(d_j.r+s*(d_k.t-d_j.t))$. Thus, such an arbitrary point would be in the area of intersection between the above mentioned two circles. *(Theorem 2* can be generalized for any number of detections.) ■

The TSM technique takes into consideration the average tag speed *s* irrespective of the direction of its trajectory. TSM accounts for the worst case in which the mobile tag is expected to move exactly away of reader position; giving an upper bound to all possible movement directions. This allows the TSM technique to perform consistently under different mobility models (i.e., movement patterns) and relaxes the inadequacy of linear prediction for nonlinear movement as explained in Figure 4.4.

### 4.3.2 Object Speed Estimation

TSM starts updating a tags' speed after estimating two or more locations for the tag. With



(a) Failure of linear prediction (reproduced from [145])

(b) Inadequate Prediction (reproduced from [146])

**Figure 4.4: Examples for prediction problems under different mobility patterns.**

two previous locations in hand, we measure the tag speed based on the traveled distance between them; assuming speed is constant between each consequent location as its variation is insignificant under low speed movements [143]. For three or more previous locations, methods such as Kalman filter [134] and exponentially weighted moving average (EWMA) [144] can be used to estimate the tag speed. In this work, we adopt the EWMA approach because of its simplicity and low computation cost. If there are no previous estimated locations for the tag to be localized, an initial speed is assumed though (e.g., pedestrian speed).

*Definition 1* (distance between two locations): Given two consequent locations for tag $t_i$: $loc_j(t_i)$ and $loc_{j+1}(t_i)$, the distance between the two locations is the Euclidean distance between $(loc_j.x, loc_j.y)$ and $(loc_{j+1}.x, loc_{j+1}.y)$, denoted as $dist(loc_j,loc_{j+1}) = ||loc_j - loc_{j+1}||_2$

The traveling speed of tag $t_i$ from $loc_j(t_i)$ to $loc_{j+1}(t_i)$ is:

$$speed_{j+1}(t_i) = dist(loc_j,\ loc_{j+1}) \,/\, (loc_{j+1}.t - loc_j.t) \qquad (1)$$

where $j = 1...k\text{-}1 \mid k$ is the number of previous estimated locations for tag $t_i$. Given $k\text{-}1$ successive speeds for tag $t_i$ such that $speed_{k-1}(t_i)$ is the most recent one, the exponentially weighted moving average speed for $t_i$ can be computed using the following equation:

$$EMAspeed_{j+1}(t_i)= \alpha * speed_{j+1}(t_i ) + (1\text{-}\ \alpha) * EMAspeed_j(t_i) \qquad (2)$$

where $\alpha$ is a constant factor between 0 and 1, which controls the rate of coefficients decreasing; attenuating the contribution of older speeds to the estimated speed. Thereafter in the chapter, we consider α as $(1/k\text{-}1)$.

Uncertainty in previous tags' estimated locations may significantly affect the accuracy of its speed estimation thus, previous locations set $Loc(t_i)$ can be filtered prior to speed estimation based on a predetermined threshold $\Theta$ in terms of the number of detections positively contribute to location estimation (i.e., $|D_{used}(loc_l)|$ which is detailed in next section). This threshold can be selected after normalizing the values of $|D_{used}(loc_l)|$ for all previous locations however for simplicity, in this approach such threshold is set to *3*. To understand the uncertainty of location information and its impact on speed estimation, consider the example illustrated in Figure 4.5. As shown in the figure, the tag *t* is localized over time at three sequent locations, $loc_1$, $loc_2$ and $loc_3$, however the location accuracy indicator (*LAI*) of $loc_2$ was *1* which means it is a proximate location with an upper-bound error equal to the radius of detection in hand (i.e., $d_k.r$, see Section 4.4. for proof) which contributes twice to object speed estimation. We next show and proof the constraint at which discarding such locations provides better object speed estimation.

*Lemma 1:* Suppose $loc_1$, $loc_2$ and $loc_3$ represents three points in 2D and $loc_3.x > loc_2.x > loc_1.x$. Let *L*, *R* and *h* are defined as in Figure 4.5, then $\| loc_1, loc_2\|_2 + \| loc_2, loc_3\|_2 > \| loc_1, loc_3\|_2$ by a maximum of *2h*.

*Proof*: From basic triangulation we have

$$(L+h)^2 \, = \, L^2 \, + \, h^2 \, + 2Lh$$

$$L^2 \, + \, h^2 = \, (L+h)^2 \, - 2Lh$$

For $L \geq 0$, $h \geq 0$ we have $$L^2 \, + \, h^2 \leq (L+h)^2$$

$$\sqrt{L^2 \, + \, h^2} \leq \, L+h \tag{3}$$

Similarly for $R \geq 0$ we have $$\sqrt{R^2 \, + \, h^2} \leq \, R+h \tag{4}$$

As illustrated in Figure 4.5, the distance between $loc_1$ and $loc_3$ is

$$L + R$$

and the distance between $loc_1$ and $loc_3$ passing through $loc_2$ is

$$\sqrt{L^2 \, + \, h^2} + \sqrt{R^2 \, + \, h^2}$$

From equations (3) and (4) $$\sqrt{L^2 \, + \, h^2} + \sqrt{R^2 \, + \, h^2} \leq L + R + 2h$$

Hence the maximum difference between the two distances is *2h* ■

By comparing the upper-bound error of location $loc_2$ with the value of $h$, we can formulate the constraint used to filter previous locations, prior the speed estimation, based on the predetermined threshold.

*Theorem3*: For each two time-sequent locations $loc_j$ and $loc_l$ of a tag $t_i$ such that $loc_j(t_i).LAI$ *and* $loc_l(t_i).LAI \geq \Theta$, discard a location $loc_z(t_i)$ $\forall$ $z \neq j$, $l$ and $loc_j(t_i).t < loc_z(t_i).t < loc_l(t_i).t$ if the upper-bound error of $loc_z(t_i) \geq h$, which is defined in Lemma1.

*Proof*: From Lemma1 we have that the distance a tag $t_i$ travels from $loc_j$ to $loc_l$, passing through $loc_z$, is maximum double the value of $h$. If the upper-bound error of $loc_z$ is $\geq h$ and this value contributes twice to the traveled distance, then by subtracting the accumulated error ($\geq 2h$) from the travel distance we get:

$$traveled\ distance = \begin{cases} dist(loc_j, loc_l) & \rightarrow the\ tag\ t_i\ is\ located\ on\ the\ \text{line}\ loc_j loc_l \\ < dist(loc_j, loc_l) & \rightarrow a\ contradiction\ as\ h\ \geq 0 \end{cases} \quad (5)$$

From equation (5), tag $t_i$ moves on the line $loc_j$ $loc_l$ , hence the location $loc_z$ is invalid and the theorem follows. ■

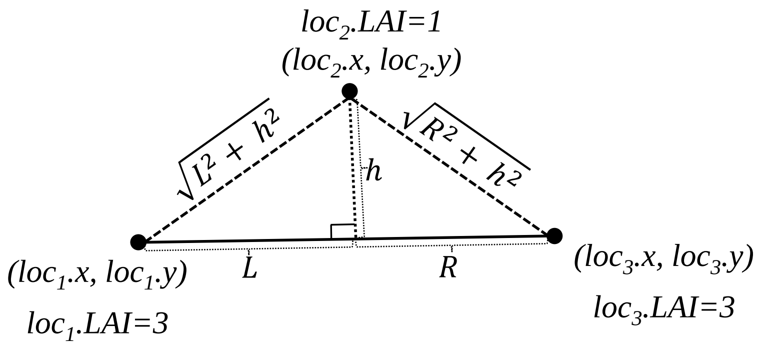


**Figure 4.5: Three locations for a tag *t* including uncertainty location information at *loc₂***

The object speed estimation process is illustrated in Algorithm 4.1.

**Algorithm 4.1** Object Speed Estimation Algorithm

**Input:** Object's location history: *Loc($t_i$)*  **Output:** estimated object: *speed($t_i$)*

```
set Loc($t_i$) = location history of a tag $t_i$ chronologically ordered   //limit the set by a parameter
if |Loc($t_i$)| ≤ 1 then
    speed($t_i$) = initial speed
else if |Loc($t_i$)| = 2 then
    speed($t_i$) = dist($loc_1$, $loc_2$) / ($loc_2$.t - $loc_1$.t)
else
    if optimize-speed-estimation = True then
        //filter locations that negatively affect tags' speed estimation
        set  $\varepsilon$ = location accuracy threshold
        set $Loc_{used}$($t_i$) = location history of tag $t_i$ with $loc_l$.LAI ≥ $\varepsilon$
        for each $loc_l$ in $Loc_{used}$($t_i$) do
            for each $loc_j$ in Loc($t_i$) do
                if ($loc_j$.LAI = 1 and $loc_l$.t < $loc_j$.t < $loc_{l+1}$) then
                    set $d_j$.r = tag-to-reader distance used to estimate $loc_j$
                    set $l_1$ = dist($loc_l$, $loc_j$)
                    set $l_2$ = dist($loc_j$, $loc_{l+1}$)
                    set $l_3$ = dist($loc_l$, $loc_{l+1}$)
                    set p = [dist($loc_l$, $loc_j$) + dist($loc_j$, $loc_{l+1}$) + dist($loc_l$, $loc_{l+1}$)] / 2
                    if $d_j$.r ≥ 2/$l_3$ * $\sqrt{p(p-l_1)*(p-l_2)*(p-l_3)}$ then delete locj
                end if
            end for
            return Loc($t_i$)
        end for
        return Loc($o_i$)
    end if
    //use exponential moving average (EMA) to estimate tag's speed based on its speed between
    each two sequent locations.
    set S = {$s_1$, $s_2$, $s_3$,…,$s_s$} //set of s previous speeds, $s_s$ is the most recent
    $EMAspeed_s(t_i) = \alpha * s_s + (1- \alpha) * EMAspeed_{s-1}(t_i)$
    speed($t_i$) = $EMAspeed_s(o_i)$
end if
return speed($t_i$)  // represents the tags' estimated speed
```

### 4.3.3 Time-Shifting Process and Multilateration

Algorithm 4.2 is designed to perform the time-shifting step in our proposed technique. It takes as input a number of asynchronous detections and based on tag speed and time difference, it expands the radius of detections accordingly and outputs a new synchronized set to which the common Multilateration can be applied.

**Algorithm 4.2** Time-shifting Algorithm

**Input:** asynchronous detection set *D* $(t_i)$ **Output:** synchronous detection set $D_{sync}(t_i)$

```
set $D(t_i)$ = set of $k$ detections of tag $t_i$ chronologically ordered
set $speed(t_i)$ = estimate tag speed using equation 1 or 2
    for $j=1$ to $k-1$ do
        $\Delta r_j = speed(t_i) * (d_k.t - d_j.t)$
        $d_j.r = d_j.r + \Delta r_j$ // increase its radius by distance traveled
    end for
return $D(t_i)$ // represents $D_{sync}(t_i)$
```

Using Multilateration, the coordinates of the tag *(x, y)* should satisfy the following equation:

$$(x - x_i)^2 + (y - y_i)^2 = dist_i^2 \qquad (6)$$

where $(x_i, y_i)$ are the *x* and *y* coordinates of the $i^{th}$ anchor node and the $dist_i$ is the measured distance between such anchor node and the tag to be localized. Typically in the literature, $dist_i$ includes a measurement error $\varepsilon_i$ , which is a zero-mean white Gaussian process $(\mathcal{N}(0, \sigma_i^2))$, where $\sigma$ is a variance correlated to the noise free distance and signal to noise ratio (SNR) as $\sigma^2 = (noise\_free\ distance)^2/SNR$ [147]. We assume that $\varepsilon_i$ is positive and we ignore it in our study.

Equation (6) can be modified to include the time-shifting step as follows:

$$(x - d_j.x)^2 + (y - d_j.y)^2 = (d_j.r + (s * (d_k.t - d_j.t)))^2 \qquad (7)$$

Equation (7) can be solved using Least Square Method as in [141] and [148] with computational complexity of $O(k^3)$ where $k$ is the size of the set $D_{used}(loc_l)$.

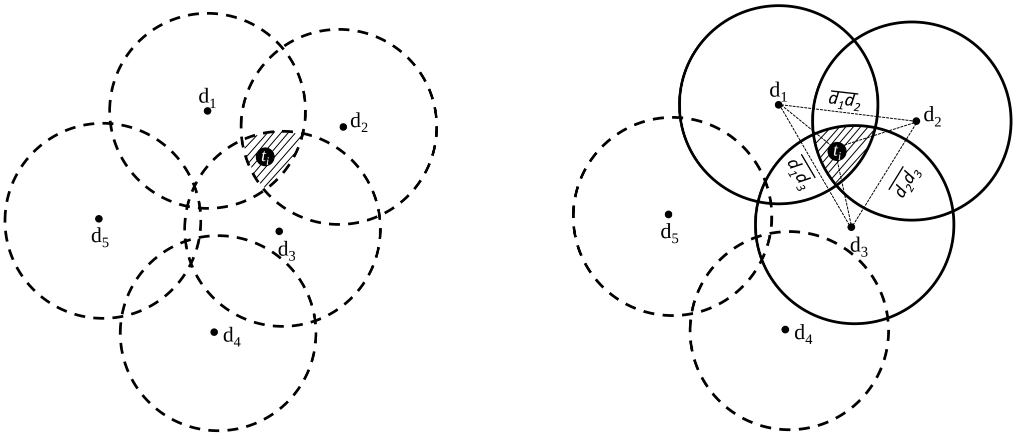
Generating $D_{used}(loc_l)$:

Given all available time-shifted detection set $D_{sync}(t_i)$ to estimate the location of a tag $t_i$, $loc_l$, $D_{used}(loc_l)$ is a subset of $D_{sync}(t_i)$ such that each detection in $D_{used}(loc_l)$ is intersected with all other detections as explained in Figure 4.6. Starting with the detection with smallest radius and follow with others one by one, we exclude detections that do not contribute to the intersect area of all previous detections. This process is explained further in Algorithm 4.3.

**Algorithm 4.3** Generating $D_{used}(loc_l)$

**Input:** synchronous detection set $D_{sync}(t_i)$ **Output:** valid detection set $D_{used}(loc_l)$

```
set $D_{sync}(t_i)$ = set of $k$ time-shifted detections of tag $t_i$ inversely ordered w.r.t. $d(t_i).r$
for each detection $d_j$ in $D_{sync}(t_i)$ do
    if $j = 1$ then $D_{used}(t_i)$.add ($d_j$)
    else
        $to\_add\_flag$ = True
        for each detection $d_k$ in $D_{used}(t_i)$ do
            if (distance between $(d_j.x, d_j.y)$ and $(d_k.x, d_k.y)$ > $(d_j.r + d_k.r)$) then
                $to\_add\_flag$ = False
                break
            end if
        end for
        if ($to\_add\_flag$ = True) then $D_{used}(t_i)$.add ($d_j$)
    end if
end for
return $D_{used}(t_i)$
```



(a) All available time-shifted detections.

(b) The filtered detections as solid circles.

**Figure 4.6: Generating the most valid detection set *$D_{used}$(loc$_l$)*.**

If a tag is stationary, all detections can be considered as synchronized and for error-free measurements, the tag position is exactly the intersection point of all detections represented as circles. For positive-error measurements, a tag definitely exists inside the intersection area of all detections represented as disks. However if a tag is mobile, what is the confidence that the tag exists in the intersection area of time-shifted detections in the set $D_{used}$? This confidence is related to the difference between the estimated average speed and the actual speed of the tag to be localized. In the following section, we study the effect of average tag speed on location accuracy and define upper bounds on the location error considering two cases: (1) the estimated tag speed is less than its actual speed and (2) the estimated tag speed is greater than its actual speed.

## 4.4 TSM Properties and Location Accuracy Upper Bound

In this section, we study the properties of the TSM technique and show that TSM can confine the location of a mobile tag to an upper-bounded set. In our study, we focus on one of the most common performance metrics used in the literature to evaluate a localization technique which is Location Accuracy, defined as

- ***Location Accuracy***: is the Euclidean distance between the estimated location of a tag and its actual location; representing the actual error. A lower value indicates better performance of the localization technique.

Although a lower bound of location accuracy is extensively used to evaluate a techniques' performance, the upper bound assessment is not less important especially for location-based applications that require specific locations (not just proximity information).

### 4.4.1 Preliminaries

Consider an initial case where a tag $t_i$ is stationary:

*Lemma 2:* If there is one detection about a tag $t_i$, $d(t_i)$, and the estimated location of $t_i$ is the reader position ($d.x$, $d.y$) then the upper bound of location accuracy is $d.r$.

*Proof*: A detection of a tag is represented as a disk centered at ($d.x$, $d.y$) with a radius $d.r$ equal to the estimated distance between a tag and a reader including an error $\varepsilon$. Considering that the error $\varepsilon$ is positive then $t_i$ is located at an arbitrary point inside the area $A= \Pi*d.r^2$. The worst case, if the estimated location of the tag is the center, is when it is actually located at any point on the edge. Thus, the Euclidean distance between estimated location and actual location $\leq d.r$. ■

*Definition 2*: Given the intersected detection set of a tag $t_i$, $D_{used}(t_i)$, the boundary set $S_B = \{s_1, s_2, s_3, \ldots, s_s\}$ is the set of all intersection points belong to the intersection area of all $d_k \in D_{used}(t_i)$ defined by $\bigcap_{i=1}^{k} d_i$ as explained in Figure 4.7.

For small $S_B$, we extend it by adding a set of virtual points $S_V$ which includes the middle point of each arc contributes to the intersection area. Thereafter, we refer to $\{S_B\} + \{S_V\}$ as $S$.

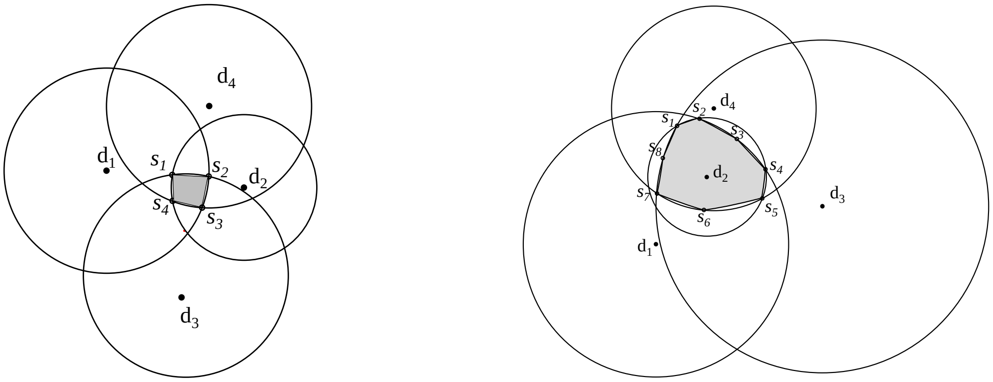


**Figure 4.7: Boundary set $S = \{s_1, s_2, s_{3, \ldots}\}$ and examples for relaxed intersection area.**

*Lemma 3:* Given the set $D_{used}(t_i) > 1$ for a tag $t_i$, if the estimated location of $t_i$ , $loc(t_i)$, is a point with the least square distance error with respect to every point $s \in S$ then the upper bound of location accuracy is $\max_{\forall s_z, s_s \in S,\ z \neq s} dist(s_z\, , s_s)$. ■

*Proof*: Given that the measurement error in each detection $d_i \in D_{used}(t_i)$ is assumed to be positive, the position of $t_i$ is definitely in the intersection area of $D_{used}(t_i)$. Irrespective of the estimated position of $t_i$, the upper-bound error can be formulated as the maximum distance between any two points in the intersection area $\bigcap_{i=1}^{k} d_i$, including the set $S$. By considering the estimated location, which has the least square distance error with respect to every point in $S$, the intersection area can be relaxed to be the area defined by all points in the set $S$ which is a convex polygon. Thus, the upper-bound error is the maximum distance between any two points in such convex polygon which is: $\max_{\forall s_z, s_s \in S,\ z \neq s} dist(s_z\, , s_s)$. ■

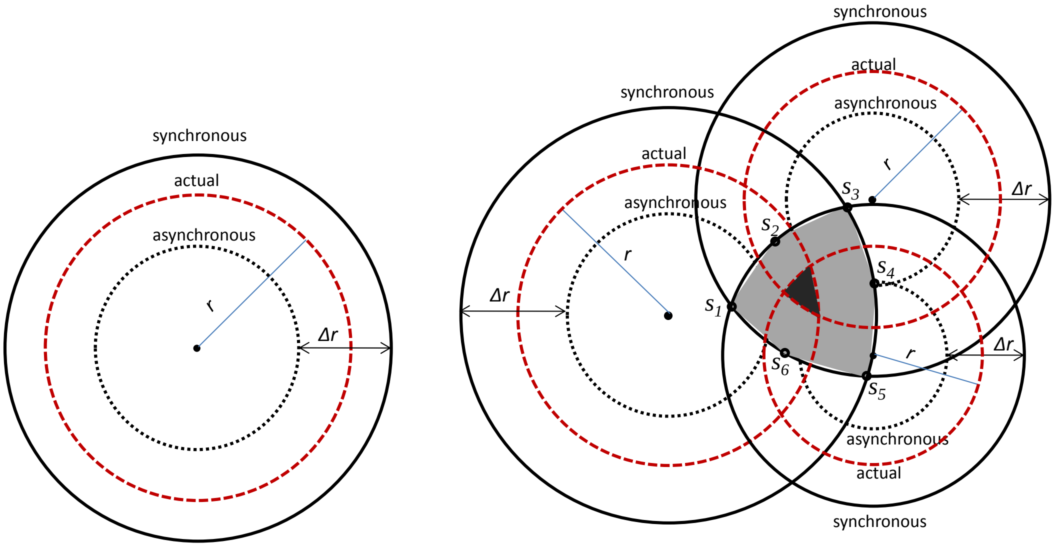
### 4.4.2 TSM Location Accuracy Upper Bound

In this section we study the upper-bound error for TSM technique under two different scenarios: (1) the estimated objects' speed is greater than its actual speed and (2) the estimated objects' speed is smaller than its actual speed. In both scenarios we confine that the location accuracy is upper-bounded.

### 4.4.2.1 Estimated speed is greater than actual speed

When the estimated speed is greater than the actual speed, the $\Delta r$ used in time-shifting process will be greater than the actual distance traveled by the object as illustrated in Figure 4.8 (a). Considering the assumption that the measurement error $\varepsilon_i$ is not negative, the actual area where the object is expected to be located in is included as a subarea in the area defined either by one detection $d(t_i)$ if $|D_{used}(t_i)| = 1$ or by all points in the set S and the upper bound of location accuracy is:

$$upper\ bound = \left\{ \begin{array}{ll} d(t_i).r & \rightarrow as\ per\ Lemma\ 2 \\ \max\limits_{\forall s_z, s_s \in S,\ z \neq s} dist(s_z, s_s) & \rightarrow as\ per\ Lemma\ 3 \end{array} \right.$$



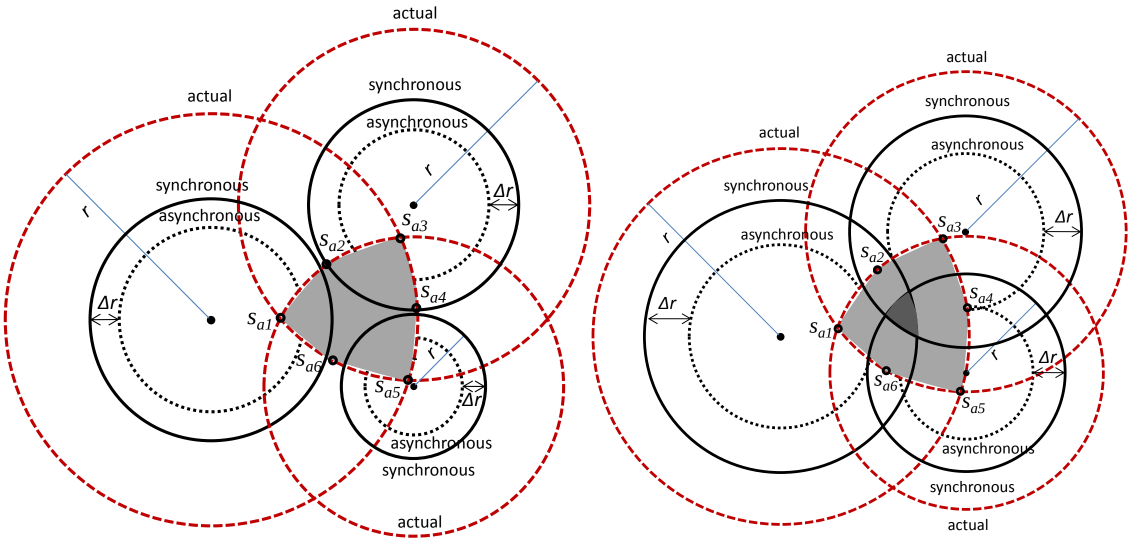
(a) Estimated speed > actual speed. (b) The convex polygon includes the actual location.

**Figure 4.8: Location accuracy upper bound in case of estimated speed is greater than actual speed.**

#### 4.4.2.2 Estimated speed is less than the actual speed

When the estimated speed is less than the actual speed, the $\Delta r$ used in time-shifting process will be smaller than the actual distance traveled by the object which may result in: (1) The synchronous detections do not intersect (i.e., $|D_{used}(t_i)|$ =1) and the proximity location estimation is applied w.r.t. the detection with the smallest radius amongst $D_{sync}(t_i)$ (see Figure 4.9 (a)) or (2) The synchronous detections intersect (i.e., $|D_{used}(t_i)| > 1$) and the estimated location of the tag is a point with the least square distance error with respect to every point $s \in \mathcal{S}$ (see Figure 4.9 (b)). To examine both cases we define another boundary set $S_A$ as follows.

*Definition 3*: Given the actual intersected detection set of a tag $t_i$, all with none negative error $\varepsilon$., the actual boundary set $S_A = \{S_{AB}\} + \{S_{AV}\} \neq 0$ where $S_{AB}$ is the set of all intersection points belong to the intersection area of all actual detections and $S_{AV}$ is the set of virtual points represent the middle point of each arc contributes to the intersection area as shown in Figure 4.9.



(a) None intersected synchronous detections. (b) Intersected synchronous detections.

**Figure 4.9: Location accuracy upper bound in case of estimated speed is less than actual speed.**

*Lemma 4:* If the synchronous detections do not intersect, $|D_{used}(t_i)|$ =1, and the estimated location of $t_i$ is the reader position ($d.x$, $d.y$) in $d(t_i) \in D_{used}(t_i)$, then the upper bound of location accuracy is: $\max_{\forall s_z \in S_A} dist((d.x, d.y), s_z)$, given that $|S_A| \geq 1$.

*Proof*: Given that all measurement errors are not negative, the synchronous detections, using the actual speed, are definitely intersected and the tag is expected to be at an arbitrary point inside this intersection area however the estimated tag location is the reader position which may be located outside such an intersection area. If the reader position is located inside the intersection area, Lemma 3 is valid and this lemma is proved. If the reader position is located outside the intersection area, then the upper bound of location accuracy is the maximum distance between the reader position and any point inside the intersection area. This can be relaxed to the maximum distance between the reader position and any point s $\in S_A$ and the lemma follows. ■

*Lemma 5:* If the synchronous detections intersect and the estimated location of the tag is a point with the least square distance error with respect to every point $s \in S$ then the upper bound of location accuracy is: $\max_{\forall s_s, s_z \in \{S_A \cup S\},\ z \neq s} dist(s_s, s_z)$, given that $|S_A| \geq 1$ and $|S| \geq 1$.

*Proof:* If the estimated intersection area is fully included in the intersection area of the synchronous detection using actual speed then definitely the two points, which define the maximum distance, $s_s$ *and* $s_z \in S_A$ thus Lemma 3 is valid and this lemma is proved. If the estimated intersection area is partially included in the intersection area of the synchronous detection using actual speed then the upper bound is defined as the maximum distance between any point in the actual intersection area and any point in the estimated intersection area. As in Lemma 3, this problem can be relaxed by considering the polygon defined by points in $\{S_A \cup S\}$

thus the upper bound of location accuracy is the maximum distance between its vertices and the lemma follows. ■

## 4.5 Simulation Results and Analysis

In this section, we evaluate the TSM technique and investigate its performance under different dynamicity settings through extensive simulations using ns-3. During the conducted experiments, we compare the performance of TSM with the Multilateration technique. Specifically we are interested in *average speed accuracy* and *average location error* metrics defined as follows:

- ***Average Speed Error*** which represents the difference between the object actual speed and the speed estimated by the TSM technique. This metric is strongly related to the availability of objects' location history and how accurate these locations are. Lower values for this metric indicate better performance.

- ***Average Location Error*** which represents the Euclidean distance between the actual location of an object and its estimated location (i.e., actual error). This metric is directly affected by the valid detections used in location estimation as well as how accurate those detections are. A lower location error indicates better performance.

### 4.5.1 Simulation Setup

We adopt the ns-3 network simulator [149] to implement all processes of TSM technique and to simulate an IoT scenario as shown in Figure 4.10. This scenario represents an open area of *250m* x *250m* in which we randomly deploy abundant number of passive objects or simply tags (i.e., *1000* passive tags) to simulate a typical crowd event such as a city Fair or festival. We only consider *100* randomly selected tags to be localized. In addition, we deploy a number of mobile readers that, along with the tags, move based on Random Way Mobility Model with a pedestrian speed ranging from *0.7m/sec* to *1.5m/sec*. While they move, readers are allowed to detect and notify surrounding tags, with a reading range of *7m*, and localize interrogated tags based on TSM technique. Localization information is disseminated using the RICTags technique explained in Chapter 3.

We adopt the Received Signal Strength Indicator (*RSSI*) for estimating the distance between a tag and a reader in each detection; considering measurements error. In measuring a distance between a tag $t_i$ and a reader $r_j$, we introduce an error $\epsilon_{i,j}$ as a zero-mean white Gaussian process ($\mathcal{N}(0,\sigma_{i,j}^2)$), where $\sigma$ is a variance correlated to the noise free distance and signal to noise ratio (SNR) as $\sigma^2$= $(noise\_free\ distance)^2/SNR$, which is assumed to be accurately estimated and is a known priori [147]. In addition, as we assume that readers are capable of acquiring their

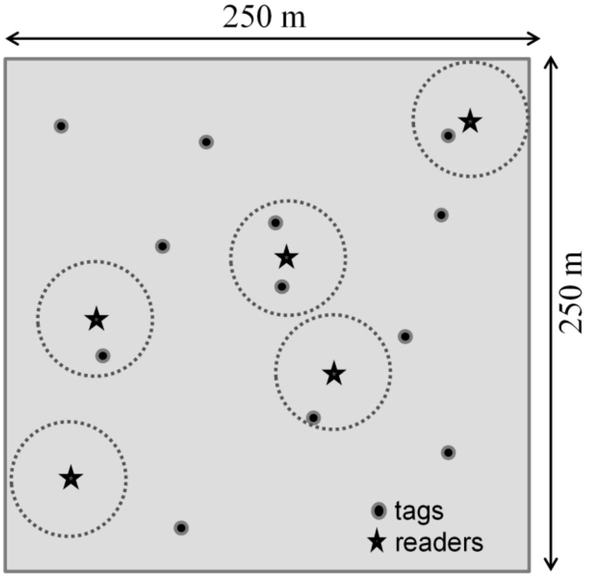


**Figure 4.10: An open area where tags and readers move using random way mobility model.**

positions which may contain error as well, we consider an error in both *x* and *y* coordinates of a reader position as a zero-mean white Gaussian process ($\mathcal{N}(0,\sigma_i^2)$), where $\sigma$ is a value ranged from *0.2m* to *1.4m*. When this error is not mentioned, we assume that mobile readers are accurately localized.

We perform the simulation experiments under different settings in terms of the number of mobile readers and mobility speed of both readers and tags. The results shown are averaged over ten different independent runs with distinct random seeds with a total simulation time of *5000sec* per run. The results are within ± 3% of the average with 90% confidence level.

### 4.5.2 Results and Analysis

We examine the simulation results for two cases: when an object location is estimated using one or more detections ($LAI \geq 1$) and when at least three detections are used in location estimation (LAI ≥ *1)*. The latter will naturally result in higher localization accuracy hence less location error, but may not always be feasible. In the case of only one detection, both TSM and Multilateration techniques give an approximate location which is the reader position at time of detection however TSM benefits from old detections and estimated tag speed to enhance the location accuracy.

**Average Speed Error:**

In this experiment, we study how accurate TSM estimates tag speed and show the impact of tags' mobility on the average speed error, while considering different numbers of mobile readers (*50, 100*, and *150*). The results, as depicted in Figure 4.11, show that TSM gives lower average speed error at low mobility (from *0.8m/sec* to *1.2m/sec*) for different numbers of mobile readers: average of *9%*, *7%*, and *4%* for *50*, *100*, and *150* mobile readers, respectively. Increasing the number of mobile readers decreases the average location error (as shown in Figure 4.13) which

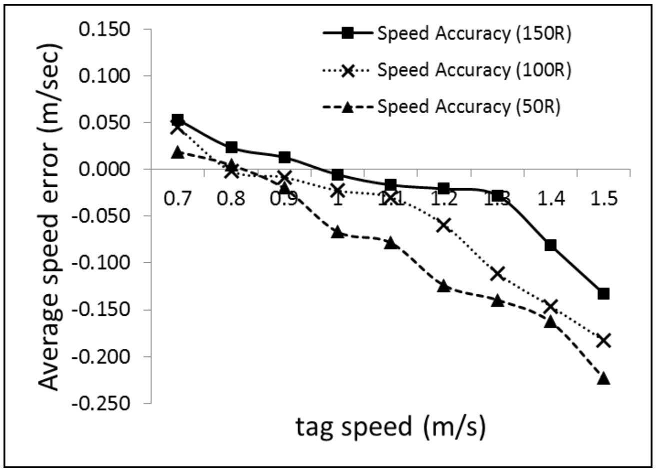


**Figure 4.11: Impact of mobility on average speed error for three different # of mobile readers (*50, 100, and 150 mobile readers*).**

helps TSM to have more accurate location history, which enhances the speed estimation. Note that even the average speed error increased to *13*% at high mobility (in case of *150* mobile readers), the average location error is not aggressively affected as showing in Figure 4.15. From Figure 4.11, we also conclude that within the same scenario, the upper bound defined in Section 4.4.2.1 can be applied at low mobility while the upper bound defined in Section 4.4.2.2 can be applied at high mobility. The definition of low or high mobility depends on the value of application parameters such as the number of mobile readers and the default speed in the absence of location history (we adopt *0.9m/sec* as a default speed in our scenarios).

We also study the average speed error over the simulation time for high mobility speeds. In this experiment we measure the average speed error over *5* sequenced time intervals (*500sec* per each); considering *3* different tags' mobility (*1.1m/sec*, *1.3m/sec*, and *1.5m/sec*) as shown in Figure 4.12. The figure shows that even though the average speed error during the first time interval is high (*26%* for *1.5m/sec*), it enhances over time and reaches *14%* at time interval *5*. This enhancement over the simulation time is a result of the availability of more accurate tag

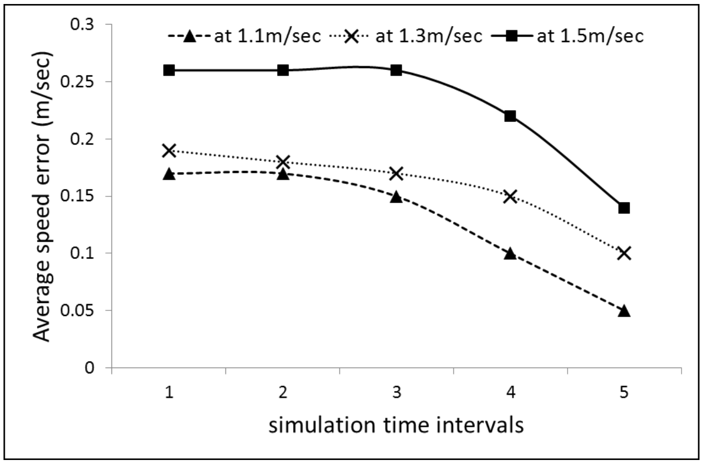


**Figure 4.12: Average speed error over simulation time for 3 different mobility speeds (*150 mobile readers*).**

location history even at high mobility. Such a location history is built by TSM by adopting detections to either enhance location estimation and/or estimate tags' location when no synchronous detections are available (see Figure 4.15 which shows how TSM maintains its performance in terms of average location error).

**Average Location Error:**

In the first experiment, we study the impact of the number of mobile readers on the average location error while considering LAI ≥ *1*. The results, as depicted in Figure 4.13, show that increasing the number of mobile readers helps in localizing more tags and/or increasing the number of detections used in localization, thus both Multilateration and TSM show better average location error. However, TSM shows an average enhancement of up to *6%* over Multilateration. This enhancement is a result of the time-shifting process, which adapts detections based on the estimated tag speed, allowing more detections to contribute to the localization estimation.

In Figure 4.13, we allow tags to move using random speeds ranging from *0.7m/sec* to *1.5m/sec,* while in Figure 4.14 we focus on low mobility and high mobility and show the average location error accordingly. The results in Figure 4.14 coincides with Figure 4.13 in that the average location error is enhanced for both schemes when the number of mobile readers increases but TSM has better enhancement even under high mobility. The figure shows that TSM outperforms Multilateration by *4%* at low mobility and *13%* at high mobility for *LAI ≥ 1*.

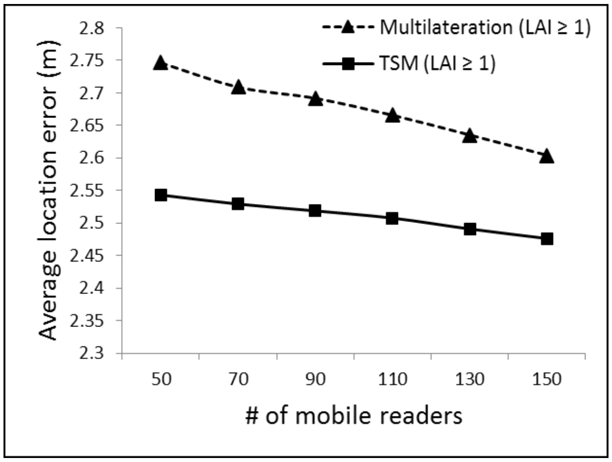


**Figure 4.13: Impact of number of mobile readers on average location error using random speed (***ranged from 0.7m/sec to 1.5m/sec***).**

Next we study the impact of the tags' speed on the average location error, while considering *LAI ≥ 1 and LAI ≥ 3.* During the experiment, we increase the tags' speed from *0.7m/sec* to

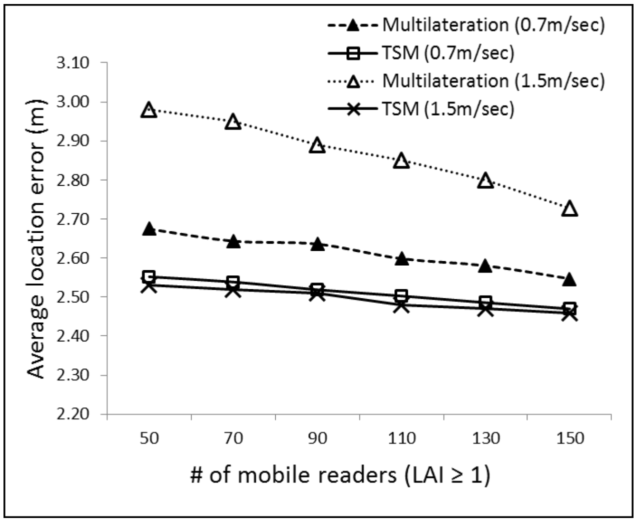


**Figure 4.14: Impact of number of mobile readers on average location error using two different tags' speed (*0.7m/sec and 1.5m/sec*).**

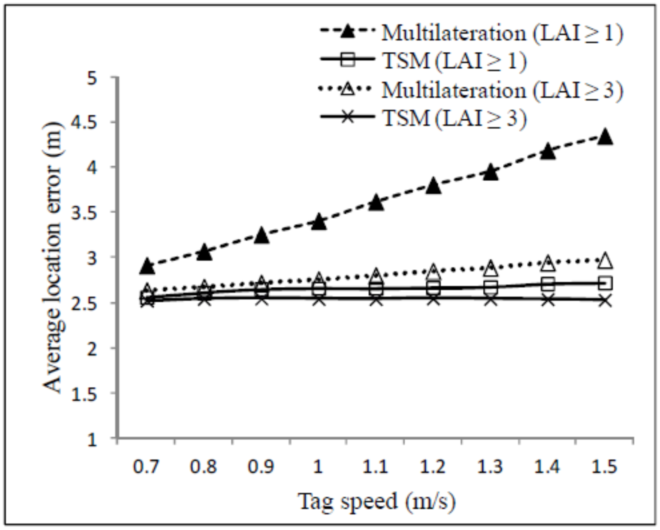


**Figure 4.15: Impact of tags' speed on average location error (*100 mobile readers*).**

*1.5m/sec* while using *100* mobile readers as shown in Figure 4.15. As depicted in the figure, both schemes have better accuracy at low mobility and/or when *LAI* ≥ *3*. Note though that TSM is less affected by tags' speed than Multilateration even when *LAI* ≥ *1* due to the time-shifting process. For *LAI* ≥ *3*, the average location error of TSM and Multilateration converge at low tags' speed values (*0.7m/s*) with TSM outperforming Multilateration by *9%*. At high tags' speed (*1.5m/s*), the average location error of Multilateration for both *LAI* ≥ *1* and *LAI* ≥ *3* increased due to lack of valid detections at high mobility, whereas better result is for *LAI* ≥ *3*. On the other hand, TSM maintains its performance in terms of the average location error as the tags' speed estimation and time shifting processes alleviate the negative effects of tag speed. In fact, the accuracy of TSM with *LAI* ≥ *1* at high mobility shows *100%* improvement over Multilateration.

In the aforementioned experiments, we assume that mobile readers are accurately localized however this assumption may not be realistic in practice. For location determination of mobile readers in outdoor environments, GPS-based positioning, coupled with street maps, is used with typical accuracies of *1-3* meters. While indoors, where GPS signals are no longer available, wireless technologies such as WiFi, Ultra Wide Band (UWB), Ultrasonic, or RFID can be used for positioning, providing meter-level accuracy [103] [104]. To justify the TSM performance in such scenarios, we investigate the effect of error in readers' position (ranged from *0.2m* to *1.4m*) on the tags' average location error. Figure 4.16 shows that the error in mobile readers' positions does not aggressively affect the average location error. As shown in the figure, the average location error of Multilateration and TSM are respectively affected by *8%* and *10%* when the error in mobile readers' positions moves from *0.2m* to *1.4m*.

Another parameter that has impact on the average location error is how frequent mobile readers generate detection records about tags in their interrogation zones. In this experiment, we investigate such impact considering *5* different detection frequencies as shown in Figure 4.17. As depicted in the figure, at higher detection frequencies (say every *2sec*) more detections positively contribute to the localization estimation thus both Multilateration and TSM show better average

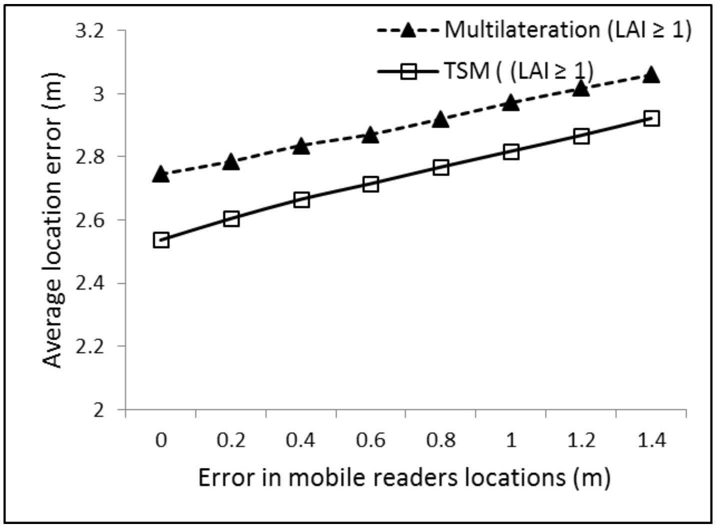


**Figure 4.16: Effect of error in mobile readers' position on average location error (*100 mobile readers*).**

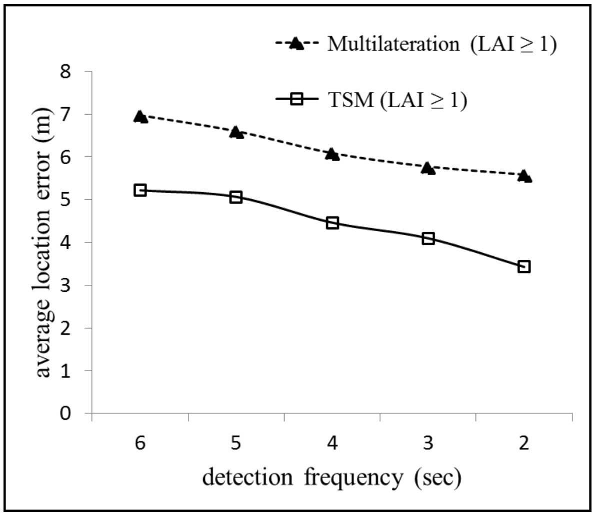


**Figure 4.17: Impact of detection frequency on average location error (*50 mobile readers*).**

location error. However, TSM outperforms Multilateration by *25%* at low detection frequencies and *40%* at high detection frequencies. This better performance even at low detection frequencies is due to the time shifting process which overcomes the challenge of having a lack of detections.

## 4.6 Conclusion

This chapter tackles the problem of accurately localizing passive objects in distributed and dynamic environments (i.e., typical IoT scenarios). The main challenge in these environments is to have sufficient synchronous spatial information about mobile objects in order to localize them, which is assumed to be available typically in most of lateration-based localization techniques. This assumption, however, may not be reliable in a typical dynamic environment where anchors are mobile and heterogeneous with different and/or short communication ranges.

To overcome this challenge, this chapter proposes the technique Time-Shifted Multilateration (TSM) which utilizes the available asynchronous spatial information to localize and/or enhance objects' location estimation in the absence of sufficient concurrent readings. TSM shifts the available spatial information based on the estimated objects' speed and time differences to reflect the expected current spatial information of the object providing better accuracy. Its properties show that TSM can confine the location of a mobile object to an upper-bounded set. In addition, we assess TSM performance in a fully distributed and dynamic environment through extensive simulations using ns-3 focusing on actual error in object's speed estimation and location estimation. The results show that under different dynamicity settings, TSM is able to estimate objects' speed with average error of *4%* and outperforms Multilateration in all scenarios in terms of average location error and shows up to *90%* improvement at high mobility. We remark that TSM can be combined with any other lateration-based localization technique to enhance mobile object location estimation when the number of spatial information at a specific time is not sufficient.

# Chapter 5

# Maintaining Availability of Location Information

In Chapter 3, we propose two different fully distributed RFID-based cooperative localization schemes: ReaDS and RICTags. These schemes allow a group of ad hoc heterogeneous and independent mobile RFID readers to estimate locations of surrounding passive-tagged objects (or simply tags) in a typical IoT dynamic environments. The operation of both schemes results in either each mobile reader knowing the locations of a subgroup of tags as in ReaDS or a tag maintaining its own location as in RICTags. Such location information has to be available to system participants by means of a dissemination technique. Such dissemination technique can provide location information availability by managing location queries and replies when certain location information is required.

In this chapter, we propose two different location information dissemination strategies. First, we propose GOSSIPY Pull strategy where the mobile readers exchange location queries or replies when a certain tags' location is required through a reactive protocol. In this strategy, mobile RFID readers have to communicate with one another in a peer-to-peer mechanism to ensure timely dissemination of location information. Second, we propose using a simple, inexpensive and flexible component known as "memory spots" to disseminate location information and to exchange location queries without the need for direct communication amongst readers. To evaluate our proposed strategies, we introduce two main performance metrics: *localization delay*, which represents the average time the system takes to respond to a location query generated by any interested participant, and *average overhead*, which represents the average number of messages the system participants exchange to respond to a location query. We use ns-3 to simulate a typical IoT dynamic scenario and investigate through several experiments the impact of different dynamicity settings on the aforementioned performance metrics. Our results show that although GOSSIPY Pull strategy provides the recent location information through exchanging queries among readers, it requires readers to identify each other to communicate and the scheme may have scalability issues. On the other hand, using memory spots performs well in terms of localization delay and average overhead under different dynamicity settings. As a proof of concept, we demonstrate an actual deployment of memory spots for information dissemination and obtain real measurements through carrying on an indoor experiment using passive RFID tags and readers.

The remainder of this chapter is organized as follows: Section 5.1 presents some of the related work and motivates our proposed strategies. Components, notations and assumptions are given in Section 5.2. GOSSIPY Pull strategy is detailed in Section 5.3. Section 5.4 presents the dissemination strategy using memory spots. Performance evaluation and result analysis are given in Section 5.5 and the chapter is concluded by a discussion in Section 5.6.

## 5.1 Related Work and Motivation

Typically, in RFID-based tag localization systems, tags are localized through a set of fixed and coordinated RFID readers which detect surrounding tags and report spatial information about detected tags to a central location server. This central server is in charge of estimating tag locations and providing location information to interested users. Although this centralized approach is robust and caters to a wide range of applications, it provides limited scalability and may not be a practical solution for dynamic IoT. Thus, for distributed localization systems using RFID technology, there is a lack of a distributed information dissemination strategy. Such a strategy has to be scalable, should ensure timely dissemination of location information among a system's participants, and must require minimal central infrastructure.

For data dissemination and queries processing in ad hoc networks, abundant research has been conducted tailored for different ad hoc network paradigms such as WSNs and Delay Tolerant Networks (DTNs). In WSNs, queries, as well as data are disseminated by means of different routing protocols such as directed diffusion [150], Two-Tier data dissemination [151][152] , and Gradient broadcast [152] - [155]. The main goal of these routing protocols is to collect sensing data from sensor nodes and transmit to a sink (i.e., queries generator) which could be mobile following a certain trajectory [156], while conserving sensor energy to maximize the network lifetime. In achieving their goal, the aforementioned protocols organize WSNs into different topologies (e.g., cluster-based or tree-based) or dynamically construct a chain from a sensor node to the sink considering that the sensor nodes are static. Geographic routing is another routing paradigm for WSNs where the interest is sent to sensor nodes in a specified region to serve region-based queries. Geographic routing protocols such as EAGR [157] utilize the geographical location information of each sensor node to deliver data over a network towards the destination.

Due to the limited resources of sensor nodes, solutions for query processing in WSNs are not supportive of remote and global queries. To overcome this limitation, researchers investigated using more powerful mobile devices as a second tier for the purpose of query processing. The research in [158] proposes using ad hoc mobile devices as a second tier in a systematic framework for end-to-end query processing in traditional static WSNs. These mobile devices are used as query generators, query carriers, query injectors to the region of interest in the sensor nodes tier and query result collectors. For mobile devices to exchange query and query's result, the author adopted the geographic routing paradigm in the mobile devices layer since they consider region-based queries instead of global network and they assume that mobile devices are location aware.

DTN on the other side has been proposed to design protocols to ensure data transmission in intermittently connected mobile networks with high and unpredictable delay typically based on a store-carry-forward approach. The two main strategies for data dissemination and query processing in such network paradigm are flooding and random walk. In flooding-based strategies, each node sends a copy of carried data or query to all nodes it meets and dissemination is controlled by means of Time to Life (*TTL*) or hop count. While dissemination strategies based on random walk use probabilistic paths to reach to destination or responder [159]. Although flooding-based strategies have high probability of delivery, they suffer from high overhead, which degrades their overall performance. Other work such as [160] - [163] are proposed to bound the overhead in terms of number of copies and transmissions per message; comparing to flooding-based strategies. For example, Spray and Wait [160] and Spray and Focus [161] spread only small number of copies of a message to the first few relays encountered (spray), and wait for any one of the relays to transmit the message directly to the destination as they move into the network (wait) or each relay forwards its copy further close to the destination using a single-copy utility-based function (focus).

Although there is some similarity between data dissemination and queries processing in WSNs and DTN and these in IoT RFID systems, the problem we intend to solve has different characteristics: (1) RFID readers can change their locations over time and each is only aware of its current location, (2) location information is fully distributed and each mobile reader may be interested in location information managed by other readers, (3) tags to be localized are passive and have limited resources in terms of memory, power and communication capabilities, and (4) the interested readers may not know the exact geographic area where the tag(s) of interest exist.

Considering these characteristics, we devise reactive strategies to ensure timely dissemination of location information in distributed RFID systems.

## 5.2 Components, Notations and Assumptions

We consider a group of ad hoc mobile RFID readers, where each reader maintains location information about surrounding tags (as explained in Chapter 3) and is willing to share such location information with other readers in a timely fashion. This information may range from simple proximity location to more accurate location based on readers' capabilities (i.e., adaptive power, antenna array, etc.). For simplicity, we consider only proximity location information which contains: *time*, *tag_id* and *tag_position* (i.e., reader position at time of location estimation). The operation of each proposed strategy depends on all or some of the following components:

1. ***Tags***, representing the objects to be localized, which can be either stationary or mobile and are identified by passive RFID tags.
2. ***Readers***, representing the ad hoc uncoordinated, dynamic and heterogeneous RFID readers, which can localize surrounding *Tags* and willing to share their location information with others.
3. ***Memory Spots***, representing inexpensive, flexible and limited storage and processing devices that can use WiFi, Bluetooth or RFID technology, which can be distributed in smart areas to offer information storage and retrieval for mobile devices.
4. ***Location Information***, a table for each reader. It contains time-stamped estimated positions of interrogated tags and location information received from neighbors through information dissemination process. Attributes of location information are given in Chapter 3, however no need to recall it as it does not affect the operation of our dissemination strategies.
5. ***Location Queries***: hold queries about tags that are of interest and need to be localized.

Hereinafter we use the notations defined in Table 5-1 to refer to the aforementioned components, as well as other design parameters in algorithms and examples.

**Table 5-1: Used Notations**

| *Component/Parameter* | *Notation* |
|---|---|
| *Tags* | $T = \{t_1, t_2, t_3 ...\}$ is the set of $n$ mobile /stationary tags. |
| *Readers* | $R = \{r_1, r_2 ...\}$ is the set of $m$ mobile readers. |
| *Memory Spots* | $MS = \{ms_1, ms_2 ...\}$ is the set of $l$ memory spots. |
| *One-hop neighbors* | $NR_i = \{nr_{i1}, nr_{i2}, ...\} \subseteq R$ is a subset of the mobile readers that are in the neighborhood of reader $r_i$ at time $\tau$. |
| *Query Timeout* | *q_timeout* is the time to life *TTL* of generated location queries. |
| *Update Interval* | *up_int* is the time interval between successive queries from $r_i$ to $NR_i$. |

In this work, we assume that *Readers*, as mobile devices, can acquire their own locations at any given time via one of the positioning systems for mobile devices (e.g., GPS, WiFi, or anchors). *Readers* can reach neighboring readers to share location information (GOSSIPY Pull strategy) or communicate with and update all shared *Memory Spots* (infrastructure-assisted strategy).

## 5.3 GOSSIPY Pull Strategy

GOSSIPY Pull is a flooding-based strategy in which *Readers* behave reactively with location information queries generated by interested *Readers*. The process starts with a reader sending a time-stamped query message asking for a tag(s) of interest to its one-hop neighbors. For simplicity, we explain the process assuming that *Readers* break down query for multiple tags into multiple single tag query messages. The query message consists of *requestor_id, query_id,* *q_timeout* and *tag_id*. Each reader receiving this query plays the role of either responder or forwarder. If it has the tags' location, it generates a reply message containing the location information (*responder_id, time, tag_id* and *tag_position*) and sends it to its one-hop neighbors. Else, it forwards the query itself with no modification. Doing so allows the query to disseminate between *Readers* until a responder(s) is found or the query has expired based on the *query timeout* parameter. Each reader maintains *requestor_id* and *query_id* for forwarded query as long as the query is not expired; allowing it to avoid any loopback in queries forwarding process. Due to *Readers* mobility, there is no guarantee for a reply message to travel along the same path of the query message back to the requestor. Thus, we allow reply messages to be disseminated as query message; allowing the forwarder *Readers* to update such reply message or their location table accordingly.

When a reader receives a reply message, it carries out one of the following actions:

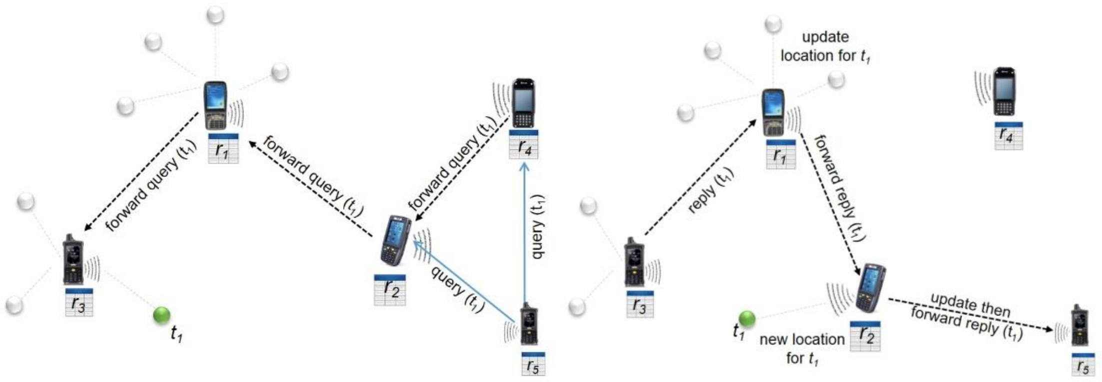
- Discard the reply message if it is originally created by the reader as a responder and has been forwarded back during the dissemination process.
- Update its location information and stop forwarding the message if the reader is the requestor.
- Update its location information if the tag of interest is unknown and discard the reply message if it is outdated or forward it to its one-hop neighbors.
- Update the reply message if the reader has more recent location information about the tag of interest and forward the updated reply message to its one-hop neighbors.

Algorithm 5.1 details how the entire process takes place.

**Algorithm 5.1:** GOSSIPY Pull strategy - run by *Readers*

**Input:** Tag of interest **Output:** updated location tables

```
if $tag_i$ is to be localized then
    Send Query Message ($tag_i$_id)
end if
if query message QM is received then
    //check if the query is outdated or it is a loopback message to ignore
    if (QM.requestor_id is my ID or QM.time <current time – q_timeout) then
        Ignore message
    //reply to the query if there is unexpired location information or forward the query
    else if QM.$tag_i$_id exists in location table then
        set $NR_i$ = current neighbors
        set reply message (responder_id, QM.q_id, QM. $tag_i$_id, time, location, q_timeout)
        send reply message to $NR_i$
    else
        set $NR_i$ = current neighbors
        send QM to $NR_i$
    end if
end if
if reply message RM is received then // at any mobile reader in $NR_i$
    if RM.responder_id is my ID then
        Ignore message   //it is a loopback
    else if RM.time <current time – q_timeout then
        if RM.$tag_i$_id is unknown then
            Update location table (RM)  //add location information of unknown Tags
        end if
    else if RM.time > location record (RM. $tag_i$_id).time  then
        Update location table (RM)  //update location information with most recent and forward
        set $NR_i$ = current neighbors
        send RM to $NR_i$
    end if
end if
-----------------
procedure Send Query Message ($tag_i$_id)
    set query message = (requestor_id, q_id, $tag_i$_id, time, q_timeout)
    while $tag_i$_id is not localized do
        for each up_int do
            set $NR_i$ = current neighbors
            send query message to $NR_i$
        end for
end procedure
```



(a)

(b)

**Figure 5.1: GOSSIPY Pull Strategy**

If the requestor does not receive a reply to its query, it resends another query after waiting for *update interval* as explained in Algorithm I. Due to the randomness of query and response messages, the requestor may get multiple replies for the same query. In this case, the requestor selects the most recent location information based on the time stamp of location information included in each reply. Figure 5.1 shows an illustrative example on how query and response are forwarded amongst *Readers*. As shown in Figure 5.1 (a), $r_5$ is interested in localizing $t_1$ which is out of its interrogation zone; hence, it disseminated a query to its one-hop neighbors ($r_2$ and $r_4$). Accordingly, $r_4$, which does not have location information about $t_1$, forwarded the query to its one-hop neighbor $r_2$. At $r_2$, only one copy of the query was handled and re-forwarded to $r_1$ then $r_3$. As explained in Figure 5.1 (b), when $r_1$ received a reply from $r_3$, it updated its location information and forwarded the reply to its neighbors including $r_2$. However $r_2$ has more recent location information for $t_1$, so it updated the reply message and forwarded it to its neighbors including $r_5$ which is the requestor.

## 5.4 Dissemination Using Memory Spots Strategy

In the second strategy, we introduce the concept of "*Memory Spots*" and design a proactive/reactive distributed protocol for location information dissemination indirectly amongst *Readers* with the support of these *Memory Spots*, providing high location information availability with lower overhead. As previously mentioned in Section 5.2, *Memory Spots* are limited storage and processing devices [164][166] that can use WiFi, Bluetooth or RFID technology, and are typically distributed in smart areas to be used by mobile devices. *Readers* periodically synchronize *Tags*' location information with *Memory Spots* they may pass by. In the synchronization process, a reader: (1) updates the *Memory Spots* with interrogated *Tags* (i.e., proactive), (2) obtains location of *Tags* beyond its interrogation zone and (3) either replies to or carries on and propagates location queries that may exist (i.e., reactive). Carrying a query allows rapid propagation of such query towards other *Memory Spots*. *Readers* interested in the location of a tag can interrogate the nearest *Memory Spot* to pull location information obtained from other passing *Readers*, or to register a location query. The general framework of this strategy is illustrated in Figure 5.2. In this strategy we define two events: *synchronize* event at which a reader communicates with any *Memory Spot* it may pass by, updates the location information on such *Memory Spot* and carries location information that needs to be disseminated for interests of

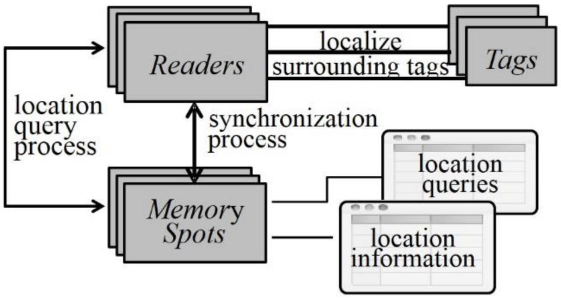


**Figure 5.2: General framework of the dissemination strategy based on Memory spots.**

other *Readers*. The second is an occasional event, named *query*, based on need. When a *Reader* is interested in localizing a tag(s) beyond its interrogation zone, it checks any *Memory Spot* it may pass by to pull the required location information or to submit a location query to be manipulated during coming *synchronize* events. A different location information dissemination strategy is required for *Memory Spots* in order to identify which location information to be carried by passing *Readers* in each synchronization process (see Figure 5.3 where a flag, named *to_carry*, is added and further explanation is given in Section 5.4.2). The following subsections detail the two main processes: location query and synchronization processes from the perspective of *Readers* and *Memory Spots*.

### 5.4.1 Location Query

This process is executed by a reader when it is interested in localizing tag(s) out of its current interrogation zone. As explained in Algorithm 5.2, a reader interested in a tags' location, first looks for the tag of interest in its local location information. If the required location information does not exist, the reader starts communicating with *Memory Spots* it passes by to either pull the required location information or registering a location query which contains: *requestor_id*, *query-*

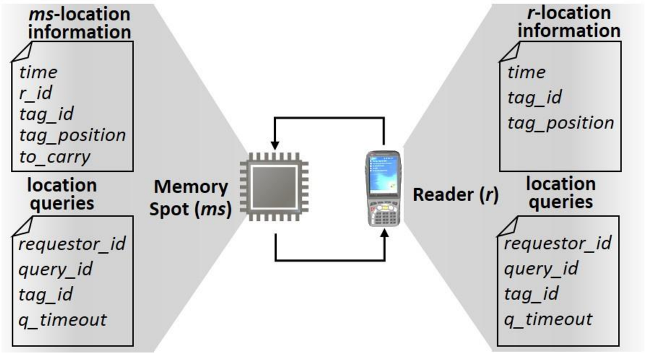


**Figure 5.3: Attributes of location information and location queries on both *Memory Spots* and *Readers*.**

*id*, *tag_id*, and *q_timeout*.

**Algorithm 5.2** Location query Algorithm - run by *Readers*

**Input:** tag ID **Output:** tags' location information

```
set loc(requestor_id) = r_id.location information
set loc(MS_id) = ms_id.location information
set query(MS_id) = ms_id.location queries
  //look up the tag(s) of interest in reader's location information before generating query
if tag_id exists in loc(requestor_id) then
    set not localized = False
    return tag_id.position
else
  //contact Memory Spots to retrieve tags' location information or submit a location query.
    set not_localized = True
    while not_localized do
        contact MS
        for each contacted ms do
            if tag_id exists in loc(ms_id) then
                set not localized = False
                return tag_id.position
            else        //submit a location query
                generate Q(requestor_id, query_id, tag_id, q_timeout)
                add Q to query(ms_id)
            end if
        end for
end if
```

### 5.4.2 Synchronization Process

The main process in this location information dissemination strategy is to iteratively synchronize location information on the distributed *Memory Spots* by passing *Readers*. When a *Reader* conducts a *synchronize* event and accordingly communicates with a *Memory Spot*, the following actions take place:

1. The *Reader* updates the *Memory Spot*'s location information and acquires locations of *Tags* it may be interested in.

2. The *Reader* pushes its location queries into the *Memory Spot*, which are carried from other *Memory Spots* through previous synchronize events.
3. The *Memory Spot* filters its location queries and discards any query that has been answered or has expired. Accordingly, the *Memory Spot* switches the flag named "*to_carry*" in its location information (Figure 5.3) for *Tags* in answered queries to be disseminated by passing *Readers*.
4. The *Reader* carries both "*to_carry*" location information and remaining queries from the *Memory Spot* to propagate them into other *Memory Spots* that it may pass by.

Algorithm 5.3 details how location information is updated by passing *Reader* on the contacted *Memory Spot* (i.e., action 1), and it shows how the *Reader* pushes the queries that it holds into the *Memory Spot* to be answered by any other passing *Reader* (i.e., action 2). During the update, the most recent location information is considered in the event of conflicting results. However, in case of having more accurate location estimation technique than proximity; the accuracy has to be considered as well to decide which location information to use.

**Algorithm 5.3** Location and query information updating Algorithm - run by *Readers*

**Input:** location and query information **Output:** updated location and query information

```
set loc(r_id) = r_id.location information
set query(r_id) = r_id.location queries
set loc(ms_id) = ms_id.location information
set query(ms_id) = ms_id.location queries
for each record $rec_i$ in loc(r_id)  do
    //push Readers' location information to update contacted Memory Spot
    if ($rec_i$.tag_id is not exist in loc(ms_id)  or $rec_i$.time is most recent) then
            add ($rec_i$.time, r_id, $rec_i$. tag_id, $rec_i$.tag_position)  to loc(ms_id)
    end if
end for
    //push carried location queries to the Memory Spot
for each query $q_i$ in query(r_id) do
        add (r_id, $q_i$.query_id, $q_i$.tag_id, $q_i$.q_timeout) to query(ms_id)
end for
```

Running Algorithm 5.3 allows location queries to accumulate at the *Memory Spot* which may include answered, unanswered or expired queries that need to be manipulated. In addition, the *Memory Spot* is required to highlight the location information for the queries that have been answered to be carried and disseminated to serve interests of other *Readers*; regardless of the interests of current *Reader*. This takes place by switching the *to-carry* flags of these *Tags* to the value of 1. Algorithm 5.4 illustrates how the aforementioned steps generally run by *Memory Spots*, which represents the third action in the synchronization process.

**Algorithm 5.4** Queries filtration Algorithm - run by *Memory Spot*

**Input:** location queries **Output:** filtered location queries

```
set loc(ms_id) = ms_id.location information
set query(ms_id) = ms_id.location queries
for each query $q_i$ in query(ms_id) do
    if $q_j$.tag_id is exist in loc(ms_id) then
    //mark this record to be carried in coming synchronization event and delete the related query
        set loc(ms_id)→ $q_j$.tag_id.to_carry = 1
        delete $q_j$
    else if $q_j$.timeout < 0 then   delete $q_j$
    //maintain unanswered and unexpired queries
    end if
end for
```

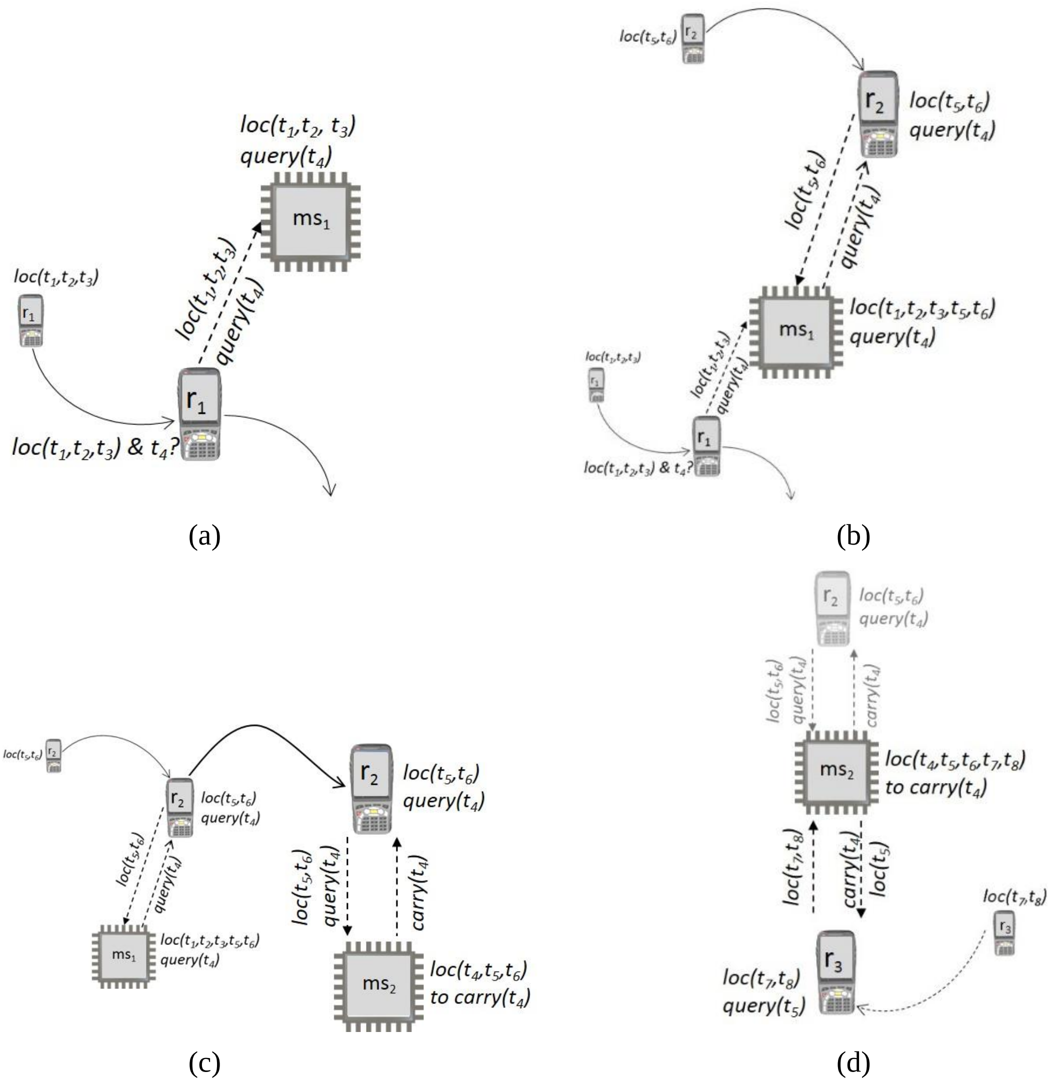
Running Algorithm 5.4 reduces *Memory Spot* resource usage and communication overhead of passing *Readers* by only serving unanswered queries and not propagating outdated ones. Algorithm 5.5 explains the last action in the synchronization process (i.e., action 4) executed by passing *Readers* for the purpose of information dissemination. In the algorithm, a *Reader* carries both the remaining unanswered queries in addition to the "*to-carry*" location information to be considered in the next synchronization process. To avoid loopback, the queries that are originally created by such *Reader* are ignored. When a *Reader* updates its location information, it only considers time-stamped *Tags* locations irrespective of the reader that localized such tags.

**Algorithm 5.5** Carry location queries Algorithm - run by *Reader*

**Input:** location queries **Output:** updated location queries

```
set loc(r_id) = r_id.location information
set query(r_id) = r_id.location queries
set loc(ms_id) = ms_id.location information
set query(ms_id) = ms_id.location queries
for each record $rec_j$ in loc(ms_id)  do
    if $rec_j$.to_carry = 1 then  //add this location information to Reader's location information table
            add ($rec_j$.time, $rec_j$.tag_id, $rec_j$.tag_position) to loc(r_id)
    end if
end for
for each query $q_i$ in query(ms_id)  do
    if ($q_i$.r_id $\neq$ r_id ) then   //carry unanswered queries that do not belong to that this Reader
        add ($q_i$.r_id, $q_i$.query_id, $q_i$.tag_id, $q_i$.q_timeout) to query(r_id)
    end if
end for
```

Figure 5.4 shows an illustrative example for the operation of our proposed strategy. In Figure 5.4 (a), the reader $r_1$ localized tags: $t_1$, $t_2$, $t_3$ and it was interested in localizing $t_4$. Then, $r_1$ communicated with the nearest memory spot $ms_1$, updated $ms_1$ with locations of $t_1$, $t_2$, $t_3$ and registered a query asking about $t_4$. The reader $r_2$ was in the vicinity of $ms_1$ as well (see Figure 5.4 (b)). So, $r_2$ updated $ms_1$ with locations of $t_5$, $t_6$ and carried the query that was generated by $r_1$ about $t_4$. As in Figure 5.4 (c), $r_2$, while it moves, communicated with memory spot $ms_2$ and did the following: updated $ms_2$ with locations of $t_5$, $t_6$, pushed the query about $t_4$ into $ms_2$. $ms_2$ had the location of $t_4$ which was previously updated by another passing reader, therefore $ms_2$ turned the *to_carry* flag of $t_4$ location to *1*. Accordingly, $r_2$ carried this location information to disseminate it across other memory spots. As in Figure 5.4 (d), at $ms_2$, another reader $r_3$ was interested in locations that was updated by $r_2$, so it acquired such information (location of $t_5$) without query registration. In addition, $r_3$ updated $ms_2$ with locations of $t_7$, $t_8$ and carried the location of $t_4$ as well. $r_1$ while it moves, communicated with a memory spot that has been updated by either $r_2$ or $r_3$ hence, it acquired the location of $t_4$.



(a) (b)

(c) (d)

**Figure 5.4: Illustrative example for dissemination strategy based on Memory Spots.**

(a) $r_1$ updates $ms_1$ and register a query about $t_4$. (b) $r_2$ updates $ms_1$ and carry the query generated by $r_1$. (c) $r_2$ updates $ms_2$ and disseminates the query generated by $r_1$, accordingly $ms_2$ marks $t_4$ as "*to_carry*" thus, $r_2$ carries $t_4$. (d) $r_3$ updates $ms_2$, looks for $t_5$ and carries $t_4$ as well

## 5.5 Performance Evaluation

In this section, we evaluate the proposed location information strategies through simulation. We investigate the performance of the two strategies presented: GOSSIPY Pull which is presented in Section 5.3 and using memory spots as presented in Section 5.4. These strategies are based on RFID crowdsourcing by ad hoc mobile readers, passive tags and memory spots in a dynamic IoT environment. Accordingly, their performance may be influenced by a number of dynamic factors such as area topology, number of readers, mobility speed of readers and how frequently they contact each other or memory spots to disseminate or acquire location information. We are interested in two performance metrics:

- ***Localization Delay*** which represents the average time the system takes to respond to a location query generated by any of interested readers. This metric is affected by the availability of location information and lower localization delay indicates better performance.

- ***Average Overhead*** which represents the average number of messages the system's participants exchange (amongst readers or between readers and memory spots) to respond to a location query. The lower number of messages exchanged to fulfill a location query indicates better performance.

### 5.5.1 Simulation Setup

We adopt the ns-3 network simulator [149] in order to simulate a typical IoT scenario as shown in Figure 5.5, to evaluate our proposed strategies. In this scenario, we simulate a mini attraction area of *200m* x *200m* containing *14* point of interest, which are linked using pathways of *8m* width. We randomly deploy abundant number of tags (*1000*) while we consider only a *100* randomly selected tags to be used during generating queries.

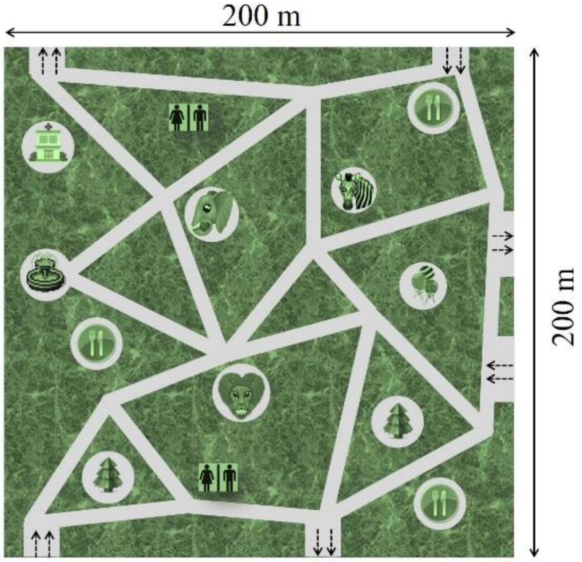


**Figure 5.5: Simulated environment: representing an area with preplanned pathways where tags and readers are only allowed to move through.**

In addition, we deploy different numbers of mobile readers that move based on Graph Based Mobility Model with a pedestrian speed (range from *0.7m/sec* to *1.5m/sec* [170*]*). Based on this mobility model, the mobile readers are randomly deployed at points of interest and are only allowed to move on those pathways to a randomly selected point of interest. We also allow them to pause for a period of time (say *10sec*) at each point of interest during their movement. After the pause period, each mobile reader changes its speed and moves to another randomly selected point of interest. Mobile readers are in charge of localizing surrounding tags using one of our distributed localization systems proposed in Chapter 3 and either directly communicate with each other to exchange location queries and responses (i.e., GOSSIPY Pull) or communicate with memory spots deployed in the area. Each mobile reader has a reading range of *5m* to interrogate surrounding tags and communicate with each other and a *30m* reading range to communicate with memory spots. We also deployed different numbers of memory spots at points of interests and in the pathways. The default *update interval* (or *synchronize interval* in case of using memory spots) is set to *30sec* while each mobile reader is allowed to be interested in a random tag and accordingly generate a location query every *60sec (*named, *request interval)*. These parameters can be set to other values according to given experiment.

Considering this scenario, we investigate the effects of various parameters such as number of mobile readers, number of memory spots, update interval, request interval and mobility speed on the performance of our proposed strategies in terms of localization delay and average overhead. In calculating the localization delay, we allow readers to generate location queries every *query interval* then we compute the time it takes for the reader to get a reply and take the average over all generated queries. The results shown are averaged over ten different independent runs with distinct random seeds with a total simulation time of *5000sec* per run. The results are within ± 4% of the average with 90% confidence level.

### 5.5.2 Simulation Results

**<u>Localization delay:</u>**

In the first experiment, we examine the impact of the number of mobile readers on the localization delay and compare the performance of the two proposed strategies while considering *3* different synchronization or update intervals (*60sec*, *90sec* and *120sec*) as shown in Figure 5.6. The results show that increasing the number of mobile readers enhances the localization delay for both strategies however, the better delay for both strategies is for the scenario of more frequent synchronization or update events (i.e., *update interval* of *60sec* in our experiment).

In case of GOSSIPY Pull strategy, the localization delay is dramatically decreased by increasing the number of mobile readers but on account of average overhead as seen in Figure 5.11. As depicted from the figure, using memory spots allows readers to rapidly disseminate

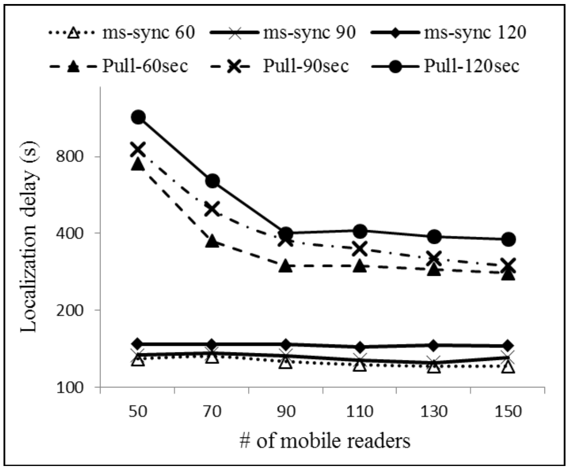


**Figure 5.7: Impact of number of mobile readers on localization delay for three different synchronization intervals (using 14 memory spots at points of interest)**

location queries hence answers more queries with considerably lower delay than the GOSSIPY Pull strategy.

In the second experiment, we study the impact of the number of memory spots on the localization delay (does not apply for GOSSIPY Pull strategy). The results show that increasing the number of memory spots slightly affects the localization delay (by an average of only 3% as

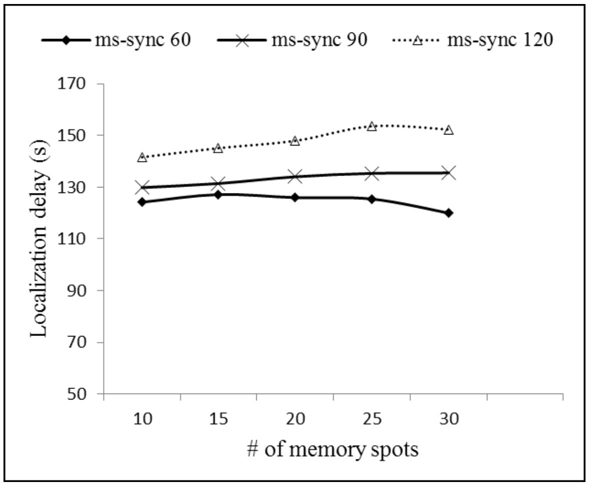


**Figure 5.6: Effect of number of memory spots on localization delay (using *75* readers)**

shown in Figure 5.7). This effect is due to the travel time of either the location queries or replies may be longer at higher number of memory spots with less frequent synchronization events. In addition, the long travel time of either the location queries or replies may result in increasing the average overhead as explained in Figure 5.12.

We also conduct an experiment to further investigate the impact of the frequency of *synchronize event* considering different number of mobile readers (as shown in Figure 5.8). The results support our observations in Figure 5.6 and show that the better localization delay is for higher number of mobile readers, which synchronize their location information with memory spots more frequently (e.g., by an average of 11% when the number of mobile readers is doubled and an average of 15% when the synchronization frequency is quadrupled). However, more attention should be given to the average overhead which is proportional to the synchronization frequency but slightly affected by the number of mobile readers (see Figure 5.13).

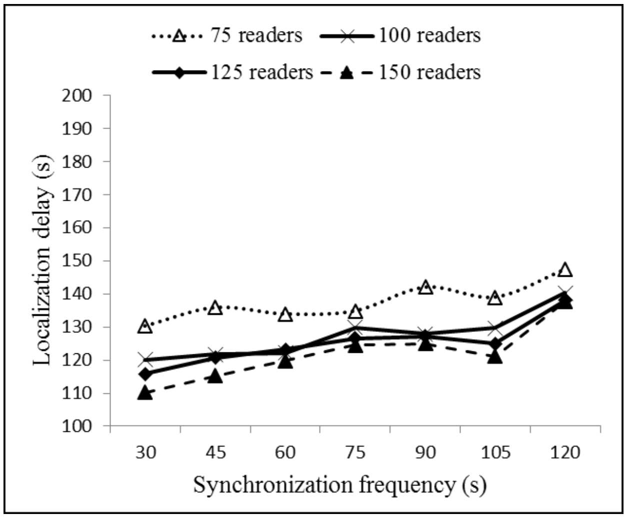


**Figure 5.8: Impact of synchronization frequency on localization delay for different number of mobile readers (using 14 memory spots at points of interest).**

Another parameter that may have effect on localization delay is how frequent location queries are generated by interested mobile readers (*request interval*). In this experiment, we investigate such impact considering *4* different synchronization intervals (*45sec*, *60sec*, *75sec* and *90sec*) as shown in Figure 5.9. The results in the figure coincide with the results shown in Figure 5.8 in terms of less frequent synchronization events resulting in lower localization delay. However, generating more queries slightly affects the localization delay (e.g., by an average of 3.7% when the rate of generating location queries is quadrupled – from every *120sec* to every *30sec*); providing a consistent performance in terms of localization delay.

The impact of mobile readers' speed is depicted in Figure 5.10 with respect to localization delay. The figure shows that high speed mobility enhances the localization delay with an average of *8%* when the speed is almost doubled. This enhancement is due to:(1) the mobile readers move faster hence, rapidly update memory spots; resulting in location information dissemination prior query generation; and (2) fast mobility allows readers to carry and forward unanswered location queries in a shorter time compared with low mobility speed which results in less response time.

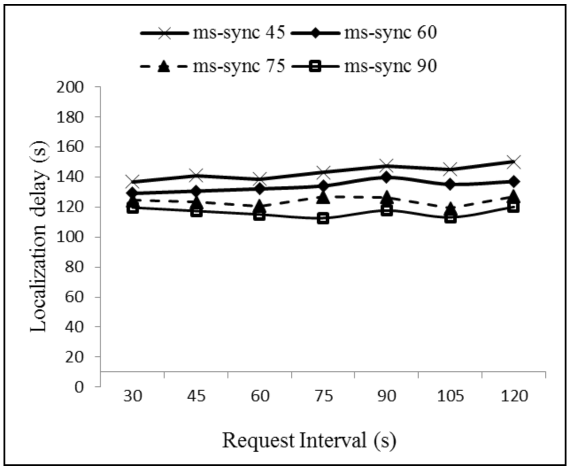


**Figure 5.9: Impact of different request intervals on localization delay for different synchronization intervals (using *14* memory spots and *75* mobile reader).**

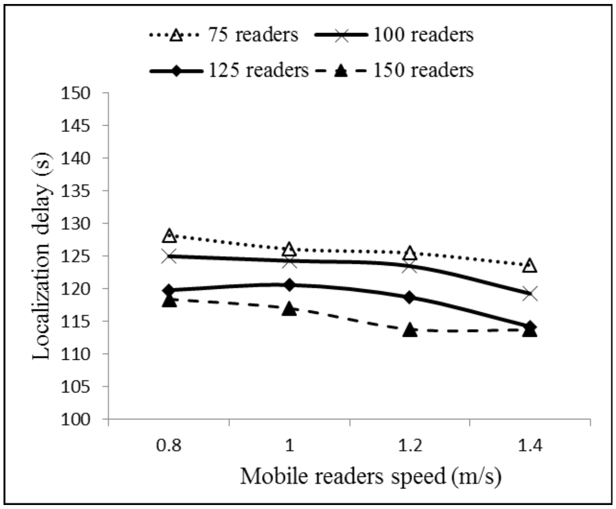


**Figure 5.10: Impact of mobile readers speed on localization delay for different number of mobile readers (using 14 memory spots at points of interest).**

**Average Overhead:**

The impact of the number of mobile readers on average overhead is investigated for both strategies as illustrated in Figure 5.11. We measure the average overhead under 3 different synchronization or update intervals (*60sec*, *90sec* and *120sec*) and typically lower overhead is observed when synchronization events occur less frequently but on account of localization delay (refer Figure 5.6). As shown in Figure 5.11, increasing number of mobile readers in pull strategy magnifies the average overhead due to broadcasting location queries and replies among mobile readers. Using memory spots as focal points amongst mobile readers (i.e., no direct communication), the average overhead is reduced by an average of 70% when the number of mobile readers is tripled (from 50 readers to 150 readers) when compared to GOSSIPY Pull strategy even at less frequent synchronization. This enhancement is because more mobile readers, memory spots are updated in a timely fashion, hence location queries are satisfied with fewer carry and forwarding messages.

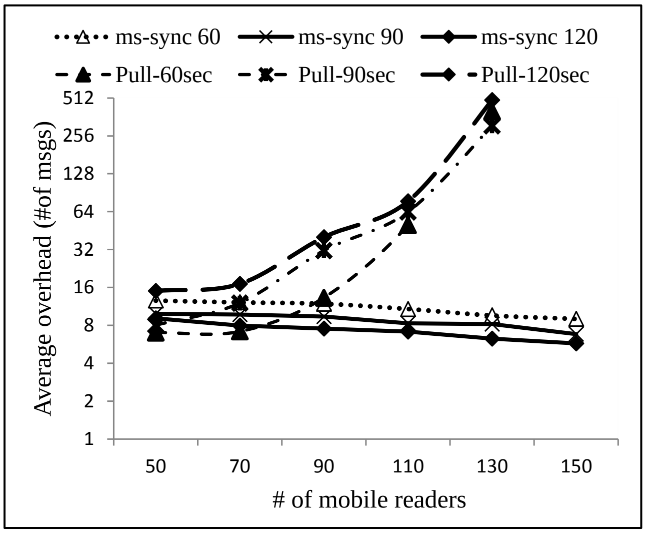


**Figure 5.11: Effect of number of mobile readers on average overhead (using 14 memory spots).**

Next we study the impact of the number of memory spots on the average overhead as shown in Figure 5.12 (does not apply for GOSSIPY Pull strategy). When the number of memory spots is tripled (i.e., by adding more memory spots in the pathways not only at points of interest), the average overhead is adversely affected (68% worse). This is because location queries and replies are required to traverse more memory spots, which generates more messages and increases the

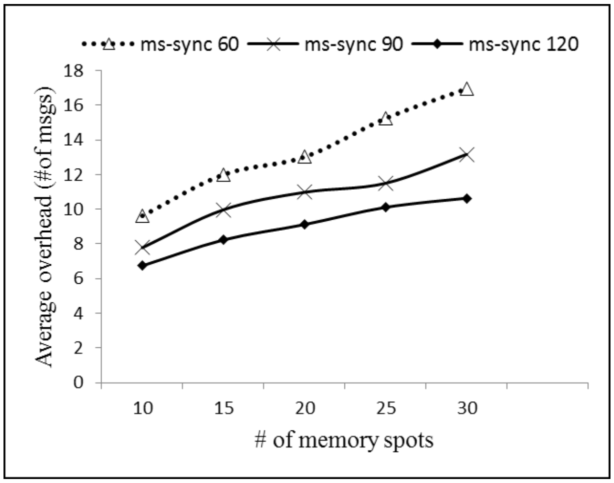


**Figure 5.12: Effect of number of memory spots on average overhead (using *75* readers).**

average overhead. The least effect on the average overhead is for the less frequent synchronization (i.e., *120sec*) where mobile readers maintain their location information and carried queries for longer time before updating memory spots which consequently decreases the traversed memory spots.

While increasing the frequency of event synchronization enhances the localization delay as shown in Figure 5.8 however, it has an adverse effect in terms of average overhead as seen in Figure 5.13. Increasing the frequency of event synchronization significantly affects the average overhead while the number of mobile readers has an insignificant effect (*3%*) which might be neglected compared to the effect of synchronization frequency. The same adverse effect of increasing the synchronization frequency is also experienced when interested readers generate location queries more frequently (4 times faster) as illustrated in Figure 5.14. Although increasing the frequency of event synchronization results in higher overhead.

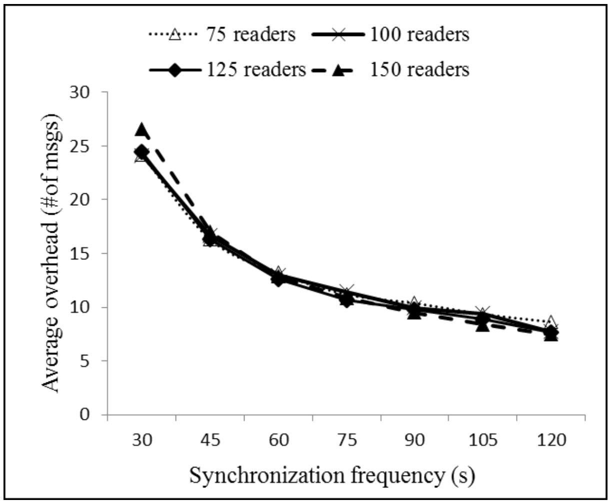


**Figure 5.13: Impact of synchronization frequency on average overhead for different # of mobile readers (using 14 memory spots).**

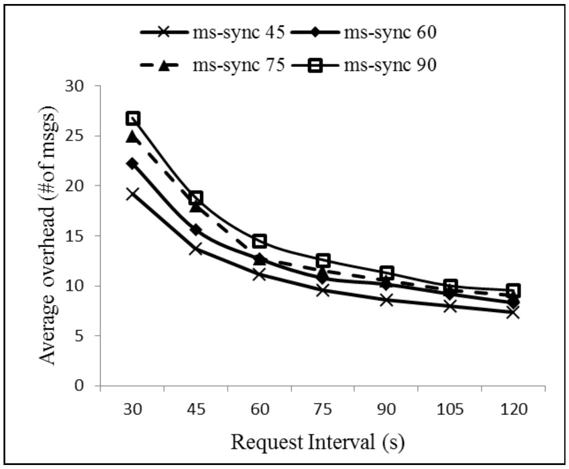


**Figure 5.14: Impact of different request intervals on average overhead for different synchronization intervals (using 14 memory spots and 75 mobile readers).**

As observed in Figure 5.14, at a high rate of location queries, more frequent synchronization events result in lower overhead. Faster location information dissemination can satisfy more queries through location information updates with no need for carry and forward messages; decreasing the average overhead by an average of 24%.

In our last experiment, we investigate the impact of mobile readers' speed on the average

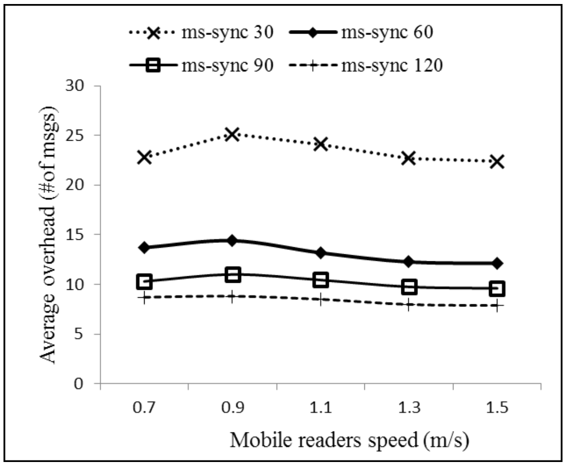


**Figure 5.15: Impact of mobile readers speed on average overhead for different synchronization intervals (using 14 memory spots and 75 readers).**

overhead while considering four different synchronization intervals (*45sec*, *60sec*, *75sec*, and *90sec*). The results (see Figure 5.15) show that the mobility speed of mobile readers slightly enhances the average overhead (i.e., by an average of *7%* when the speed is doubled), with better performance at lower synchronization frequency.

### 5.5.3 Field Experiment

In this section, we demonstrate using memory spots for information dissemination and obtain real measurements through carrying on an indoor experiment using an actual passive RFID system. In the experiment we design a cubicle-level localization system using passive mobile readers to localize passive-tagged objects (e.g., staff properties such as laptops, bags and personal items) over three labs at the School of Computing at Queen's University as illustrated in Figure 5.16. We allow mobile readers to provide cubicle level accuracy with support of RFID reference tags, which are carefully deployed in each cubicle such that the power level of mobile readers is adjusted accordingly. For location information dissemination, we deploy three BeagleBoard-xM [167] which act as memory spots in the corridor which are programmed to run Algorithm 5.5

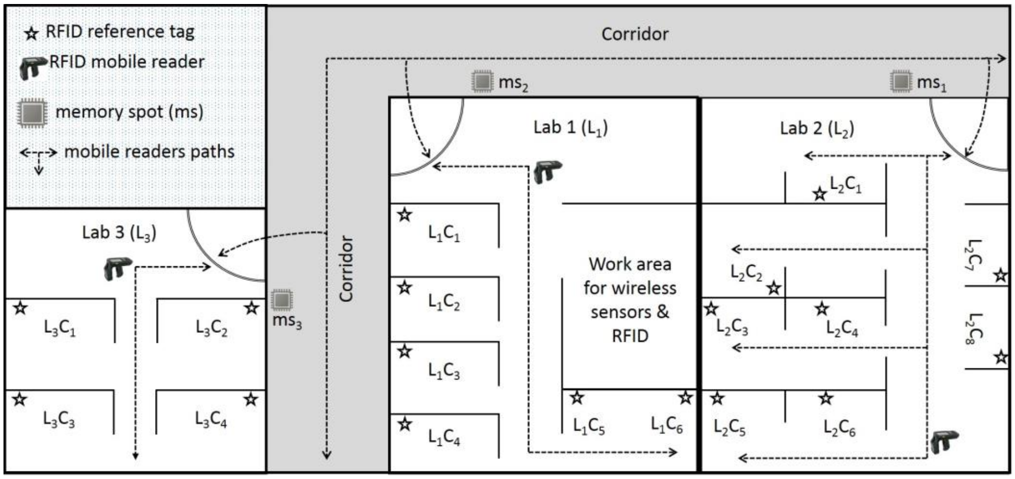


**Figure 5.16: Lab testbed experiment environment.**

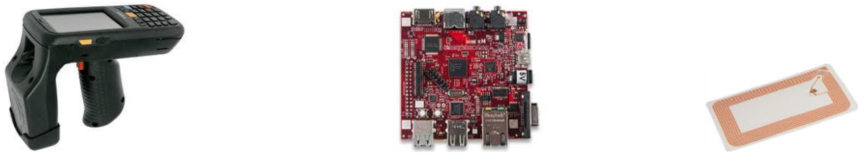
defined in Section 5.4. The mobile readers, while moving, share locations of localized tagged-objects and submit location queries by executing Algorithm 5.3, 5.4 and 5.5 defined in Section 5.4 with respect to deployed memory spots ($ms_1$, $ms_2$ and $ms_3$ as in Figure 5.16). In the following, we present the system components, the system execution process including setup and implementation details and experimental use cases along with results.

**System components**

1. RFID passive mobile reader (*DotH-300U*): is a UHF RFID range hand-held terminal operating in the *860* MHz ~ *960* MHz UHF (see Figure 5.17 (a)) with reading range up to *5* meters. The reader is based on the EPCglobal Gen2 specification [16] with *512MB* NAND flash memory and *256MB* of memory MDDr. The Operating system is Windows® CE with a reader display of *3.5* inches touch screen. The mobile reader communicates through Wireless LAN IEEE 802.11b/g/n, and Bluetooth.

2. BeagleBoard-xM: is a low cost ARM Cortex A8 board with Texas Instruments Cortex A8 1GHz processor and *512* MB RAM [167] and runs a custom Ubuntu 11.4 operating system designed for the BeagleBoard and is enriched by USB WiFi Dongle (see Figure 5.17 (b)).

3. RFID passive tag: is a read/write RFID transponder (see Figure 5.17 (c)) operating in the *860* MHz ~ *960* MHz UHF range and is based on the EPCglobal Gen2 specification with *96* bits of user programmable memory field with read, write and lock capabilities. Its package is a cut



(a) DotH-300U RFID reader. (b) BeagleBorad xM. (c) RFID passive tag.

**Figure 5.17: System components used in field experiments.**

tape of *88.9*mm x *25.40*mm which allow it to be easily attached to objects of interest. Some of them are to be used as reference tags to provide cubicle-level accuracy while others are to be used to identify objects of interest (e.g., staff properties such as laptops, bags and personal items).

**System execution process**

**Setup**: In the first stage, we create a *reference database*, which binds a tag ID to each cubicle included in our experiment as illustrated in Figure 5.16; assigning a name to each tag which is the student name of corresponding cubicle. These tags are then deployed in corresponding cubicles to function as reference tags such that the location of interrogated tags is determined based on the concurrently detected reference tag (e.g., proximity-level localization). In the second stage, we identify a collection of students' properties with passive tags and create a *dictionary database* that maps each tag to a name corresponding to the object identified by such tag. The aforementioned databases are used in determining the proximate location of tagged-objects and to provide names of objects when maintaining location information and satisfying location queries. Last, we use KeyWedge, an application on the *DotH-300U* mobile reader, to adjust the power level and interrogation frequency of the mobile readers by iteratively walking through cubicles based on given pathways described in Figure 5.16 while interrogating tags, including reference tags, using different power levels and/or interrogation frequencies. We consider the power level and interrogation frequency that allows the reader to only interrogate tags in given cubicle when it pass through and be close to such cubicle for a specific period of time.

**Implementation details**:

Most of the implementation steps take place on the mobile readers and memory spots as the tags are passive. For the mobile readers which run Windows handheld classic version 6.5, we developed two console applications using Microsoft Visual Studio 2008. The first application periodically takes the file generated by KeyWedge which contains scanned tags within a certain scan period (*5sec*) and with support of *reference database*, it generates location information file and automatically beams it to memory spot using Wi-Fi. This location information file contains *date, time, tag_id, cubicle* for all interrogated tags and is used to update location information on memory spot as per Algorithm 5.3. The second application receives location query database from a memory spot, dumps expired queries and pushes the remaining queries to any other contacted memory spot. We also developed a Windows application using Microsoft Visual Studio 2008 to allow mobile readers (and possibly any mobile device) to generate location query and send it to contacted memory spot as indicated in Algorithm 5.2. For memory spots, we programmed three applications using C++ to receive location information file sent by readers and update local location information file accordingly, receive location query generated by a mobile reader and either fetch required location if available and send it back to the requestor or save the query in location query database to be pushed to other contacted mobile reader.

**Experiment scenarios and results:**

In our experiment, we evaluate the performance of the system while considering different scenarios in terms of mobile readers' scan period, how frequent mobile readers synchronize their data with memory spots and how frequent mobile readers may pass through memory spots to carry and forward location information and/or location queries. In the experiment we distribute *50* tagged-objects on *18* cubicles over *3* labs. Then, we allow *3* mobile readers to move through the *3* labs (*1* mobile reader per lab) using pedestrian speed, stop for the scan period at each cubicle and send the generated location information to the memory spot close to the lab every synchronization interval. In addition, we allow each mobile reader to generate a location query asking for a randomly selected tag at every synchronization interval. For evaluation, we measure the following metrics:

Localization quality: this metric consists of three complement percentages; the percentage of *correctly-localized* tags (the estimated location is same as the actual location), the percentage of *incorrectly-localized* tags (the estimated location is not the same as actual location[1]) and the percentage of *un-localized* tags (the tags are not interrogated at all).

Localization delay: This metric represents the average time it takes for a tag location to be distributed on the *3* memory spots. We measure this time for each tag of interest as the difference between its query time and the maximum of its update times at the 3 memory spots. Then we take the average over all tags of interest (*correctly-* and *incorrectly-localized*).

Average Overhead: the number of messages exchanged amongst mobile readers and memory spots to disseminate the location of tags of interest. We measure this metric by counting the messages exchanged until all locations of tags of interest are disseminated and take the average over all localized tags (*correctly-* and *incorrectly-localized*).

The results are shown in Table 5-2 where we run the experiment for *10* times and take the average. As indicated in the table, inflating the scan period decreases the percentage of un-localized tags but on account of the percentage of incorrectly localized tags. This is due to that standing longer at each cubicle may allow the mobile reader to interrogate more tags including tags close to the edge of neighboring cubicles. However, the high ratio of un-localized tags is for cubicles close to the work area for wireless sensor & RFID ($L_1C_2$, $L_1C_3$, $L_1C_5$ and $L_1C_6$) as shown in Figure 5.16, which contains many unknown tags resulting in high collision. In addition, increasing the synchronization frequency positively affects the localization delay on account of

[1] If two reference tags are detected during the same scan period, the more recent detected one is considered in localizing other interrogated tags which may result in *incorrectly-localized* tags.

the average overhead which conforms to the simulation results. The table also shows that the passing frequency adversely affects the average overhead, with lower overhead when synchronization is less frequent.

**Table 5-2: Testbed experimental results**

| Scan period (in seconds) | Localization Quality (in percentage) | | | Synchronization frequency (in seconds) | Passing frequency (in seconds) | Localization Delay (in seconds) | Average Overhead (in # of messages) |
|---|---|---|---|---|---|---|---|
| | Correctly localized | Incorrectly localized | Un-localized | | | | |
| 5 | 73% | 4% | 23% | 10 | 60 | 139.39 | 12.57 |
| | | | | | 120 | 194.57 | 16.52 |
| | | | | 20 | 60 | 142.78 | 7.40 |
| | | | | | 120 | 199.18 | 9.03 |
| 10 | 77% | 11% | 12% | 10 | 60 | 158.51 | 16.22 |
| | | | | | 120 | 222.65 | 20.79 |
| | | | | 20 | 60 | 162.89 | 9.55 |
| | | | | | 120 | 226.40 | 11.53 |

## 5.6 Discussion and Conclusion

Although centralized RFID systems typically are more robust, they require fixed and coordinated infrastructure and suffer from limited scalability; rendering them less suitable for large and dynamic environments such as IoT scenarios. In these scenarios, distributed solutions especially solutions based on crowdsourcing may provide an acceptable level of robustness, while being scalable and less expensive. In this chapter, we proposed two strategies to cooperatively disseminate location information among participants for distributed RFID-based localization systems. The first strategy is flooding-based, where participants directly communicate with each other to exchange location information in a distributed manner. As a flooding-based strategy, its performance is significantly worse at large scale and/or high dynamicity settings. The second proposed strategy offers an indirect cooperative dissemination where participants, thanks to a cost effective minimal infrastructure, do not have to directly communicate with one another. In this strategy, participants proactively disseminate their information over “memory spots” distributed in smart areas and reactively carry and forward location queries and responses; providing high location information availability with less overhead as concluded through our extensive simulation experiments. The availability of memory spots in smart areas can become a reality by considering the market awareness of the business values generated by deploying smart building solutions (residential or commercial). However, in the absence of such “memory spots”, a cost effective alternative is a semi-passive RFID tag, which has a relatively long read/write range compared with a typical passive RFID tags, on-board processor and data storage. Mobile devices, according to their different wireless communication capabilities, can use compatible memory spots to disseminate location information or location queries. As a conclusion, in dynamic and mobile IoT environments there are abundant crowd resources that can be leveraged to provide scalable distributed solutions where centralized and/or fixed infrastructure-based approaches are infeasible.

# Chapter 6

# Summary and Conclusions

The vision of IoT is a world where every object or “thing” has the ability to interact with its surrounding environment; forming pervasive computing environments. IoT builds upon many technologies amongst which RFID stands at the forefront for the purposes of object identification and tracking. IoT applications span a wide and diverse range of domains such as transportation, healthcare, and smart environments to offer users more convenient context-aware and location-aware services. For context information to be useful, and for enabling location-based services, the ability to locate objects is essential. To accurately estimate a location of objects in this dynamic distributed environment and make such location information accessible is a big challenge which is addressed by this thesis. Solutions to this problem are proposed based on different technologies such as UWB, Infrared, Ultrasound, and RFID which can provide localization services but on account of expensive infrastructure deployed especially to provide localization. In this thesis we devise two distributed crowdsourcing schemes which leverage the available RFID resources in terms of mobile readers and RFID tags to provide a localization service. In addition we propose a localization technique to enhance location accuracy when the available information used in location estimation is not sufficient. Finally, we design two distributed information dissemination techniques based on none or minimal infrastructure.

## 6.1 Summary

In Chapter 3, we devised two distributed cooperative localization schemes: ReaDS and RICTags. Our first scheme, ReaDS, depends on the mobile readers’ ability to communicate with each other to share spatial information about surrounding objects within a one-hop neighborhood. Each reader periodically estimates the locations of its surrounding tags using collected and exchanged spatial information. Our second scheme releases the dependency on direct communication amongst readers and utilizes tags' residual memory instead. In this scheme, the cooperation takes place through storing spatial information on tags' memory which can be retrieved by other passing readers to estimate tags' location and write them back on their memories.

In Chapter 4, we studied the problem of accurately localizing a passive mobile object when the available concurrent spatial information about that object is not sufficient, which is a common challenge in dynamic and distributed environments. To overcome this challenge, we proposed a Time-Shifted Multilateration technique in which asynchronous spatial information is shifted based on object speed and time differences to provide better location accuracy. TSM starts by estimating the objects' speed based on old locations and uses such speed to estimate the current location of such an object. Our simulation experiments show that, recursively, TSM is able to accurately estimate object speed and to enhance location accuracy in highly dynamic scenarios.

In Chapter 5, we addressed the problem of maintaining location information availability among system participants through two different distributed information dissemination strategies. The first strategy is a reactive one where mobile readers "gossip" amongst each other their location queries/responses with no need for any supportive infrastructure. However for large scale scenarios and/or when readers do not communicate with each other, we proposed the use of memory spots. Memory spots are inexpensive and flexible components deployed in a given environment to be used by readers to indirectly disseminate location information. Both simulation and field experiments indicate that the use of memory spots is a promising direction not only for location information availability but for other smart applications in various domains.

## 6.2 Limitations and Future Directions

We identify two limitations of this research work. First, the proposed solutions are based on the penetration of RFID crowdsourcing represented in handheld RFID readers, RFID readers embedded in mobile devices and passive tags with considerable memory. Second, we assume that mobile RFID readers are capable of acquiring their locations at any given time.

For the first aspect, and due to the great interest of RFID manufacturers, RFID development achieved unceasing technical progress in addition to cost reductions and standardization in the past few years. Examples of such technical progress as explained in Section 3.6 are the rapid advancements in antenna design for handheld RFID readers which result in longer reading and writing ranges. In addition, readers are supported with Wi-Fi and Bluetooth which allow them to communicate each other and/or with other wireless devices. Tags, even the passive ones, became more capable to store data in addition to their unique identifiers (e.g., Tego Launches 32-Kilobyte EPC RFID tag). Consequently, typical expectations are: (1) objects can be easily identified by passive RFID tags, which are inexpensive, and widely available, (2) embedded RFID readers in mobile devices will be rapidly adopted, and (3) Tags' memories will play a significant role in data exchange.

For the second aspect, although knowing the position of mobile readers is a challenge, GPS-based positioning, coupled with street maps, is used in outdoor environments with typical accuracies of 1-3 meters. While indoors, where GPS signals are no longer available, wireless technologies such as WiFi, Ultra Wide Band (UWB), Ultrasonic, or RFID can be used for positioning, providing meter-level accuracy. From our simulation experiments explained in Chapter 3, we conclude that although errors in mobile readers' position affect the object location accuracy, the effect is not aggressive.

Our work opened several research directions. For example, in Chapter 3 mobile readers cooperate directly or indirectly to localize surrounding objects while they move. An interesting direction is to investigate their mobility pattern and study how it may positively or negatively affect location accuracy, and to figure out how we can benefit from the relation between their mobility and objects' mobility for better location accuracy. In Chapter 4, TSM considered object speed to enlarge asynchronous detections irrespective to the object moving direction. We plan to involve the objects' moving direction to either enlarge or shrink detections based on such estimated direction and study the behavior of TSM accordingly. Another direction is to further investigate the impact of error in mobile readers' position on location accuracy and derivate a mathematical model to formulate such an effect. We proposed the use of memory spots in Chapter 5. Future work could investigate the parameters that control the optimal deployment of memory spots in a given area and to study how these parameters are correlated. We assumed that memory spots can only communicate with mobile readers which are the initiator of the synchronization process. It would be interesting to investigate the case where memory spots can communicate with one another and/or are able to trigger the synchronization event at need.

## 6.3 Concluding Remarks

The tremendous proliferation of mobile devices, along with the adoption of different wireless capabilities such as WiFi, Bluetooth, and RFID in such devices, originates the concept of wireless crowdsourcing in distributed and/or ad hoc environments. The use of wireless crowdsourcing to provide a variety of services is a promising approach especially when having a centralized and/or fixed infrastructure is infeasible. Several questions must be considered though. Do we have enough participants to provide the service? Are the participants intended to directly cooperate with one another to achieve a common goal or do they have to cooperate indirectly? Does the heterogeneity of participants hinder their cooperation? Our work has shown that the use of RFID crowdsourcing coupled with the deployment of a simple storage and information exchange component (memory spots) can provide an effective and practical object localizations service.

Our work has shown that the number of available crowdsourcing participants affects the accuracy of the localization service. When resources are sparse, some techniques such as TSM can be used to limit such effect. In some scenarios, proximity location is sufficient which means that only one, current detection of the object of interest is enough to determine the area where it is currently located. When accurate positioning is desired, in the absence of sufficient concurrent detections, TSM can calculate the current location of an object based on estimated objects' speed and old detections. If objects move in pathways with known width and direction, this knowledge can be coupled with TSM to estimate more accurate locations.

We conclude by remarking that utilizing crowdsourcing participants along with knowledge of their behavior can be used to provide a multiplicity of services with no need for expensive and/or central infrastructure.

# Bibliography

[1] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the Internet of Things," Internet Computing, IEEE, vol. 14, no. 1, pp. 44-51, 2010.

[2] E. Kosmatos, N. Tselikas and A. Boucouvalas, "Integrating RFIDs and Smart Objects into a Unified Internet of Things Architecture," Advances in Internet of Things, vol. 1 no. 1, pp. 5-12, 2011.

[3] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A Survey,” Computer Networks, vol. 54, no. 15, pp. 2787-2805, 2010.

[4] S. Haller, S. Karnouskos, and C. Schroth. “The Internet of Things in an Enterprise Context,” In Future Internet FIS 2008, vol. 5468 of Lecture Notes in Computer Science, pages 14-28, 2009.

[5] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, “Internet of things: Vision, applications and research challenges,” Ad Hoc Networks, vol. 10, no. 7, pp. 1497-1516, 2012.

[6] G. Han, H. Xu, T. Q. Duong, J. Jiang, T. Hara, “Localization algorithms of Wireless Sensor Networks: a survey,” Telecommunication Systems, vol. 52, Issue 4, pp 2419-2436, 2013.

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

[8] Y. Liu, Z. Yang, X. Wang, and L. Jian, “Location, localization, and localizability,” Journal of Computer Science and Technology, vol. 25, pp. 274-297, 2010.

[9] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Win, “Ranging with ultrawide bandwidth signals in multipath environments,” Proceedings of the IEEE, vol. 97, no. 2, pp. 404-426, 2009.

[10] D. Zhang, F. Xia, Z. Yang, L. Yao and W. Zhao, "Localization Technologies for Indoor Human Tracking," International Conference on Future Information Technology, vol. 5, pp. 1-6, 2010.

[11] J. Torres-Solis, T.H. Falk, T. Chau, “A review of indoor localization technologies: towards navigational assistance for topographical disorientation,” Ambient Intelligence, pp. 51-84, 2010.

[12] Q. Yang, S.J. Pan and V.W. Zheng, "Estimating Location Using Wi-Fi," IEEE Intelligent Systems, vol. 23, no. 1, pp. 8-13, 2008.

[13] Z. Junyi, and S. Jing, “RFID localization algorithms and applications—a review,” Journal of Intelligent Manufacturing, Vol. 20, pp. 695-707, 2009.

[14] E. Welbourne "Building the Internet of Things Using RFID: The RFID Ecosystem Experience", IEEE Internet Computing, vol. 13, no. 3, pp.48-55, 2009.

[15] B. Khoo, "RFID- from Tracking to the Internet of Things: A Review of Developments," in Proc. IEEE/ACM I nt’l Conference on Green Computing and Communications & Int’l Conference on Cyber, Physical and Social Computing, ser. GREENCOM-CPSCOM, pp.533-538, 2010.

[16] EPC Global, Class 1 Generation 2 “UHF air interface protocol standard”, v1.2, [Accessed online] http://www.epcglobalinc.org, 2008.

[17] K. Ahsan, H. Shah and P. Kingston, "RFID applications: An introductory and exploratory study," IJCSI International Journal of Computer Science Issues, vol. 7, Issue 1, no. 3, 2010.

[18] M. Bouet, A. L. dos Santos, "RFID tags: Positioning principles and localization techniques", 1st IFIP Wireless Days, pp.1-5, 2008.

[19] T. Sanpechuda and L. Kovavisaruch "A review of RFID localization: Applications and techniques", Proc. 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, vol. 2, pp.769-772, 2008.

[20] P. V. Nikitin and K. V. S. Rao "Helical antenna for handheld UHF RFID reader", Proc. IEEE Internationa Conference on RFID, pp.166-173, 2010.

[21] Q. Liu, J. Shen; H. Liu, Y. Wu, M. Su, Y. Liu, "Low-cost compact circularly polarised directional antenna for universal UHF RFID handheld reader applications," IEEE Antennas and Wireless Propagation Letters, vol. 14, no.99, pp.1326-1329, 2015.

[22] I. Farris, A. Iera, and S. C. Spinella, “A novel paradigm to exchange data in RFID piconets,” IEEE International Conference on RFID, pp.215-222, 2013.

[23] M. Zorzi et al., “From Today’s INTRAnet of Things to A Future INTERnet of Things: a Wireless and Mobility-Related View,” IEEE Wireless Communication, vol. 17, no. 6, pp. 44-51, 2010.

[24] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” Computer Network, vol. 54, no. 15, pp. 2787-2805, 2010.

[25] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (IoT): A vision, architectural elements, and future directions,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1645-1660, 2013.

[26] A. Whitmore, A. Agarwal, and L. D. Xu, "The Internet of Things-A survey of topics and trends," Information Systems Frontiers, pp. 1-14, 2015.

[27] S. Li, L. Xu and S. Zhao, "The internet of things: A survey", Information Systems Frontiers, vol. 17, Issue 2, pp 243-259, 2015.

[28] C. Perera , A. Zaslavsky , P. Christen and D. Georgakopoulos "Context aware computing for the Internet of Things: A survey", IEEE Communication Surveys & Tutorials, no. 99, pp.1-41, 2013.

[29] Z. Song, M. T. Lazarescu, R. Tomasi and L. Lavagno, “High-Level Internet of Things Applications Development Using Wireless Sensor Networks,” in Internet of Things: Challenges and Opportunities, Switzerland, Springer International Publishing, pp. 75-109, 2014.

[30] L. Da Xu; W. He; S. Li, "Internet of Things in Industries: A Survey," IEEE Transactions on Industrial Informatics, vol.10, no.4, pp.2233-2243, 2014.

[31] R. Want, “An Introduction to RFID Technology,” IEEE Pervasive Computing, vol. 5, no. 1, p. 25-33, 2006.

[32] P. V. Nikitin and K. V. S. Rao, “Theory and measurement of backscattering from RFID tags”, IEEE Antennas and Propagation Magazine, vol. 48, no. 6, pp. 212-218, 2006.

[33] H. Saleem, M. Khan, S. Afzal, “Review of Various Aspects of Radio Frequency Identification (RFID) Technology”, IOSR Journal of Computer Engineering (IOSRJCE), vol. 8, Issue 1, pp1-6, 2012.

[34] Y. Duroc and D. Kaddour, “RFID potential impacts and future evolution for green projects,” Energy Proc., vol. 18, pp. 91-98, 2012.

[35] P.M. Senadeera, N.S. Dogan, X. Zhijian, H.S. Savci, I. Kateeb, M. Ketel, "Recent trends in RFID transponders," Southeastcon, 2013 Proceedings of IEEE , pp.1-5, 2013.

[36] DL. Wu, W.W.Y. Ng, D.S. Yeung, HL. Ding, "A brief survey on current RFID applications," International Conference on Machine Learning and Cybernetics, vol.4, pp. 2330-2335, 2009.

[37] A. Costanzo, D. Masotti, T. Ussmueller, R. Weigel, "Tag, You're It: Ranging and Finding via RFID Technology," IEEE Microwave Magazine, vol.14, no.5, pp.36-46, 2013.

[38] C. Sun, “Application of RFID Technology for Logistics on Internet of Things,” AASRI Conference on Computational Intelligence and Bioinformatics, Elsevier B.V., pp. 106-111, 2012.

[39] M. Kaur, M. Sandhu, N. Mohan, and P.S. Sandhu, "RFID Technology Principles, Advantages, Limitations & Its Applications," in International Journal of Computer and Electrical Engineering, vol.3, no.1, 2011.

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

[41] G. Sun, J. Chen, W. Guo, and K. Liu, “Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs,” Signal Processing Magazine, IEEE, vol. 22, no. 4, pp. 12-23, 2005.

[42] S. Gezici, “A survey on wireless position estimation,” Wireless Personal Communications, vol. 44, pp. 263-282, 2008.

[43] M. Bal, M. Liu, W. Shen, and H. Ghenniwa, “Localization in cooperative wireless sensor networks: A review,” International Conference on Computer Supported Cooperative Work in Design, CSCWD. pp. 438-443, 2009.

[44] Y. Liu, Z. Yang, X. Wang, and L. Jian, “Location, localization, and localizability,” Journal of Computer Science and Technology, vol. 25, pp. 274-297, 2010.

[45] J. Wang, R. Ghosh, and S. Das, “A survey on sensor localization,” Journal of Control Theory and Applications, vol. 8, pp. 2-11, 2010.

[46] D. Niculescu, “Positioning in ad hoc sensor networks,” Network, IEEE, vol. 18, no. 4, pp. 24-29, 2004.

[47] J. Bachrach and C. Taylor, “localization in Sensor Networks,” in Handbook of Sensor Networks: Algorithms and Architectures, 2005.

[48] D. Zhang, F. Xia, Z. Yang, L. Yao and W. Zhao, "Localization Technologies for Indoor Human Tracking,", International Conference on Future Information Technology, vol. 5, 2010.

[49] Bluetooth Technology, Internet: http://www.bluetooth.com, [Last accessed May, 2015].

[50] Wi-Fi Alliance, Internet: http://www.wi-fi.org, [Last accessed May, 2015].

[51] ZigBee Alliance, Internet: http://www.zigbee.org, [Last accessed May, 2015].

[52] I. Guvenc and C.-C. Chong, “A survey on TOA based wireless localization and NLOS mitigation techniques,” IEEE Communications Surveys & Tutorials, vol. 11, no. 3, pp. 107-124, 2009.

[53] I. Amundson and X. Koutsoukos, “A survey on localization for mobile wireless sensor networks,” in 2nd International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT), 2009.

[54] A. Sayed, A. Tarighat, and N. Khajehnouri, “Network-based wireless location: challenges faced in developing techniques for accurate wireless location information,” Signal Processing Magazine, IEEE, vol. 22, no. 4, pp. 24-40, 2005.

[55] A. Kulaib, R. Shubair, M. Al-Qutayri, and J. Ng, “An overview of localization techniques for wireless sensor networks,” International Conference on Innovations in Information Technology (IIT), pp. 167-172, 2011.

[56] P. Bergamo and G. Mazzini, “Localization in sensor networks with fading and mobility,” 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, volume 2, pp. 750-754, 2002.

[57] E. Elnahrawy, X. Li, R. P. Martin, “The limits of localization using signal strength: A comparative study,” 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), pp. 406-414, 2004.

[58] T. Rappaport, “Wireless Communications: Principles and Practice,” Prentice Hall PTR, 2nd edition, 2001.

[59] L.W. Yeh, M.S. Hsu, Y.F. Lee, and Y.C. Tseng, “Indoor localization: Automatically constructing today's radio map by iRobot and RFIDs,” IEEE Sensors Journal, pp. 1463-1466, 2009.

[60] L. Xu, F. Yang, Y. Jiang, L. Zhang, C. Feng, and N. Bao, “Variation of received signal strength in wireless sensor network,” 3rd International Conference on Advanced Computer Control (ICACC), pp. 151-154, 2011.

[61] H. Laitinen, S. Juurakko, T. Lahti, R. Korhonen, and J. Lahteenmaki, “Experimental evaluation of location methods based on signal-strength measurements,” IEEE Transactions on Vehicular Technology, vol. 56, no. 1, pp. 287-296, 2007.

[62] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Win, “Ranging with ultrawide bandwidth signals in multipath environments,” IEEE, vol. 97, no. 2, pp. 404-426, 2009.

[63] H. Liu, H. Darabi, P. Banerjee, J. Liu, “Survey of Wireless Indoor Positioning Techniques and Systems,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 37, no. 6, pp. 1067-1080, 2007.

[64] J.Y. Lee and R. A. Scholtz, “Ranging in a dense multipath environment using an UWB radio link,” IEEE Journal on Selected Areas in Communications, vol. 20, no. 9, pp. 1677-1683, 2002.

[65] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, “Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks,” IEEE Signal Processing Magazine, vol. 22, no. 4, pp. 70-84, 2005.

[66] H. Wymeersch, J. Lien, and M. Win, “Cooperative localization in wireless networks,” IEEE, vol. 97, no. 2, pp. 427-450, 2009.

[67] I. Guvenc, C.C. Chong, “A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques,” IEE Communications Surveys &Tutorials, vol. 11, no. 3, pp. 107-124, 2009.

[68] G. Shen, R. Zetik, R. Thoma, “Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment,” 5th Workshop on Navigation and Communication (WPNC), pp. 71-78, 2008.

[69] P. Rong, M.L. Sichitiu, “Angle of arrival localization for wireless sensor networks,” 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON), vol. 1, pp. 374-382, 2006.

[70] Y. Zhu, D. Huang, and A. Jiang, “Network localization using angle of arrival,” IEEE International Conference on Electro/Information Technology (EIT), pp. 205-210, 2008.

[71] P. Rong, M. Sichitiu, “Angle of arrival localization for wireless sensor networks,” 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, (SECON), vol. 1, pp. 374-382, 2006.

[72] W. Greene. Econometric Analysis. Prentice Hall, 7th edition, 2011.

[73] G. Kuruoglu, M. Erol, S. Oktug, "Localization in wireless sensor networks with range measurement errors," Advanced International Conference on Telecommunications, pp. 261-266, 2009.

[74] Bachrach and C. Taylor. Handbook of Sensor Networks, chapter Localization in Sensor Networks, Wiley, 2005.

[75] K. Kaemarungsi, P. Krishnamurthy, “Modeling of indoor positioning systems based on location fingerprinting,” 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) , vol. 2, pp. 1012-1022, 2004.

[76] V. Ramadurai and M. L. Sichitiu, “Localization in wireless sensor networks: A probabilistic approach,” *ICWN*, pp. 275-281, 2003.

[77] A. Bekkali, H. Sanson, M. Matsumoto, “RFID indoor positioning based on probabilistic RFID map and Kalman filtering”. Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMOB), 2007.

[78] J. Huanxiang, W. Yong, T. Xiaoling, “Localization algorithm for mobile anchor node based on genetic algorithm in wireless sensor network,” International Conference on Intelligent Computing and Integrated Systems (ICISS), pp. 40-44, 2010.

[79] D-K. Kim, Y-S. Kim, J-W. Chong, “Hybrid RSS/TOA wireless positioning with a mobile anchor in wireless sensor networks,” 6th Latin America Networking Conference, pp. 25-32, 2011.

[80] K-F. Ssu, C-H. Ou, H. C. Jiau, “Localization with mobile anchor points in wireless sensor networks,” IEEE Transactions on Vehicular Technology, vol. 54, no. 3 pp.1187-1197, 2005.

[81] T. Camp, J. Boleng, V. Davies, "Mobility models for ad hoc network simulations," Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2002

[82] C-H. Ou and W-L. He, “Path planning algorithm for mobile anchor-based localization in wireless sensor networks,” IEEE Sensors Journal, vol. 13, no. 2, pp. 466-475, 2013.

[83] M. L. Sichitiu, V. Ramadurai, “Localization of wireless sensor networks with a mobile beacon,” IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), pp. 174-183, 2004.

[84] D. Koutsonikolas, S. M. Das, Y. C. Hu, “Path planning of mobile landmarks for localization in wireless sensor networks,” Computer Communication, vol. 30, no. 13, pp. 2577-2592, 2007.

[85] R. Huang, G. V Zaruba, “Static path planning for mobile beacons to localize sensor networks,” 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom), pp. 323-330, 2007.

[86] Z. Hu, D. Gu, Z. Song, H. Li, “Localization in wireless sensor networks using a mobile anchor node,” IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 602-607, 2008.

[87] M. Bouet, G. Pujolle, “3-D localization schemes of RFID tags with static and mobile readers,” NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, vol. 4982, pp. 112-123, 2008.

[88] H.J. Lee, M.C. Lee, "Localization of Mobile Robot Based on Radio Frequency Identification Devices," SICE-ICASE, International Joint Conference, pp. 5934-5939, 2006.

[89] C. Hekimian-Williams, B. Grant, P. Kumar, "Accurate localization of RFID tags using phase difference," IEEE International Conference on RFID, pp. 89-96, 2010.

[90] J. Hightower, R. Want, G. Borriello, “SpotON: An Indoor 3d Location Sensing Technology Based on RF Signal Strength,” University of Washington CSE Technical Report #2000-02-02, 2000.

[91] T.F. Bechteler, H. Yenigun, “2-D localization and identification based on SAW ID-tags at 2.5 GHz,” IEEE Transaction. on Microwave Theory and Techniques, vol. 51, no. 5, pp. 1584-1590, 2003.

[92] L. Eslim, W. Ibrahim, H.S. Hassanein, “GOSSIPY: A Distributed Localization System for Internet of Things using RFID Technology,” IEEE Global Communications Conference (GLOBECOM), pp. 162-167, 2013.

[93] Eslim, L. M., H. S. Hassanein, W. M. Ibrahim, A. Alma'aitah, "A cooperative localization scheme using RFID crowdsourcing and time-shifted multilateration", IEEE 39th Conference on Local Computer Networks (LCN), pp. 185-192, 2014.

[94] S. Han, H.-S. Lim, J.-M. Lee, “An efficient localization scheme for a differential-driving mobile robot based on RFID system,” IEEE Transactions on Industrial Electronics, vol. 54, no. 6, pp. 3362-3369, 2007.

[95] C. Wang, H. Wu, N.-F. Tzeng, “RFID-based 3-D positioning schemes,” IEEE INFOCOM, pp. 1235-1243, 2007.

[96] W. Zhu; J. Cao; Y. Xu; L. Yang; J. Kong, "Fault-tolerant RFID reader localization based on passive RFID tags," IEEE INFOCOM, pp. 2183-2191, 2012.

[97] V. Ekambaram, K.Ramchandran, "SLAC-RF: Simultaneous 3D Localization of mobile readers and Calibration of RFID supertags," EECS Department, University of California, Berkeley, Technical Report UCB/EECS-2012-188, 2012.

[98] L. M. Ni , Y. Liu , Y. C. Lau, A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," First IEEE International Conference on Pervasive Computing and Communications, vol. 10, no. 6, pp. 701-710, 2004.

[99] G. Jin, X. Lu, M. S. Park, "An Indoor Localization Mechanism Using Active RFID Tag," IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing , vol. 1, pp. 40-43, 2006.

[100] Y. Zhao, Y. Liu, L.M. Ni, "VIRE: Active RFID-based localization using virtual reference elimination," International Conference on Parallel Processing (ICPP), 2007.

[101] M. Bouet, G. Pujolle, "L-virt: Range-free 3-d localization of rfid tags based on topological constraints," Computer Communications, vol. 32, no. 13-14, pp. 1485-1494, 2009.

[102] G. Han, H. Xu, J. Jiang, L. Shu, T. Hara, S. Nishio, "Path planning using a mobile anchor node based on trilateration in wireless sensor networks," Wireless Communications and Mobile Computing, 2011.

[103] Internet: http://spectrum.ieee.org/telecom/wireless/new-indoornavigation-technologies, [Last accessed April. 4th 2015]

[104] P. Lazik, A. Rowe, "Indoor pseudo-ranging of mobile devices using ultrasonic chirps," ACM Conference on Embedded Network Sensor Systems, pp. 99-112, 2012.

[105] L. X. zheng, X. Zeming, C. Xuanliang, "A Compact RFID Reader Antenna for UHF Near-Field and Far-Field Operations," International Journal of Antennas and Propagation, 2013.

[106] F. Viani , M. Salucci , F. Robol , G. Oliveri, A. Massa, "Design of a UHF RFID/GPS Fractal Antenna for Logistics Management," Journal of Electromagnetic Waves and Applications, vol. 26, no. 4, pp. 480-492, 2012.

[107] I. Kwon; Y. Eo; H. Bang; K. Choi; S. Jeon; S. Jung; D. Lee; H. Lee, "A Single-Chip CMOS Transceiver for UHF Mobile RFID Reader," IEEE Journal of Solid-State Circuits, vol.43, no.3, pp. 729-738, 2008.

[108] P. V. Nikitin, K. V. S. Rao, "Antennas and propagation in UHF RFID systems", IEEE International Conference on RFID, pp. 277-288, 2008.

[109] S. K. Goudos, K. Siakavara, J. N. Sahalos, "Design of load-ended spiral antennas for RFID UHF passive tags using improved artificial bee colony algorithm", International Journal of Electronics and Communications, vol. 69, Issue 1, pp. 206-214, 2015.

[110] K.V.S. Rao, P.V. Nikitin, S.F. Lam, "Antenna design for UHF RFID tags: A review and a practical application," IEEE Transactions on Antennas & Propagation, vol. 53, no. 12, pp. 3870–3876, 2005.

[111] S. H. Zainud-Deen, H. A. Malhat, K. H. Awadalla, "Fractal antenna for passive UHF RFID," Electromagnetics Research B, vol. 16, pp. 209-228, 2009.

[112] T. Bjorninen, K. E. Delzo, L. Ukkonen, A. Z. Elsherbeni, L. Sydanheimo, "Long range metal mountable tag antenna for passive UHF RFID systems," IEEE RFID Technologies and Applications Conference, pp. 194-198, 2011.

[113] B. D. Braaten, "A novel compact UHF RFID tag antenna design with series connected open complementary split-ring resonators (OCSRR) particles," IEEE Transactions on Antennas Propagation, 2010.

[114] T. Diekmann, A. Melski, M. Schumann, "Data-on-Network vs. Data-on-Tag: Managing Data in Complex RFID Environments," 40th Annual Hawaii International Conference on System Sciences (HICSS'07), 2007.

[115] S. Pais, J. Symonds, "Data storage on a RFID tag for a distributed system," International Journal of UbiComp, vol. 2, no. 2, pp. 26-39, 2011.

[116] A.K. Bashir, S,H. Chauhdary, S.C. Shah, M.S. Park, "Mobile RFID and its design security issues," Potentials, IEEE , vol.30, no.4, pp.34-38, 2011.

[117] A. Juels, "RFID security and privacy: a research survey," IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 381-394, 2006.

[118] P. D’Arco, A. D. Santis, “On ultralight weight RFID authentication protocols,” IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 4, pp. 548-563, 2011.

[119] D.Z. Sun, J.D. Zhong, "A hash-based RFID security protocol for strong privacy protection," IEEE Transactions on Consumer Electronics, vol. 58, no. 4, pp. 1246-1252, 2012.

[120] J. Hermans, A. Pashalidis, F. Vercauteren, B. Preneel, “A new RFID privacy model,” 16th European Symposium on Research in Computer Security (ESORICS ’11), vol. 6879 of Lecture Notes in Computer Science, pp. 568–587, Springer, Leuven, Belgium, 2011.

[121] T. Li, W. Luo, Z. Mo, S. Chen, "Privacy-preserving RFID authentication based on cryptographical encoding," IEEE INFOCOM, , pp. 2174-2182, 2012.

[122] S. Choi, S. Lee, H. Lee, "Security enhanced authentication protocol for UHF passive RFID system", 7th International Conference on Wireless and Mobile Communications (ICWMC), pp. 307-311, 2011.

[123] A. Arbit, Y. Livne, Y. Oren, A. Wool, “Implementing public-key cryptography on passive rfid tags is practical,” International Journal of Information Security, pp. 1–15, 2014.

[124] S. Gezici, “A survey on wireless position estimation,” Wireless Personal Communications, vol. 44, pp. 263–282, 2008,

[125] J. Kuriakose, S. Joshi, R. Vikram, A. Kilaru, “A review on localization in wireless sensor networks,” Advances in Signal Processing and Intelligent Recognition Systems. AISC, vol. 264, pp. 599-610, 2014.

[126] A. Savvides, C.C. Han, M.B. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” 7$^{th}$ annual International Conference on Mobile computing and networking, pp. 166-179, 2001.

[127] J. Albowicz, A. Chen, L. Zhang, “Recursive position estimation in sensor networks,” 9th International Conference on Network Protocols’ICNP01, pp. 35-41, 2003.

[128] J. Liu, Y. Zhang, F. Zhao, “Robust distributed node localization with error management,” 7th ACM international symposium on Mobile ad hoc networking and computing, pp. 250-261, 2006.

[129] M. Li, Y. Liu, “Rendered path: Range-free localization in anisotropic sensor networks with holes,” 13th annual ACM international conference on Mobile computing and networking (Mobicom’07), pp 51-62, 2007.

[130] Y. Wang, K. Li, J. Wu, “Distance estimation by constructing the virtual ruler in anisotropic sensor networks,” 29th Conference on Computer Communications (Infocom’10), pp. 1-9, 2010.

[131] Q. Xiao, B. Xiao, J. Cao, J. Wang, “Multi-hop range-free localization in anisotropic wireless sensor networks: A pattern-driven scheme,” IEEE Transactions on Mobile Computing (TMC), vol. 9, no. 11, pp. 1592-1607, 2010.

[132] B. Xiao, L. Chen, Q. Xiao, M. Li, “Reliable anchor-based sensor localization in irregular areas,” IEEE Transactions on Mobile Computing (TMC), vol. 9, no. 1, pp. 92-102, 2010.

[133] S. Zhang; J. Wang; X. Liu; J. Cao, "Range-free selective multilateration for anisotropic wireless sensor networks," 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), pp. 299-307, 2012.

[134] R.E. Kalman, “A new approach to linear filtering and prediction problems,” Journal of Basic Engineering vol. 82, no. 1, pp. 35-45, 1960.

[135] A. Curran, K.J. Kyriakopoulos, “Sensor-based self-localization for wheeled mobile robots,” Journal of Robotic Systems vol. 12, no. 3, pp. 163-176, 1995.

[136] L. Jetto, S. Longhi, G. Venturini, “Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots,” IEEE Transactions on Robotics and Automation vol. 15, no. 2, pp. 219-229, 1999.

[137] D. Fox, W. Burgard, S. Thrun, “Active markov localization for mobile robots,” Robotics and Autonomous Systems vol. 25, no. 3, pp. 195-207, 1998.

[138] F. Dellaert, D. Fox, W. Burgard, S. Thrun, “Monte carlo localization for mobile robots”. IEEE International Conference on Robotics and Automation, vol. 2, pp. 1322-1328,.1999.

[139] A. Howard, M.J. Matark, G. Sukhatme, “Localization for mobile robot teams using maximum likelihood estimation,” IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 434-439, 2002.

[140] J. Tian, J. Hahner, C. Becker, I. Stepanov, K. Rothermel, “Graph-based Mobility Model for Mobile Ad Hoc Network Simulation,” 35th Annual Simulation Symposium, in cooperation with the IEEE Computer Society and ACM, 2002.

[141] A. Beck, P. Stoica, J. Li, “Exact and approximate solutions of source localization problems,” IEEE Transactions on Signal Processing, vol. 56, no. 5, pp. 1770-1778, 2008.

[142] M. R. Gholami, E. G. Str¨om, H. Wymeersch, M. Rydstr¨om, “Upper bounds on position error of a single location estimate in wireless sensor networks,” EURASIP Journal on Advances in Signal Processing no. 1, 2014.

[143] M. Zarifneshat, P. Khadivi, “Using mobile node speed changes for movement direction change prediction in a realistic category of mobility models,” Journal of Network and Computer Applications, vol. 36, Issue 3, pp. 1078-1090, 2013.

[144] C. Holt, “Forecasting seasonals and trends by exponentially weighted moving averages,” International Journal of Forecasting, vol. 20, pp. 5-10, Issue 1, 2004.

[145] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu, "Prediction and indexing of moving objects with unknown motion patterns," in SIGMOD, pp. 611-622, 2004.

[146] H. Jeung, Q.L. Tasmanian, H.T. Shen, X. Zhou, “A Hybrid Prediction Model for Moving Objects,” IEEE 24th International Conference on Data Engineering (ICDE), pp. 70-79, 2008.

[147] H. Wei, Q. Wan, Z. Chen, S. Ye, “A novel weighted multidimensional scaling analysis for time-of-arrival-based mobile location,” IEEE Transactions on Signal Processing, vol. 56, no. 7, pp. 3018-3022, 2008.

[148] G. Kuruoglu, M. Erol, S. Oktug, "Localization in wireless sensor networks with range measurement errors," Advanced International Conference on Telecommunications, pp. 261-266, 2009.

[149] ns-3 [Online]. Available: http://www.nsnam.org/

[150] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, "Directed diffusion for wireless sensor networking", IEEE/ACM Transactions on Networking, vol. 11, pp.2-16, 2002.

[151] F. Ye, "A Two-Tier Data Dissemination Model for Large-Scale Wireless Sensor Networks", ACM/IEEE MOBICOM, 2002.

[152] F. Ye, G. Zhong, L. Zhang, “Gradient broadcast: A robust data delivery Protocol for large scale sensor networks,” Wireless Networks vol. 11, pp. 285-298, 2005.

[153] C. Hua, T.-S.P. Yum, "Optimal Routing and Data Aggregation for Maximizing Lifetime of Wireless Sensor Networks," IEEE/ACM Transactions on Networking, vol. 16, no. 4, pp. 892-903, 2008.

[154] Y. Wu , S. Fahmy, N. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm," IEEE INFOCOM, pp. 356-360, 2008.

[155] J. Liang, J. Wang, J. Cao, J. Chen, M. Lu, “An Efficient Algorithm for Constructing Maximum Lifetime Tree for Data Gathering without Aggregation in Wireless Sensor Networks,” IEEE INFOCOM Mini-Conf. ’10, pp. 1-5, 2010.

[156] X. Xu, J. Luo, and Q. Zhang, "Delay tolerant event collection in sensor networks with mobile sink," IEEE, INFOCOM, 2010.

[157] H. Huang, G. Hu, F. Yu, “Energy-aware geographic routing in wireless sensor networks with anchor nodes,” International Journal of Communication System vol. 26, no. 1, pp. 100-113, 2013.

[158] S. Tian, S.M. Shatz, Y. Yu, J. Li, “Querying Sensor Networks Using Ad-Hoc Mobile Devices: A Two Layer Networking Approach,” Ad Hoc Networks, vol. 7, no. 5, pp. 1014-1034, 2009.

[159] Y. Ishimaru, W. Sun, K. Yasumoto, M. Ito, "DTN-based delivery of word-of-mouth information with priority and deadline," 5th International Conference on Mobile Computing and Ubiquitous Networking, 2010.

[160] T. Spyropoulos, K. Psounis, C. S. Raghavendra, “Spray and wait: An efficient routing scheme for intermittently connected mobile networks,” ACM SIGCOMM workshop on Delay-tolerant networking, 2005.

[161] T. Spyropoulos, K. Psounis, C. S. Raghavendra, “Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility,” 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007.

[162] W. Gao, G. Cao, “On Exploiting Transient Contact Patterns for Data Forwarding in Delay Tolerant Networks,” ICNP, pp. 193-202, 2010.

[163] P. Hui; J. Crowcroft, E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks," IEEE Transactions on Mobile Computing, vol. 10, no. 11, pp.1576-1589, 2011.

[164] Internet: http://www.st.com/web/catalog/mmc/FM76/CL1766/SC1412/PF219471?WT.ac=fp_jun11 [Last accessed April, 2015].

[165] Internet: http://8devices.com/ [Last accessed March 2015].

[166] N. Banerjee, M.D. Corner, B.N. Levine, “An Energy-Efficient Architecture for DTN Throwboxes,” IEEE INFOCOM, 2007.

[167] BeagleBoard-xM Rev C2. http://elinux.org/Beagleboard:BeagleBoard-xM#Rev_C2, 2012, [Online].

[168] T. Spyropoulos , K. Psounis, C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," IEEE/ACM Transactions on Network, vol. 16, no. 1, pp.77-90, 2008.

[169] J. Fan, Y. Chen, P. Du, P. Wang, Y. Sun, “DelQue: A Socially Aware Delegation Query Scheme in Delay-Tolerant Networks,” IEEE Transactions on Vehicular Technology, vol. 60, no. 5, pp. 2181-2193, 2011.

[170] R. Knoblauch, MT. Pietrucha, M. Nitzburg, “Field studies of pedestrian walking speed and start-up time,” Transportation Research Record 1538. Washington (DC): National Research Council, Transportation Research Board, pp. 27-38, 1996.