FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Muhannad Khaled M. Al-Hajri
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Object Tracking Sensor Networks Using Sequential Patterns in an Energy Efficient Prediction Technique

TITRE DE LA THÈSE / TITLE OF THESIS

A. Boukerche
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

G. Wainer

A. El Saddik

Gary W. Slater
Le Doyen de la Faculte des etudes supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Object Tracking Sensor Networks using Sequential Patterns in an Energy Efficient Prediction Technique

by

Muhannad Khaled Al-Hajri

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.Sc. degree in
Computer Science

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-74222-8
Our file Notre référence
ISBN: 978-0-494-74222-8

Abstract

Wireless Sensor Networks applications are attracting more and more research, especially in energy saving techniques/architectures which is the focal point of most researchers in this area. One of the most interesting applications of Wireless Sensor Networks is the Object Tracking Sensor Networks which are used mainly to track certain objects in a monitored area and to report its location to the application's users. This application is a major energy consumer among other Wireless Sensor Networks applications. There have been many techniques that assist in delivering the required data while maintaining a lower energy consumption than the early approaches. Our approach revolves around the ability to predict the objects' future movements in order to track it with the minimum number of sensor nodes, while keeping the other sensor nodes in the network in a sleep mode. Thus, achieving our goals while reducing significantly the network's energy consumption. The prediction technique used in our proposed solution is based on the inherited patterns of the objects' movements in the network and how to utilize data mining techniques, such as Sequential Patterns, in order to predict which sensor node the moving object will be heading next. We propose the Prediction-based Tracking technique using Sequential Patterns (PTSP), which is designed to achieve significant reductions in the energy dissipated by the OTSN network, while maintaining an acceptable missing rate levels. PTSP is tested against basic tracking techniques in order to determine the appropriateness of PTSP in various circumstances. We also test PTSP against some OTSN impacting factors, such as number of tracked objects, object speed, sampling duration and sampling frequency. We also test 3 different missing object recovery mechanisms implemented in PTSP to determine which is the most energy conservative. The experimental results had shown that PTSP outperformed all the other basic tracking techniques and contributed remarkable amounts of savings in terms of energy consumption of the entire network even through different circumstances.

Acknowledgements

I would like to express my heart-felt appreciation and gratitude to my academic supervisor, Professor. Azzedine Boukerche of the School of Information Technology and Engineering (SITE) at the University of Ottawa, for his constant support, continuous encouragement and sincere concern for the wellbeing and progress of every student under his supervision. Professor. Azzedine I'm truly thankful for all your advices and guidance that had allowed me to complete my graduate studies with firm steps towards graduation. You have taught me more than research and science, you have always pushed me and my fellow colleagues to strive for perfection in everything we do, whether it's an academic task or an ordinary activity. I have learned a lot from my tenure here in PARADISE lab, and one of those key lessons was the importance and significance of teamwork which have always gained your utmost support and encouragement, resulting in a high level of synergy between all members of PARADISE lab. Professor. Azzedine your remarkable understanding, genuine care and unlimited support have always overwhelmed me and everyone in PARADISE lab.

My fellow colleagues in PARADISE lab have always been there whenever I needed them and the least I can do is to express my thanks and appreciation to all of them for everything they have done for me. To my research colleague and mentor Dr. Samer Samarah, I'd like to tell you how thankful I'm for everything you have done for me and all the great support I have received from you which truly had made my research path a lot easier than I had anticipated when we discussed the idea the first time. I'd like also to thank all the lab's veterans, especially my dear friend Raed Jarar who had been a great source of support and encouragement in everything I have done and I would never forget his valuable assistance during my simulation implementation phase. Another dear friend is also one of the lab's esteemed veterans, Luqman Ahmad who also had always supported my work and provided me with guidance and advice throughout my graduate studies period.

I'd like also to thank my dear friends and colleagues Ahmad Shadid, Mohammad Huda, Abdulmajeed Al-Khaldi, Mohamad Eid, Ahmad Barghout, Abdulsalam Al-Sunaidi and Mohammed Abdelghani for their support and guidance throughout my academic life in the University of Ottawa. Last but not least, my dear friend and brother Abdulaziz Al-Hamidi who has always been a true brother and a friend since the moment I have stepped a foot in Ottawa till he had to leave after his graduation, Abdulaziz there are no words to thank you and all I can think of now is congratulations on getting married and I hope you have a wonderful life.

Finally, I'd like to thank my sponsoring company, the Saudi Arabian Oil Company (Saudi Aramco), for their unparalleled support and kind cooperation which have made my graduate studies experience an outstanding success.

Muhannad K. Al-Hajri, October 2010.

To

My Father, Mother, Wife and Family

With sincere love and gratitude

Publications

The following publication by the author is relevant to the work in this thesis.

- S. Samarah, M. Al-Hajri, A. Boukerche, "A Predictive Energy Efficient Technique to Support Object Tracking Sensor Networks". Accepted by the *IEEE Transactions on Vehicular Technology 2010.*

Contents

List of Tables

List of Figures

Glossary of Terms

ADC Analog-to-Digital Converter.

AP Access Point.

CH Cluster Head.

CM Continuous Monitoring.

CODA Continuous Object Detection and Tracking Algorithm.

CPU Central Processing Unit.

DAB Drain and Balance.

DAT Deviation Avoidance Tree.

DOT Dynamic Object Tracking.

GG Gabriel Graph.

MAC Medium Access Control.

MANET Mobile Ad hoc NETwork.

MCU Micro-Controller Unit.

MEMS Micro-Electro-Mechanical Systems

Mobicast Mobile Just-in-time Multicasting.

MR Missing Rate.

OCO Optimized Communication Organization.

OTSN Object Tracking Sensor Network.

PES PrEdiction-based Scheme.

PTSP Prediction-based Tracking technique using Sequential Patterns.

QCR Query Cost Reduction.

QoS Quality of Service.

RARE-Area Reduces Area REporting.

RARE-Node Reduction of Active node REdundancy.

RFID Radio-Frequency Identification.

RF Radio Frequency.

RNG Relative Neighborhood Graph.

SM Scheduled Monitoring.

SN Sensor Node

STUN Scalable Tracking Using Networked sensors.

TEC Total Energy Consumption.

WINS Wireless Integrated Network Sensors.

WLAN Wireless Local Area Network.

WSN Wireless Sensor Network.

Chapter 1

Introduction

Major advances in creating cost-effective, energy efficient and versatile Micro-Electro-Mechanical Systems (MEMS) in the last few years has contributed significantly to the vast amount of research and development projects related to the field of Wireless Sensor Networks (WSN) and smart sensors. The Sensor Node (SN) is a very small device which represents the building blocks of the Wireless Sensor Network. These nodes are being produced in a very low-cost and yet with high levels of sophistication in terms of computing power, energy consumption savings and multipurpose functionalities when compared to earlier generations of SN. WSNs are created by deploying a large number of sensor nodes in a certain area, usually called the *monitored region*, for monitoring purposes. These nodes are interconnected and together are used as a monitoring and reporting device to acquire specific types of data as desired by the application requirements. There are many applications and usages of WSNs which will be discussed to a certain extent later on in the following chapter, however our main concern in this thesis is the use of WSNs as an Object Tracking tool for the type of applications where sensing and reporting the whereabouts of moving objects is the main goal of these applications. An Object Tracking Sensor Network (OTSN) is considered one of the most energy consuming applications of WSNs. Many researches have attempted to solve the issue of the energy consumption by addressing the hardware design and how to optimize the physical design of the sensor nodes in order to produce an energy saving sensor node. Another side of the energy conservation in OTSNs was achieved by minimizing the energy consumed by the radio component (RF radio), in which these techniques had attempted to reduce the number of the messages transmitted and received, thus creating an energy saving tracking scheme [9]. However, there has been a very limited focus toward the energy lost by the computing components - Micro-Controller Unit (MCU) - and the sensing components [1]. Although, the energy dissipated by the MCU and the sensing component is less than what is consumed by the radio component, it still represent a significant source of energy consumption in the sensor node. Therefore, our concern in this thesis is to develop an energy efficient OTSN which should make remarkable reductions in the energy consumed by both the MCU and the sensing components.

In this thesis, we will provide a thorough background information in regard to the Wireless Sensor Networks and its challenges, types, applications and the sensor node architecture. Also, a detailed survey was provided regarding the Object Tracking Sensor Networks, which include the OTSN influencing factors and a related work section that details the OTSN categories along with some examples of each category. In addition, a formal definition of the problem and the application requirements for an object tracking applications is introduced. Also, we will present some basic energy-saving techniques to serve as a measuring stick for our proposed tracking technique. Moreover, we will introduce our proposed Prediction-based Tracking technique using Sequential Patterns (PTSP), which should operate in the following manner: (1) reduce the number of sensor nodes involved in the tracking of a moving object, while keeping the rest of the sensor nodes in sleep mode, (2) predict the future movement of the moving object and activate only the nodes where the moving object is heading, (3) in case the object isn't found use recovery techniques to retrieve the missing object. The two main metrics in this research are (1) total energy consumption and (2) missing rate, which will be discussed in detail later in the thesis. Our experiments results had concluded that PTSP managed to secure a significant energy savings in regard to the energy consumed by the MCU and the sensing component.

1.1 Motivations

The most important motivation of this thesis is to explore the possibility to create an energy efficient prediction-based tracking technique which will utilize the inherited patterns of objects' movements in the monitored region as a prediction mechanism which we will call it the sensor sequential patterns. The introduction of the sensor sequential patterns as a mean of prediction represents a new perspective in developing prediction-based tracking techniques, which should allow for more advanced contributions that utilize the capabilities provided by the prediction mechanism using the sensor sequential patterns in addition to the traditional prediction techniques. Other motivations include the evaluation of the effectiveness of the sensor sequential patterns generation techniques and how to improve these methods in order to produce highly accurate and adaptable sequential patterns which results in a high quality object tracking scheme.

1.2 Objectives

The objectives of this thesis are listed below:

1. Develop a prediction technique to generate the sensor sequential patterns in order to be used as a prediction tool which predicts the future movement of an object in the network.

2. Investigate and design a recovery technique in order to re-locate any missing object during the tracking process.

3. The proposed system should be an energy efficient prediction-based tracking technique, which outperforms the basic energy saving tracking technique in terms of energy consumption while keeping an acceptable percentage of the missing rate.

1.3 Contributions

During the course of work in this thesis, we have achieved the following contributions:

1. Designed and implemented a novel technique to generate the sensor sequential patterns, which represent the inherited movement patterns of the moving objects inside the network. The key aspect of this technique is to extract the tri-sensor patterns of each object's movement history and to compare these patterns to other similar patterns in order to determine which has a higher probability of existence.

2. Designed activation mechanisms which are used to determine which sensor nodes will be activated. Only the first activation mechanism was implemented and the other two will be the subject of a future research.

3. Designed and implemented 3 missing object recovery mechanisms. The main difference between these recovery mechanisms is related to which set of sensor nodes will be waken up to collaborate in re-locating the missing object before we move to the second phase of recovery, where the whole network is activated.

1.4 Organization

The remainder of this thesis is organized as follows:

- **Chapter 2: Background and Related Work.**
  This chapter introduces the reader to background information necessary to understand the scope of this thesis and the current state of the research in this topic. A detailed background information regarding the Wireless Sensor Networks is presented, which discusses the issues and challenges concerning WSNs, the types of WSNs along with their applications. A brief overview is provided regarding the sensor node architecture. Also, a brief survey is introduced in regard to the Object Tracking Sensor Networks, which include a background information and discusses

the impacting factors that affect the performance of an OTSN. Then, a classification of the object tracking methods is presented which discusses the related work of other tracking techniques based on the tracking approach. In addition, a concise background was presented regarding the two main data mining approaches related to our tracking techniques, namely the association rules and sequential patterns.

- **Chapter 3: Proposed Prediction-based Tracking Technique: Design and Implementation.**
  This chapter introduces the new proposed prediction-based tracking technique PTSP. First, the problem definition is provided with the assumptions of the application requirements. Moreover, three basic OTSN schemes are introduces and analyzed in terms of their energy consumption. Then, the proposed prediction-based tracking scheme is introduced and explained in details along with its 3 components; (1) sequential patterns generation and prediction model, (2) sensor node activation mechanism and (3) missing object recovery mechanism.

- **Chapter 4: Performance Evaluation and Analysis.**
  This chapter presents the experiments we conducted on the techniques we developed. First, the experiments parameters and metrics is presented. Then, the developed simulation framework is introduced which include the simulation assumptions, sequential patterns derivation and the experiments configuration. Finally, the performance evaluation experiments are presented and explained starting with the simulation settings and later on introducing each experiment along with its graphs and analysis.

- **Chapter 5: Conclusion and Future Work.**
  This chapter summarizes what has been done in this thesis and identifies what has been accomplished, what problems still exist, and sets future research directions.

Chapter 2

Background and Related Work

In this chapter, we present the background information concerning the Wireless Sensor Networks (WSN) and one of its main applications, the Object Tracking Sensor Network (OTSN). This chapter is divided as follows: section 2.1 introduces the WSNs and discusses their related issues and challenges, in addition to the type of WSNs and the WSN applications, and it concludes with a brief description of the sensor node architecture. In section 2.2 the Object Tracking Sensor Networks are introduced along with some background information. Also, the OTSNs impacting factors are discussed and explained. Then, a categorization of the object tracking techniques is introduced which represents the related work of this thesis, since it discusses different approaches of object tracking schemes along with some examples from each tracking approach presented. In section 2.3 we offer a brief introduction to the two main data mining approaches related to the creation of our prediction model, namely Association Rules and Sequential Patterns, where we provide for both data mining methods definitions as well as examples. The concepts of *support* and *confidence* is also presented which are mainly utilized as a mean of understanding the statistical significance of the rules or the patterns produced by each approach [34].

2.1 Wireless Sensor Networks

There have been great advances in the last few years regarding Wireless Technologies that made them highly attractive for many applications especially military and commercial applications. Since wireless networks have great advantages over wired networks, being significantly more flexible towards changes in network topology, provides ad hoc networking, allows network users to become less restricted to be in certain locations and remarkably cost-effective construction with less hassle than creating a wired network. A strongly emerging name in the wireless technologies is the Wireless Sensor Networks which became one of the most researched subject in wireless networks. Wireless Sensor Network (WSN) came to the spot light recently due to major development in the Micro-Electro-Mechanical Systems (MEMS). Where MEMS made the creation of smart sensors more feasible and less expensive in comparison with traditional sensors and since sensors are the building blocks of WSNs that made constructing WSNs more rewarding [3].

In general constructing a WSN requires no infrastructure, since the sensor network is constructed by deploying numbers of sensor nodes, which could range from few tens to thousands of sensor nodes, and depending on the type of the application the sensors' could be deployed based on 2 types of WSN [31]:

- **Unstructured WSN:** Sensor nodes are densely deployed and could be located in a random manner in the monitored area. After deploying the sensors the network will be operating unattended. In this type of WSNs the ability to maintain the network in terms of failures detection and observing network connectivity poses a hard and problematic task [31].

- **Structured WSN:** In this type of WSNs the deployment of some or all the sensors is accomplished according to pre-determined plans which improves network maintainability and lowers the management cost. It's possible also to use less number of sensor nodes since an optimal network coverage is achieved by placing the nodes in specific locations in comparison to the unstructured deployment where some areas could be out of network's coverage [31].

The collaboration of the sensor nodes in the WSN contributes to the network's goal of observing and sensing the environment it's located in and the only method of communication between those nodes are the use of their wireless communication component which allows them also to communicate with the sink node or base station in order to report their raw data after processing it internally and converting it to more useful results [4]. The sink node represents the data collection point which receives the processed data from the sensor nodes and subjects the data to more levels of processing the data in terms of additional aggregation and/or fusion to produce more intelligent and useful results for the application, therefore the sink node could be viewed as a gateway for other networks or users, Figure 2.1 depicts an example of a typical WSN scenario [4] [5].

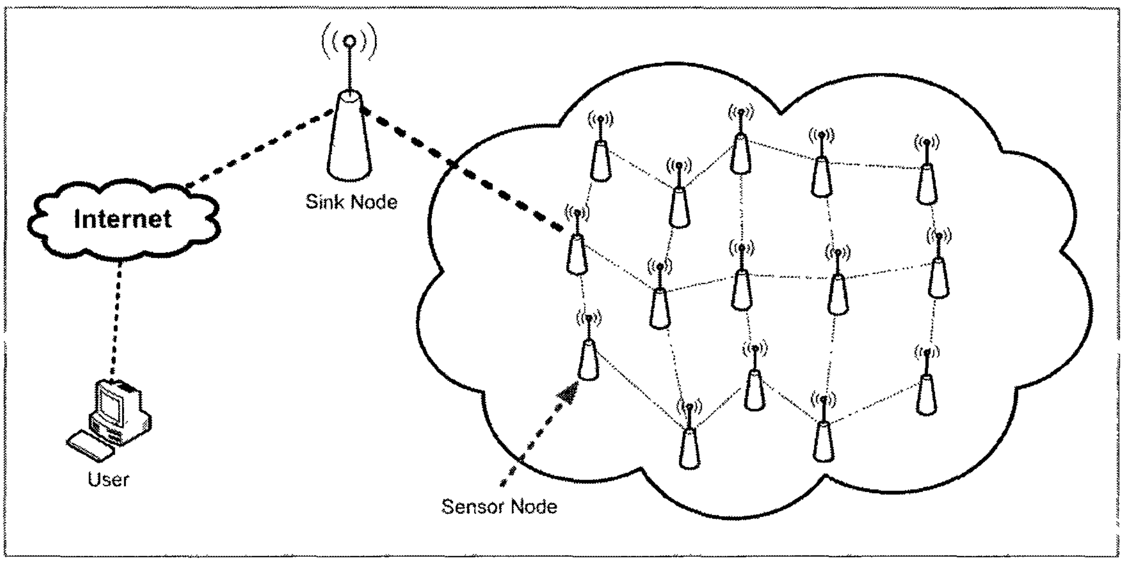

Figure 2.1: An example of a Wireless Sensor Network (WSN)

WSNs has many great usages and possess high potential to provide sophisticated solutions for many real life applications. However due to its unorthodox design and the limitations in the sensor nodes resources, especially the limited power supply since the nodes are equipped with a battery that could run for a short period of time depending on the energy consumption by the node's other components such as the sensor, processor and the communication radio, a WSN faces many issues and challenges that influence its design. The following section will provide a brief overview of the main WSN issues and challenges:

2.1.1 Issues and challenges concerning WSNs

WSNs face substantial technical issues and concerns related to energy conservation, computation capabilities, communication and managing the sensors [17]. Due to the fact that WSNs operate under rough and sometimes fierce environments in addition the energy and bandwidth restrictions levied upon the network, here we present some of the issues and challenges that would face any WSN design project [3] [4] [5] [17] [31]:

- **Scalability:** It's common that WSNs involve the use of a large number of sensor nodes, therefore it's imperative that any incorporated protocol or/and architecture has full capability to handle these large number of sensors.

- **Lifetime:** Since the nodes are powered via depletable power supply (batteries) and for the WSNs to be feasible to construct they have to endure long periods of operation. Thus, devising energy efficient protocols, WSN platforms, Sensor's Operating Systems and applications becomes a high priority issue for WSN design along with load balancing capability which also serves as an energy conserving technique.

- **Fault tolerance:** In WSNs nodes failure is inevitable due to many causes, such as running out of energy, defective parts or impairment as a result of harsh environment. Nonetheless, It's of the utmost importance that the network maintains an acceptable continues functionality, in spite of all the issues that cause the network to suffer of occasional failures.

- **Production Cost:** One of the driving forces behind the global interest in WSNs is the fact that they provide an incredibly cost-effective solutions than their peers of traditional networks that use traditional sensors. As a result, keeping the cost of constructing and utilizing a WSN as low as possible is a quintessential factor in regard to maintaining the demand for WSNs. Consequently, the production of low-cost sensor nodes becomes an obvious target. According to [3] a sensor node must not cost more than one dollar in order to keep the WSN economically appealing.

- **Environment:** A key aspect of WSNs is their autonomous operation and since they are frequently deployed in remote geographic areas, it becomes mandatory to consider the environment where the nodes will be located due to the fact that some environmental conditions may affect the network's operation, accuracy and/or maintainability.

- **Security:** Security concerns related to WSNs are of great importance, since the nature of WSNs involve nodes that utilize wireless communications among themselves and while communicating with the base station. A huge security vulnerability becomes evident when critical data is being transmitted between the sensor nodes and to the base station via wireless communication which could pose the threat of unauthorized reception by malicious or unintended parties. However, satisfying the security requirements for WSNs is not an easy task as it demands difficult and over the top algorithms and other issues which could take longer processing time in the nodes or the application in order to keep the confidentiality of the data. Examples of critical data is the military applications where encryption and preventing unauthorized exposure of the data is an extremely crucial feature of such networks.

- **Realtime:** In many real life applications of WSNs timing limitations are imposed on the network operations of sensing, processing and reporting data. In some applications of WSNs a realtime reporting of certain events poses a fundamental requirement for the WSN existence and failure to comply with this requirement renders the WSN obsolete.

2.1.2 Types of WSNs

According to [31] WSNs are being classified based on the deployment environment, since considering the environment where the WSN will be deployed will answer several questions in regard to the type of issues and difficulties the WSN will be facing and how to overcome them in order to allow the WSN to deliver its intended purpose of providing the required data and results for the users and their applications. The following list shows the 5 types of WSNs along with a brief description for each type according to [31]:

1. **Terrestrial WSNs:** It's the type of WSN where a large number of wireless sensor nodes are scattered in a specific area. The deployment method could be implemented in a random manner or it could follow a certain plan where specific locations are assigned for the deployed sensor nodes.

2. **Underground WSNs:** This type of WSNs is created by deploying the sensor nodes in the underground or by placing then inside a cave or a mine in order to collect data related to the underground conditions. An underground WSN network is usually costs more than a terrestrial WSN, since the type of equipment and the deployment methods along with the maintenance work are typically more expensive.

3. **Underwater WSNs:** In an underwater WSN the sensors and vehicles are deployed under water. A major difference between an underwater WSN and the conventional terrestrial WSN is the fact that in terrestrial WSNs the sensor nodes are deployed in huge numbers and in a the underwater WSNs a limited number of sensor nodes along with the exploration vehicles are deployed to monitor a certain phenomenon. In underwater WSNs the wireless communication between the nodes, the vehicles and the base station is achieved by using acoustic waves. By using the

acoustic waves as a wireless communication method certain issues arise due to the fact that with acoustic waves there will be bandwidth constraints, higher signal fading and more propagation delay. Sensor nodes failure underwater is another issue that must be addressed due to the difficult underwater conditions.

4. **Multi-media WSNs:** In a multi-media WSN the data collected from the monitored phenomenon are in the form of multimedia, such as video, audio and image. This type of WSNs is created by deploying a number of inexpensive sensor nodes where each sensor node has a camera and microphone to capture the required multimedia data. It's typical for the Multi-Media WSN sensor nodes to be deployed in an organized and planned manner in order to cover the designated area thoroughly. The gathered data in a Multi-Media WSN are usually quality demanding which burdens the network with the need for more bandwidth which in turn requires more energy to be consumed by the sensor nodes and it also increases the need for better data processing and compressing techniques along with improvements in the cross-layer design. An example of the high demands by the Multi-media WSNs data content is the video stream which usually needs higher bandwidth and in return a higher data rate and energy consumption.

5. **Mobile WSNs:** In Mobile WSNs the sensor nodes are capable of movement and changing their location in order to gather data of their environment. Unlike the stationary nodes the mobile nodes are capable or relocating and arranging themselves in the network in order to collect more information of their surroundings. A Mobile node is able to contact another mobile node in its range of wireless communication which allows for information sharing. Issues that face Mobile WSNs are related to deployment, localization, self-organization, navigation, coverage, energy consumption, maintenance and data processing.

2.1.3 WSN Applications

WSN applications are becoming increasingly diverse and being applied to new domains to introduce highly innovative solutions to old or emerging problems. The old fashioned methodology of introducing WSN as a solution was based on replacing the wired technologies with the wireless sensors [4]. The type of data gathered by the sensors had created the need for many types of sensors where each type is equipped to sense and monitor a certain phenomenon [3]. The following lists some examples of ambient conditions monitored by WSNs [3]:

- Temperature
- Pressure
- Noise levels
- Humidity
- Vehicular movement
- The existence of specific types of objects
- Soil makeup
- Lightning condition

As we mentioned earlier in this section that the potential for new applications for WSNs is significantly high with numerous prospects. As a result there is no one way of classifying WSN applications, however the following presents a corroborated list of the most classic application domains for WSNs [3] [17] [5] [4], figure 2.2 provides an overview of WSN applications:

- **Smart Homes:** The use of WSNs in creating smart homes had been the most important factor in their realization. Major energy conservation improvements were attained by reducing the amount of wasted energy due to the smart WSN applications that control the proper usage of the Humidity, Ventilation and Air Conditioning.

- **Environmental Monitoring:** One of the major uses of WSNs is to monitor environmental conditions which usually include the deployment of a large number of sensors in what is called a large-scale WSN. Environmental monitoring applications include weather conditions, water levels, agriculture and wildlife, chemical plants pollution levels and early warning systems for environmental catastrophes, such as forrest fires, earthquakes, tsunami, hurricanes and floods. WSNs has many advantages when used in environmental monitoring due to their nature of construction and operation, since they provide their services for a long-period of operation while being unattended and unobtrusive to their environment.

- **Military:** In military applications WSNs enjoy the fact that due to their cheap cost and disposability they make their destruction by the enemy has a insignificant impact to the military operations which serves the purpose of using more WSNs in military applications. There are many usages of WSNs in military applications which mainly revolves around intelligence, surveillance, reconnaissance, targeting and communications infrastructure.

- **Traffic Control:** Recently WSNs are being used more and more in traffic control and monitoring traffic jams or accidents. Improved versions of traffic control applications allows for more management operations where for example the control of a signpost based on the traffic condition sensed by the network.

- **Security:** WSNs are being employed in security applications in an increasing manner where they are replacing and/or collaborating with wired systems, since WSNs posses advantages over wired application because of their highly flexible deployment factor along with their adaptivity and fault tolerance. Examples of security applications of WSNs are: intrusion detection, borders-crossings monitoring and identity authentication systems.

- **Health Care:** Although the use of WSNs in health applications could be considered somehow invasive to patients privacy, it actually had many useful usages that could literally save lives. An example of a health care application of WSNs is the use of a wireless body sensors in order to monitor the vitals or certain medical related conditions in the patient, using the wireless sensors would allow the patient to gain more mobility compared to the wired sensors which restricts the patient movement. Also, the use of WSN networks in hospitals to locate doctors, nurses and medical equipments facilitates the sudden urgent need to form response teams for emergency situations.

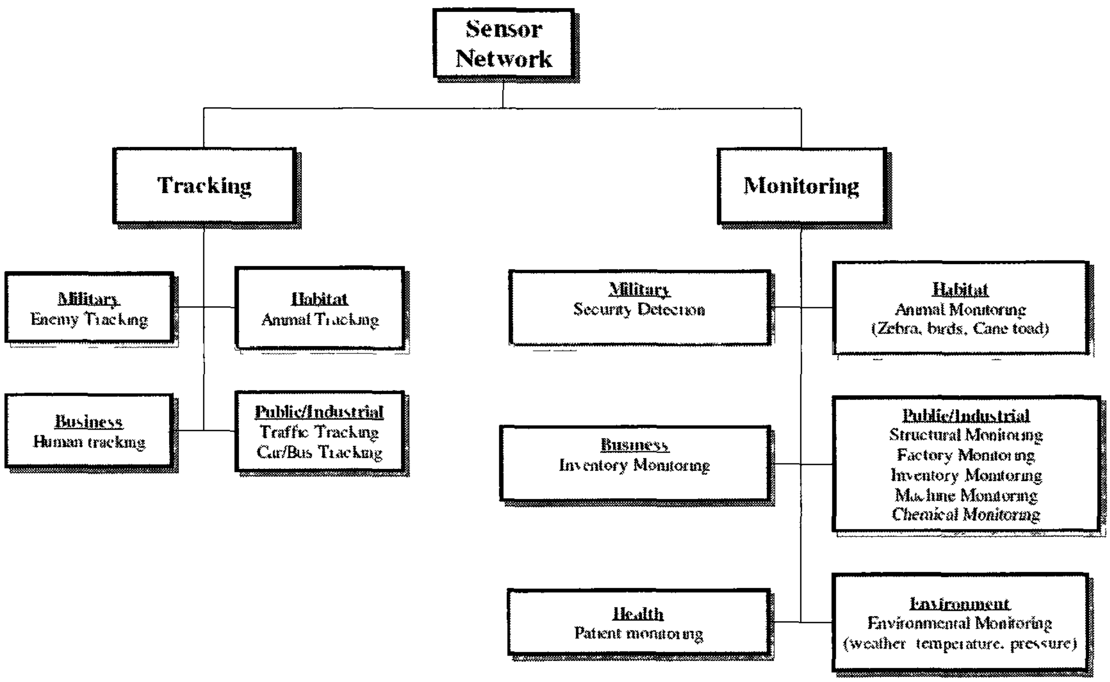

Figure 2.2: An overview of WSN applictions [31].

Regardless of the application type, there are common operations conducted by all WSN applications, such as collecting data and processing it, then reporting results.

According to [3] [5] [4] the following are the four major tasks independent from application type:

1. **Monitoring:** Reporting the value of a monitored parameter in a periodic manner of a certain location of the entire monitored region, such as temperature, pressure, humidity or water level.

2. **Event Detection:** In this task the network only reports if certain defined event has occurred or not along with its parameters.

3. **Object Classification:** In this type of tasks it becomes usually necessary for multiple sources of data to collaborate in collecting and processing the data in order to determine the identity of a certain object or an event.

4. **Object Tracking:** In this task the network traces the movement and location of a mobile object inside the monitored region of the network. This type of tasks is our main focus in this thesis.

2.1.4 Sensor Node Architecture

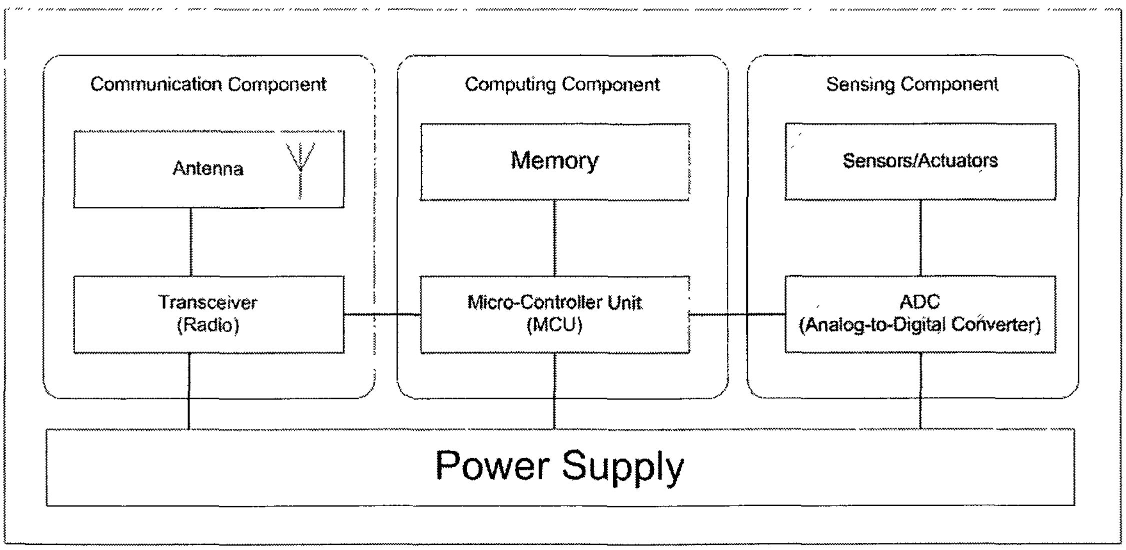

Figure 2.3: A general WSN node hardware architecture.

In the process of creating a WSN the application has the most influential factor in selecting a certain type of a sensor node based on its size, cost, capabilities and energy consumption. In designing WSNs the main question when choosing a sensor node becomes how to balance between the required features in the node and the cost of the node, since the more advanced the node the more expensive it will be. Figure 2.3 presents a general WSN node hardware architecture. Also, despite the fact that there are many types and models of sensor nodes, it's almost a fact that the following five main components must be available in any sensor node:

- **Communication component:** In order to allow the sensor node to be part of the network and communicate with other sensor nodes, sink nodes or base stations. The communication component has a transceiver that contains both a transmitter and receiver of RF radio communications which is attached to an antenna [3] [10].

- **Micro-Controller Unit (MCU):** The MCU in a sensor node is the main part of the computing components and acts like a CPU of a computer, since it handles the computation tasks of processing the gathered data from the sensing components and/or the communication device. Also, the MCU is responsible of controlling the functionality of the other components in the node and executing the sensor node's operating system/applications commands [11] [12].

- **Memory:** An external memory is commonly used in nowadays sensor nodes and it's part of the computing components. The type of the memory and its storage capacity is usually application dependent, although it's common to use flash memory or an on-chip memory of a MCU. There are two main types of sensor memory where (a) User memory which is used to store user data or application related data. (b) Program memory which usually contain data to program the sensor node and may contain identification data of the sensor node [33].

- **Sensing Component:** The sensing components could include sensors and/or actuators. Sensors are the parts of the node that has the ability to retrieve readings and measurements of the physical condition they are sensing, such as temperature, pressure and movement. It's also common to use an Analog-to-Digital Converter (ADC) in the sensing components in order to digitize the analog signals from the sensor which is later passed to the MCU for further processing. Actuators are mechanical devices that are used to perform pre-defined mechanical motion in order to carry out a certain task. An actuator could be apart of a sensor node or attached to a sensor node where the actuator performs its task based on the readings or commands from the sensor node, an example of an actuator are electrical motors, hydraulic pistons and comb drives. [5] [33]

- **Power supply:** This part of the sensor node provides energy to power the other components of the sensor node. The main power source for sensor nodes are usually batteries which could be chargeable or non-chargeable [33].

2.2 Object Tracking Sensor Network

Object tracking which also called target tracking is a major field of research in Wireless Sensor Networks and has many real life applications such as wild life monitoring, security applications for buildings and compounds to prevent intrusion or trespassing, international borders monitoring for illegal crossings. Furthermore, object tracking is considered one of the most demanding applications in Wireless Sensor Networks due to its application requirements which places a heavy burden on the network resources, especially energy consumption. The main task of an Object Tracking Sensor Network (OTSN) is to track a moving object and to report its latest location in the monitored area to the application in an acceptable timely manner, and this dynamic process of sensing and reporting keeps the network's resources under heavy pressure. Therefore, there has been many research efforts to optimize the process of object tracking in sensor networks and a great number of research issues has been identified which are related to the design and implementation of OTSNs, and examples of those research issues are: data aggregation, localization, coverage, data fusion, routing and energy conservation. Energy conservation in OTSNs is regarded as the most crucial research issue, since OTSNs are operated with a limited power supply like other types of WSNs. [1]

2.2.1 Background

In an OTSN application, if a moving object enters the area called the *monitored region* which has the sensor nodes of the network deployed and operating, the sensor nodes that are able to detect the object will continue monitoring the object while it's under their detection area. At the same time, all the other sensor nodes in the network will maintain the status of being inactive in order to save their energy. Furthermore, while the moving object is continuously changing its location, it becomes critical for the application to create a group of sensor nodes that are switched to active mode in order to keep monitoring the object when it arrives to their respective locations [30]. The *monitored region* geographical limits and measurements are known to the OTSN application in addition to vital information about the tracked object, such as location, velocity, direction, size and shape [1]. The medium of communication between the application and the OTSN is similar to other WSNs, where a base station or a gateway is used to provide an interface channel that collects the data from the network and forwards them to the application, while also distributing the application's commands to the sensor nodes. Also, since OTSNs are notoriously famed for their high energy consumption due to the nature of their tasks, therefore their sensor nodes similarly to other WSNs are able to shut down their basic components (sensor, MCU, RF radio) in an independent manner where each component could be turned on/off without affecting the other components status for the sake of preserving the node's limited energy. [1]

As identified earlier that the main goal of an OTSN network is to monitor and track any object that enters the monitored region where the nodes must report the objects location to the application along with any additional information in a predetermined time-dependent updates. [1]

Each sensor node in the network is required to monitor its *detection area* in anticipation of an intrusion by a moving object. Thus, when the sensor nodes start sensing their *detection area* so as to retrieve the object's attributes, this sensing task lasts for a specified period of time called the *sampling duration.* Furthermore, while the sensor node is going through the sampling duration its MCU and sensing components are turned on in order to obtain the sensed data and then process it. However, the communication component in the sensor node (RF radio) is most likely kept inactive when there are no communication requirements. The task of sensing the *detection area* is usually called sampling and it takes place after the passing of a determined time period resulting in a specific *sampling frequency* which is influenced by they network and the application requirements. Moreover, the *reporting frequency* is also controlled by the application requirements where the sensor nodes report to the application the presence of an object in their *detection area* every specified period of time. Alternatively, the communication component (RF radio) is kept in inactive mode if the sensor node has nothing to report. The application in an OTSN network could control the degree of data accuracy in regard to the objects' movement by adjusting the *reporting frequency.* [1]

2.2.2 OTSN Influencing Issues

There are many elements that affect the behavior and operation of an OTSN, many of those are related to the fact that OTSN is a WSN application where all communications are wireless and the sensor nodes are distributed over a certain geographical area where they conduct their operations autonomously. Furthermore, OTSNs has many distinctive influencing issues which requires thorough examination before the design and construction of an OTSN. According to [1] the following are considered the most critical issues related to OTSNs which have direct consequences on energy consumption:

- **Number of moving objects:** Obviously an OTSN is not usually made to monitor one object at time, hence addressing the number of objects tracked by the network is directly related to the network's energy consumption, seeing that any increase in this number means an increase in the sampling and reporting operations which ultimately impacts the energy consumption of the network and therefore can be considered as the network's workload.

- **Reporting frequency:** it represents the number of reports received by the application during a certain period of time. As a result, the reporting frequency is an application-dependent element, where increasing the number of reports per specified period of time would eventually give the application a better view in regard to the object location in the *monitored region.* On the other hand, lowering the number of reports received by the application would decrease the amount of communication carried out by the sensor nodes to the base station or the gateway which conserve the network's energy.

- **Sampling frequency:** It's considered a crucial and significant element that has a great impact on OTSNs. Since if the sensor sampling frequency was lower than what it should have been, then the application will be running the risk of losing track of the moving object. Nevertheless, maintaining a high sampling frequency in addition increasing the application's data accuracy it also drives the network to consume more energy.

- **Location models:** creating accurate location models affects on the retrieval of the most vital information received from the senor nodes, which is the location information. Depending on the variations of locations of objects along with measuring the passing time, it allows for the calculation of other information related to the moving object such as speed and direction. There are two types of location models: *geometric model* and *symbolic model* where the application requirements decides which location model will be employed in the design and implementation of the OTSN [1].

- **Data precision:** this issue is directly affected by the aforementioned elements: sampling frequency, sampling duration and location models. Consequently, for the sake of attaining a high level of data precision, then it's imperative to increase the data retrieval, processing and reporting which incurs consuming more energy on the main sensor nodes operations: sensing, computing and communicating.

- **Object's movement speed:** this element is addressed along with the sensor sampling frequency since to maintain a successful tracking operation of an object the OTSN will be required to conduct more sampling in case the monitored objects move in high speed, which results in higher energy consumption.

2.2.3 Categorization of Object Tracking Methods

By considering the various research projects in the object tracking field, it's yet to be clearly defined in the literature with an appropriate categorization of the many algorithms and protocols [30]. However, according to [30] the most comprehensive classification of those approaches would generate the following five categories:

1. **Tree-based tracking**

2. **Cluster-based tracking**

3. **Prediction-based tracking**

4. **Mobicast message-based tracking**

5. **Hybrid tracking schemes**

2.2.4 Tree-based tracking

In this approach of object tracking the network follows a hierarchical organization that could be described as a hierarchical tree. In this graph the sensor nodes in the network are shown as vertices where the edges that connect those vertices denote the communication links between the sensor nodes which are able to establish a direct line of communication with each other. [30] [36]

There has been a number of algorithms and protocols which implemented the tree-based approach as their framework, examples include the Dynamic Object Tracking (DOT) protocol [51], Deviation Avoidance Tree (DAT) technique [53], Heuristic Object Tracking algorithm [45] and Scalable Tracking Using Networked sensors (STUN) [52].

The Dynamic Object Tracking (DOT) protocol [51] serves the purpose of tracking a moving object and guiding a moving source to its location in order for the source to capture the object. In this protocol each sensor node determines its neighbors by using the spatial neighborhood represented in a planar graph when they first convert the network into a planar graph and use Gabriel Graph (GG) and the Relative Neighborhood Graph (RNG) to discover the face neighbors for each node in the network. When the source is trying to locate the object in the target discovery step it begins sending requests to the sensor nodes and once the target is discovered by a node it replies back to the source. Also, to maintain a continuous monitoring on the moving object, the sensor node which is detecting the object starts waking up its spatial neighbors to keep track of the object in case it leaves the ingress node detection area. In the target tracking step the beacon node which is the node that has the moving object location information answers the queries issued by the source in regard to the location of the object which eventually guides the source to the location of the object. [30] [51]

In STUN [52] the assumption is that the targets to be tracked have a predictable movement in terms of their speed which is limited and therefore by determining the period of time during which the target was moving then the maximum distance calculation becomes possible based on the fact that this target has a maximum speed. In this method for every two nodes the link is calculated by Euclidean distance and is allocated a cost. Since this method is basically a hierarchy then the leaf nodes have the task of detecting the moving objects and then informing the sink by transmitting their data through the intermediate nods. Message-pruning is a concept applied in STUN where the intermediate nodes ignore redundant messages from the leafs, see figure 2.4. A record of information about the monitored objects is maintained by the intermediate nodes and if there are any changes in regard to the tracked objects information the record will be updated and the new information is also sent to the sink. There two main weaknesses in regard to STUN. First, its logical tree DAB (Drain and Balance) could cause dramatic increases in communication costs, since it doesn't reflect the physical layout of the deployed network, and what seems as a single link between two nodes in the DAB could be actually an edge that made up of multiple nodes and as result what should have been a single hop turns out to be a multi-hop communication stream, hence increasing the energy consumption in the network. The second weakness is related to the query costs in DAB, since DAB is constructed without taking into account query optimization which drives the cost of queries in some cases more than the acceptable levels. [30] [36] [52]

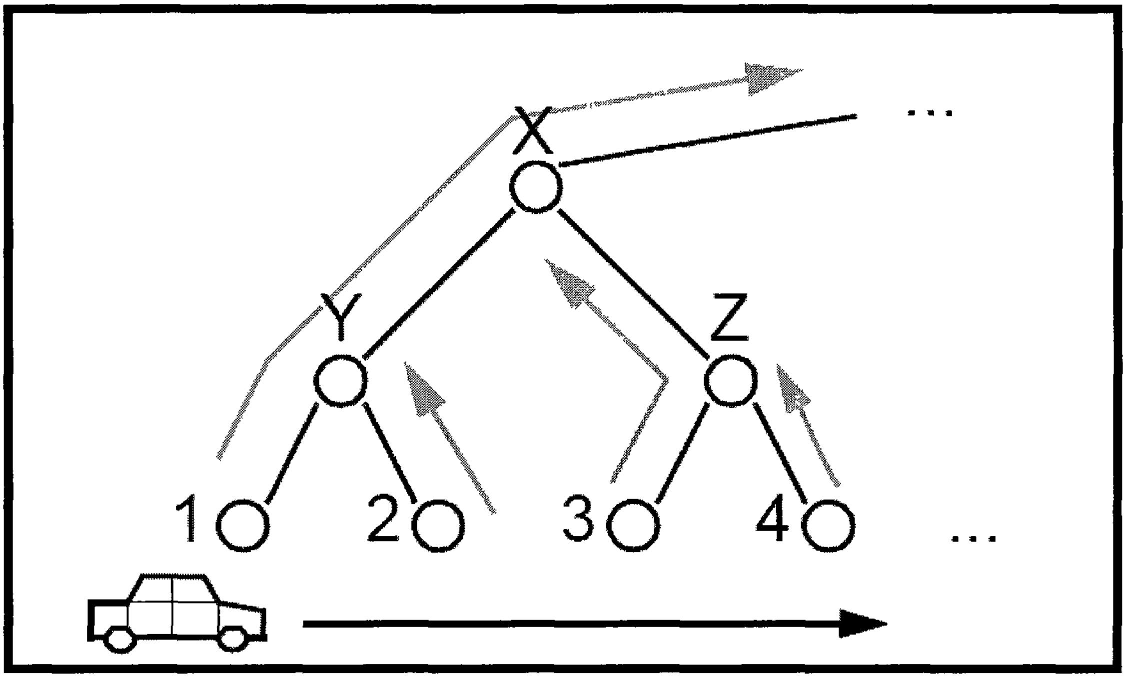

Figure 2.4: An example of message-pruning in STUN. [52]

The Deviation Avoidance Tree (DAT) technique [53] was devised as an improvement to the STUN approach [52] where they have arranged the sensor nodes to form a logical tree which expedites the in-network data processing and also contributes in the overall cutback of the communication cost sustained by tracking the moving object. Moreover, the physical layout of the network is utilized in the section of location update and also using the Deviation Avoidance Tree (DAT) algorithm to result in a significant reduction in the cost of location update. For the second part of the technique the query cost is considered by using the established query rates of the sensor nodes and the Query Cost Reduction (QCR) algorithm to decrease the query cost. [30] [53]

In [45] a novel technique is introduced which basically uses heuristics in object tracking where the problem is constructed as 0/1 integer programming problem. In addition, this technique introduces a Lagrangean Relaxation-based (LR-based) heuristic algorithm to provide solutions for the optimization problem. The LR-based algorithm plays a major role in making the object tracking more efficient by reducing the cost of the total communication of the object tracking tree, however the algorithm suffers from scalability issues. Also, to enable the measurement of tracking accuracy the algorithm requires some changes and amendments. [30] [45]

Another example of the tree-based tracking approaches is the Optimized Communication Organization (OCO) [62]. In this approach the object tracking is achieved with minimal computation overhead on the sensor nodes while offering itself with autonomous organization and routing abilities. In OCO the object tracking process is divided into four stages: Position collection, Processing, Tracking and Maintenance. The position collection stage is basically performed since the network is deployed in a random manner when it's initially constructed and the location of each sensor node in the network is not identified by the base. Therefore, in this stage the neighboring nodes of the base will begin receiving messages from the base which informs them of the base's ID and requests the neighbors IDs. The process is repeated throughout the whole network till all the nodes IDs and location are known by the base node. In the processing stage three actions are performed (1) Removing the redundant nodes which are the nodes that has a monitoring area being covered by one or more nodes, (2) Identifying the border nodes by using the border detection algorithm, (3) The shortest path from each node to the base is determined by applying an algorithm to find the shortest path. In the tracking stage and depending on the type of the sensor nodes (Type A or Type B) the moving objects are identified and a tracking process is activated to monitor the object during its existence inside the network coverage area. The maintenance stage is a crucial phase of the technique since it makes the network topology adaptable to necessary changes in order to continually track moving objects inside the network, especially when there is a node failure or changes to the network topology due to nodes repositioning. A major weakness in this approach is caused by keeping the border nodes active all the time which eventually leads to the node consuming all its energy faster than the inner nodes in the network. [30] [62]

2.2.5 Cluster-based tracking

In object tracking the use of cluster-based approaches has been utilized in many projects and it had a significant impact on the design and implementation of the object tracking systems, since clusters are created to provide more collaboration in the data gathering and processing mechanisms. A cluster of sensor nodes contains a sensor node chosen as the cluster head (CH) and other sensor nodes represent the members of the cluster. There are two methods of creating clusters of nodes in the network: static clustering and dynamic clustering. In static clustering the creation of the clusters are accomplished during the early stages of the network deployment along with all the attributes of each cluster, for example, the list of nodes included in the cluster, the monitored area and the designated Cluster Head (CH). Although static clustering offers more direct approach in creating clusters and lacks the complexities related to the dynamic creation of clusters, it has many shortcomings mainly related to its inflexibility, especially when fault tolerance is addressed, since if a cluster had many node failures it will not be able to perform its tasks properly and it can not include other sensors from different clusters to participate in the monitoring process. Furthermore, the sharing of information between sensors in different clusters is not allowed due to the static sensors' cluster membership. On the other hand, the use of dynamic clustering grants many benefits directly related to its flexibility which allows the network to establish new clusters of nodes as the need arises. Hence, if a certain sensor node starts sensing movement in its coverage area then it could establish its own cluster and invites other sensor nodes to become members of the newly formed cluster in order to monitor the moving object continuously and this process could be restricted to sensor nodes that possesses enough power supply and adequate computing ability. Also, in dynamic clustering it's imperative to keep only one cluster head (CH) actively tracking objects and this is usually achieved by the use of decentralization methods. Due to the nature of the dynamic clustering each sensor node in the network might become a member of different clusters during the network's operation as long as it doesn't become a member of more than one cluster at any given time to avoid redundancy and inaccuracy in the gather data. In addition, keeping only one cluster actively monitoring an object decreases the collection of redundant data and contains potential interference. [30] [36] [54]

Some examples of cluster-based approaches include the following: DELTA [46], RARE [56] and CODA [59].

DELTA [46] is a distributed event localization and tracking algorithm which uses light measurements as the bases of its tracking approach. DELTA was designed to utilize the known positions of the sensor nodes in order to track a person moving at a constant speed who's holding a flashlight. The algorithm creates dynamically clusters of sensor nodes and selects as the cluster head (CH) the sensor node which has the best location to track the object as well as communicate with the cluster members. The mechanism used in DELTA to chose the leader sensor node of a cluster is described as a fast and highly accurate which manages in a greatly flexible manner to select best candidate. The tracking precoces in DELTA is mainly carried out by the leader of the cluster which cooperates with other sensor nodes in the cluster to achieve constant monitoring of the moving object. The main disadvantage of DELTA is that it was designed to monitor objects which moves in a constant speed. [30] [46]

When RARE [56] was developed it had a goal of achieving target tracking while being energy efficient. RARE was designed with two algorithms RARE-Area (Reduces Area REporting) and RARE-Node (Reduction of Active node REdundancy) and both algorithms use static clustering. The goal of RARE-Area is to ensure that only nodes that have a good coverage of the moving target will be involved in tracking process, and thus preventing any node that couldn't offer a good contribution to the tracking of the moving object based on its location from the object. So the further the node is from the moving object the lower its chance to be allowed to participate in the tracking process. As for RARE-Node main goal is to reduce the redundancy of the collected data from overlapping sensor nodes by first determining the nodes with redundant data due to the fact that they share their coverage area with other nodes and then prevent those nodes with redundant data from participating in the tracking process. [56]

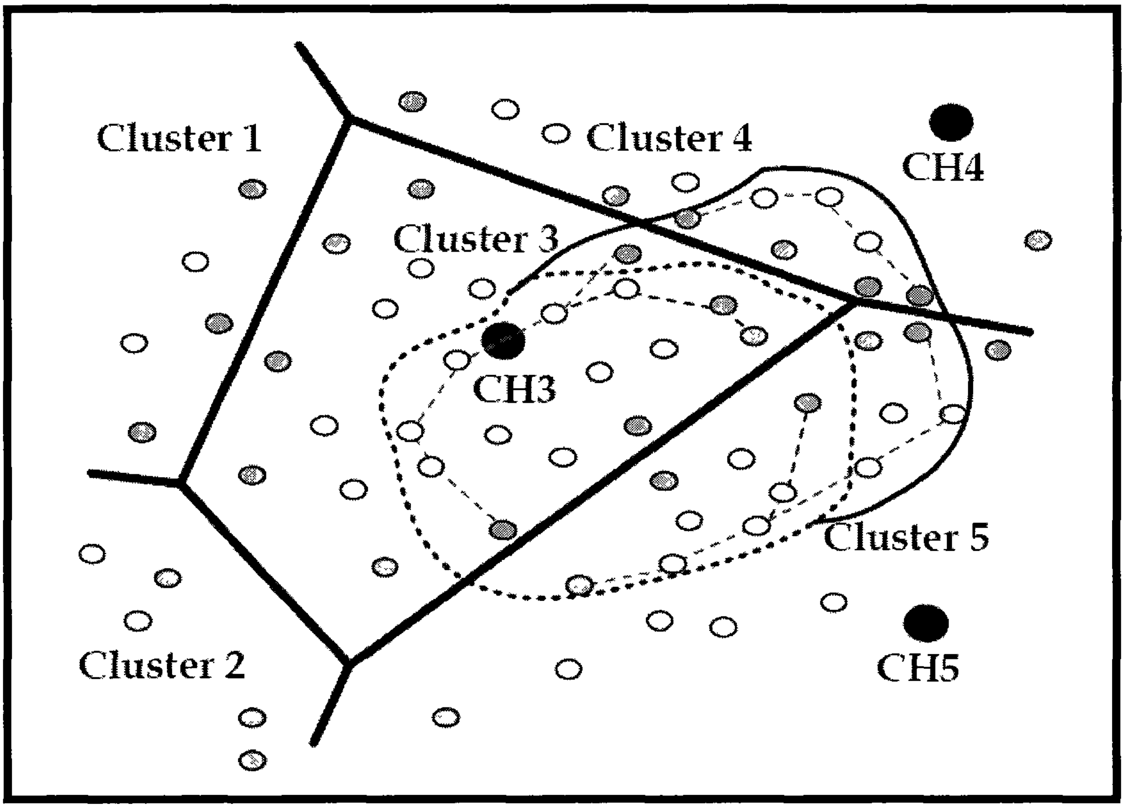

Figure 2.5: Boundary sensors being updated while tracking the moving object in CODA. [59]

CODA [59] (Continuous Object Detection and Tracking Algorithm) represents a good example of a cluster-based approach which utilizes both static and dynamic clustering in its algorithms which is considered a hybrid clustering approach. The key issue with CODA is that allows the sensor nodes to form dynamic clusters inside the static clusters where those dynamic clusters include the nodes that form the boundaries around the moving object and these boundary clusters are changed when the object starts to move away of its former boundary sensors coverage area, see figure 2.5. The methods applied in CODA are reducing the communication overheads resulted by the estimation process of the moving object boundary, since the main weight of performing these calculation is assigned to the static cluster head (CH) which later on communicates these information to the network sink. The main weaknesses in CODA is the fact that it doesn't provide any process to achieve recovery when there is a missing track and it doesn't give any data regarding its tracking accuracy. [30] [59]

2.2.6 Prediction-based tracking

The majority, if not all, of prediction-based techniques are built on top of a tree-based and/or cluster-based approaches which are modified to include a prediction mechanism usually called the prediction model. Furthermore, prediction models are typically designed in a way that employs heuristics which follow certain assumptions, for example:

- Objects tracked by the network are going to proceed with their movement while maintaining a certain speed and direction every few seconds.
- Based on the object's movement history, it will be possible to determine its future speed and direction for the next few seconds.
- For each stage in the recorded history a certain weight can be assigned to determine its value for prediction purposes.

The main objective of utilizing these prediction techniques is to minimize the number of sensor nodes participating in the tracking of a moving object which consequently results in significant energy savings. In spite of the encouraging results achieved by the prediction-based techniques, these methods still suffer from the same drawbacks as their base approach (tree or cluster). [36]

An example of a prediction-based approach is presented by S.-M. Lee et al [8] where two prediction algorithms are designed to provide error avoidance and error correction. By using the error avoidance technique the algorithm will attempt to predict the location error of the moving object since determining the future wake-up zone is possible by calculating the object's trajectory and monitoring the object while it's still moving inside the detection area of the current wake-up zone. Moreover, if the predicted future wake-up zone times-out before detecting the object it will inform the current wake-up zone which launches the error correction method to enlarge the wake-up zone based on the last known object movement direction in order to detect the object when it changes its direction. This technique has proven that it handles very well the changes in direction in the object movement, however, it doesn't address changes in the speed during the object movement in the network. [30] [8]

Reducing the energy consumed during object tracking was the main goal of both the dual prediction [7] and the prediction-based strategies for energy savings [1]. In both techniques, energy savings were achieved by limiting the tracking process of a moving object to the fewest number of nodes while maintaining the rest of the nodes in the network operating in sleep mode, thus saving their energy until the need arises and they are activated by an active node to participate in the monitoring and tracking process. The prediction mechanism in both approaches depends on the use of the recorded historical data of the object movements which are used to calculate the future position of the moving object. The calculations of the future location of the moving object are done in both the sensor node and the sink. Also, to reduce the amount of communication taking place between the sensor nodes and the sink, the updates are not sent to the sink while the object movements in harmony with the predictions. [1] [7] [30]

2.2.7 Mobicast message-based tracking

Mobicast message-based techniques rely on prediction to track moving objects. Mobicast (Mobile Just-in-time Multicasting) schemes are mainly spatiotemporal multicast methods which provide a spatiotemporal coordination for applications such as object tracking in sensor networks. The main concept of a mobicast message-based approach is the spatiotemporal multicast method that delivers a message to a group of sensor nodes called the *delivery zone* which its membership changes over time based on the predictions of the movements of the tracked object. The nodes in the delivery zone receive timely messages from the Mobicast protocol to activate before the arrival of a moving object and the delivery zone movement speed is relatively as the same speed of the moving object. Moreover, any sensor node in the network gains membership of the delivery zone by receiving a mobicast message [30] [44].

The Variant-Egg-based Mobicast Routing protocol (VE-mobicast) [49] utilizes Cassini oval equation to allow the nodes in the "hold-and-forward" state to evaluate the initial shape and size of the forwarding zone while adjustments are being applied in a dynamic manner by a distributed algorithm using a simple control packet. In the HVE-mobicast [50] technique a major power efficiency enhancement was achieved by allowing message delivery to take place from cluster to cluster in comparison to the old technique where message delivery is possible between node to node. In addition, HVE-mobicast addresses the issue of the delivery zone changing speed. [30] [49] [50]

Another mobicast protocol was presented by Huang et al. [44] where the delivery zone changes its size dynamically depending on the local compactness properties of a node's neighborhood. This approach exploits the tradeoff between delivery guarantees and communication overhead and it reduces communication to the forwarding zone which could affect the delivery guarantee since it gets also decreased as a result. [30] [44]

2.2.8 Hybrid tracking schemes

Hybrid tracking schemes are referred to the type of tracking methods, protocols and algorithms which are based on more than one tracking approach, such as when a cluster-based approach is also tree-based then this approach would be called a hybrid tracking technique. There are many examples of hybrid protocols since a great deal of the designed tracking schemes usually involve more than one tracking approach and these examples include M. Abdelkader [47], DPT [2], W.-P. Chen et al. [54], Xu et al. [9], PES [1], Huang et al. [44], Jin et al. [58], Pino et al. [60], C.-Y. Chong et al. [61], HPS [57], VE-mobicast [49], HVE-mobicast [50]. HVE-mobicast is mainly a mobicast message-based tracking approach, yet the network design involves the creation of clusters and it utilizes prediction techniques to provide approximate estimations of the shape and size of both the forwarding zone and the delivery zone in order to convey the mobicast message (multicast message) to the nodes in the dynamically formed clusters of both zones. [30]

2.2.9 OTSN Categorization Summary

The following table 2.1 provided by S. Bhatti et al. [30] shows a comprehensive overview of a number of tracking protocols and their respected category. Also, identifying hybrid schemes becomes obvious based on the fact that for each protocol that is classified under more than one category then it's certainly a hybrid scheme. In addition, the table demonstrates other facts about the listed protocols which include the existence of a data aggregation mechanism, query processing and/or the utilization of scheduling strategies. It's also worth mentioning that the use of data aggregation mechanisms and scheduling techniques assists in the reduction of the energy consumed by the network during the monitoring and tracking process of a moving object [30]. We have also included our proposed prediction technique to the classification table to show how our technique fits between other tracking schemes and protocols and it can be clearly noticed that our proposed technique has similar properties as the PES [1] since we have adopted their framework in our scheme.

Table 2.1: **Classification of Target Tracking Approaches [30]. The last technique (this thesis) was added to the table for comparison**

Tracking Protocols	Tree-based	Cluster-based	Prediction-based	Mobicast message-based	Scheduling	Query	Aggregation
Heuristic Object Tracking Algorithm [45]	✓						
DOT [51]	✓				✓	✓	
STUN [52]	✓						
DAT [53]	✓					✓	✓
OCO [62]	✓						
M Abdelkader et al [47]	✓		✓		✓		
F Zhao et al [63]		✓				✓	
DELTA [46]		✓			✓		
RARE [56]		✓					
DPT [2]		✓	✓		✓		
W-P Chen et al [54]	✓	✓					
Y Zou et al [55]		✓					
S-M Lee et al [8]			✓				
Y Xu et al [9]		✓	✓		✓		
PES [1]		✓	✓		✓		
DPR [7]			✓				
VE-mobicast [49]			✓	✓			
HVE-mobicast [50]		✓	✓	✓			
Q Huang et al [44]			✓	✓			
G-Y Jin et al [58]		✓	✓		✓		
HPS [57]	✓	✓	✓				
CODA [59]		✓					✓
S Pino et al [60]		✓	✓		✓		✓
C-Y Chong et al [61]		✓	✓				
Proposed Prediction Technique (This thesis)		✓	✓		✓		

2.3 Association Rules and Sequential Patterns

In this section we will present the reader with a brief introduction to two data mining techniques, namely Association Rules and Sequential Patterns, which are utilized as methods of unveiling hidden relationships between inherently unrelated data items and since we use them in our predication-based tracking technique to create our prediction model as explained in Chapter 3 [20] [34] [71].

2.3.1 Association Rules

One of the famous data mining techniques is association rules mining where it opens the door to discover a great deal of interesting associations and/or correlation relationships between large sets of data items and variables in large databases [34]. It all began when Piatetsky-Shapiro [69] introduced the analysis and discovery of strong rules in databases by applying various levels of interestingness. Later on and building on the concept of strong rules, Agrawal et al. [68] presented association rules which are able to uncover hidden relations between products in large databases and used a supermarket database as a case study for their techniques and algorithms. To illustrate the idea of association rules consider the following example. If for example the recorded sales transactions at a supermarket indicates that a significant number of the sales transactions where a customer buys Potato chips and Popcorn together, he/she is likely to also buy Coke. Then the rule will be represented as $\{Potatochips, Popcorn\} \Longrightarrow Coke$ which can be used to further enhance the decisions made about how to market these items and other items as well and whether to place them in promotional packaging and how to figure the best product placement strategy for the supermarket items which could result in increasing sales and eventually profits. Furthermore, the concepts applied in the supermarket transactions analysis and their association rules can also be applied to various application fields, for example, Bioinformatics, intrusion detection and Web usage mining [34] [71].

Association Rules Definition

To introduce a definition of the association rule mining process we will be adhering to the definition provided by Agrawal et al. [68] where it states: Let $I = \{i_1, i_2, i_3, ..., i_n\}$ be a set of n binary attributes called *items*. Let $T = \{t_1, t_2, t_3, ..., t_m\}$ be a set of transactions called *database of transactions*. In the database of transactions T each transaction will be assigned a unique transaction ID and inside each unique transaction it will include a subset of the items in I. In addition, to provide a visual representation of each rule in the application it can be shown as an implication of the form: $X \implies Y$ where $X, Y \subseteq I$ and $X \cap Y = \phi$. Also, some naming conventions are presented by Agrawal et al. [68], for example, the set of items (*itemset* as a short alias) X which appears in the left-hand-side (LHS) is called the *antecedent* and the other itemset which appears in the right-hand-side (RHS) is called the *consequent*.

In order to shine more light on how the concepts of association rules work, we will introduce an example which represents a very small snippet of a supermarket transactional database. In the table 2.2 we show a list of transactions where each transaction has its own transaction ID and some items from the set of items $I = \{milk, cereal, cheese, coke\}$ and to represent the existence of a certain item in the transaction it will be shown as 1, and 0 will represent the nonexistence of the item in the transaction.

Table 2.2: Example of Supermarket Transactional Database with 4 items and 5 transactions. [34]

Transaction ID	Milk	Cereal	Cheese	Coke
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Therefore, to provide an example of a rule in the supermarket database we could say: $\{milk, cereal\} \implies \{cheese\}$ which can be interpreted as if a customer buys milk and cereal then he/she will also buy cheese. It's also worth mentioning that, although this example serves the purpose of illustrating how association rules work in real life applications, the number of transactions provided is awfully small, since in real life applications in order for a certain rule to be considered it requires the support of hundreds of transactions which would make it eventually of statistical importance and also the databases usually include thousands or even millions of database transactions [68] [71] [34].

Rules Selection Constraints

The importance of the rules generated in an application can only be measured by satisfying certain constraints which could identify the interestingness and significance of a certain rule among the set of all possible rules. The most important and famous constraints are minimum thresholds on support and confidence [68] [71].

- The *support* of an item or set of items (can be represented by variable s or supp(X)) can be defined as the ratio of transactions where the item or the set of items (can be called *itemset*) X exist to the total number of transactions. Therefore, based on the previous example, the itemset $X = \{Milk, Cereal\}$ which represents the union of two items (Milk and Cereal) as one itemset had occurred in 2 transactions among the 5 transactions in the database, which indicates that the support of the itemset X will be as follows: $supp(X) = 2/5 = 0.4$, so we can conclude that the itemset X had occurred in 40% of all transactions in the database. Furthermore, the itemset $\{Cheese, Coke\}$ has no support, due to the fact that those two items had never appeared in the same transaction in the database. Therefore, the support of the association rule $X \implies Y$ can be formally defined as the percentage of transactions in the database which contain $X \cup Y$. In addition, another way to represent the *support* of an item or sets of items (itemsets) X is by using the probabilistic notation as follows:

$$P(X) = \frac{\text{number of transactions where X occurs}}{\text{total number of transactions}}$$

Hence, $P(\{Milk, Cereal\}) = 40\%$. [20] [34] [68] [71]

- The *confidence* of a rule (denoted by *conf* or by variable α) can be viewed as the strength of the rule, since it's value represents the percentage of the number of transactions which contain the union of the itemsets X and Y $(X \cup Y)$ to the number of transactions which contain X [20]. In other words, confidence will be the result of dividing the *supp*$(X \cup Y)$ by *supp*(X), which can be shown as follows: $conf(X \implies Y) = supp(X \cup Y)/supp(X)$. In our earlier example, we can generate the confidence of the rule $\{Milk, Cereal\} \implies Cheese$ by calculating the support of the itemset $\{Milk, Cereal, Cheese\}$ which represents *supp*$(X \cup Y)$ and equals 0.2, since there is only one occurrence of the transaction that contains Milk, Cereal and Cheese among the 5 transactions in the database. Then calculating the support of the itemset $\{Milk, Cereal\}$ which represents *supp*(X) and equals 0.4 as shown previously. Finally, dividing $supp(X \cup Y)/supp(X)$ which is $0.2/0.4 = 0.5$ gives us the confidence of the rule $X \implies Y$ or as in our example $conf = (\{Milk, Cereal\} \implies Cheese) = 0.5$, therefore we can notice that for 50% of the transactions that contain Milk and Cereal the rule is correct. Furthermore, another approach to understand confidence by considering it as an estimate of the probability $P(Y|X)$, where we calculate the probability of the LHS of the rule in the transactions in the database while clinching to the condition that these transactions will also include the LHS of the rule. Furthermore, by using the probabilistic notation we can represent *confidence* of the rule $X \implies Y$ as follows:

$$P(X|Y) = \frac{P(X \cup Y)}{P(X)}$$

Therefore, the confidence of the rule $\{Milk, Cereal\} \implies Cheese$ can be expressed as

$P(\{Milk, Cereal\}|Cheese) = \dfrac{P(\{Milk, Cereal, Cheese\})}{P(\{Milk, Cereal\})} = 0.2/0.4 = 50\%$. [20] [34] [68] [71]

2.3.2 Sequential Patterns

Another form of data mining is called sequential patterns mining. In sequential patterns mining the data are represented as sequences where these sequences contain sets of items [18]. Sequential patterns can also be viewed as a special case of association rules, since in association rules the main concern is the relation of the items in one transaction, for example what are the items which were bought together in one transaction, so in a way we can claim that association rules represent the intra-transaction patterns, which differs than sequential patterns where it deals with the inter-transaction analysis of the data sequences [18] [19] [73]. This new data mining technique was introduced by Agrawal et al. [19] where they introduced the problem of discovering sequential patterns and viewed their input data as a set of sequences, called *data sequences*. These data sequences were basically lists of *transactions* from the database and in each transaction it contains a set of items or alternatively called itemsets. The main factor that contributes to the discovering of the sequential patterns is the fact that for each transaction it's associated with time-stamp which represents the time of the transaction's occurrence and it's order among the other transactions [73]. Also, any sequential pattern contains a list of sets of items [73]. The goal of this data mining approach is to identify all the sequential patterns which meet a predefined minimum *support*, where similar to the support in association rules, the support of a sequential pattern represents the ratio of the data sequences that contain the pattern [73]. We can consider a certain sequential pattern as a frequent pattern if it satisfies the user-defined minimum support threshold and as a result it can be of statistical significance in the database [18]. There are many applications where mining sequential patterns is of great use and benefits due to the fact that many applications are storing their data in a time-related manner which can be a fertile ground for generating useful sequential patterns that can aid in the understanding of the customer buying behavior and how to predict the occurrence of certain events based on earlier patterns found in the data [18].

Sequential Patterns Definition

In this formal definition of sequential patterns we will be following the original definition provided by Agrawal et al. [19]. In a supermarket database the transactions of the customers are stored in a database we will call it D. In the database D the transactions are created and stored with several fields which include a Customer ID, Transaction Time-stamp which designates the time when the transaction took place in the database which coincide with the time when the customer purchased the items stored in the transaction as a set of items. In the database D the primary key of the transactions table is the composite key of both the customer ID and the transaction Time-stamp, therefore, it's impossible to have two transactions for the same customer in the same Transaction Time-stamp [19]. Moreover, the generation of the sequential patterns is not concerned with the quantities of the items listed in the transactions, where the mere existence, and conversely the nonexistence, of the item is the only issue that matters in generating the sequential patterns [19].

Similar to the definition in the association rules the *itemset* is a set of items which can not be empty, whereas the *sequence* is basically an ordered list of itemsets. For the sake of clarity we will humor the assumption that the set of items is mapped to a set of adjacent integers. An itemset i is represented by $(i_1 i_2 i_3 ... i_m)$, and where i_j is an item. Also the sequence s is represented by $[s_1 s_2 s_3 ... s_n]$, where s_j is an itemset. Furthermore, the sequence a, indicated by $[a_1 a_2 a_3 ... a_n]$, is considered included in the sequence b, denoted by $[b_1 b_2 b_3 ... b_m]$, if there exist integers $i_1 < i_2 < i_3 < ... < i_n$ where $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, a_3 \subseteq b_{i_3}, ..., a_n \subseteq b_{i_n}$. [19]

Therefore, based on the definition provided earlier the sequence [(2)(34)(7)] is considered contained in the sequence [(6)(27)(8)(346)(7)], since the itemsets in the first sequence are contained in itemsets of the second sequence with respect to the order of occurrence, so $(2) \subseteq (27)$, $(34) \subseteq (346)$ and $(7) \subseteq (7)$. On the other hand, when considering the sequence [(7)(9)] it's clearly not included in the sequence [(79)], and also the sequence [(79)] is not contained in the sequence [(7)(9)], since in the sequence [(7)(9)] the items were recorded in two transactions respectively, or in other words the two items were bought one after another, whereas in the sequence [(79)] the two items were recorded in one transaction or alternatively saying they were both bought at the same time-stamp. Also a sequence s of a set of sequences is considered *maximal* when it's not included in any other sequence in the set of sequences. [19]

In the database D when customers buy items from the supermarket the recorded transactions of their purchases is considered as a sequence, where we can view the transaction as a set of items or itemset, and when the transactions is ordered in a list based on each transactions time-stamp then the ordered list can be viewed as a sequence. Hence, to provide more formal definition of the relation of the transactions and the sequences we can represent the transactions and arrange them based on their chronological order in the following manner: $T_1, T_2, T_3, ..., T_n$ and the set of items in T_i can be symbolized as itemset(T_i). Thus, to produce the customer-sequence for a certain customer the sequence can be expressed as $[itemset(T_1)itemset(T_2)itemset(T_3)...itemset(T_n)]$. [19]

When considering a certain sequence then this sequence can be supported by a certain customer if and only if the sequence is included in the customer-sequence. Therefore, a sequence a is considered supported by customer c if and only if the sequence a is included in the customer-sequence of the customer c. Moreover, the *support* of a sequence is similar to the support of a rule in the association rules, since it represents the percentage of customers who support the sequence among all the customers. Therefore, the main goal in mining sequential patterns is to find all the sequences in the database that satisfy the user-defined minimum support among all the possible sequences in the database, where each maximal sequence is considered as a sequential pattern, also for each sequence that satisfies the minimum support constraint is called a *large* sequence. [19]

The following brief example will provide more clarification to the concepts discussed earlier regarding sequential patterns. We will use the famous supermarket and customers database example as used in Agrawal et al. [19]. In the table 2.3 we are showing a snippet of the database D of the customer transactions where we can see that it had been grouped and ordered by the customer ID and the transaction time-stamp.

Table 2.3: Customer database ordered by customer ID and Transaction Time-stamp. [19]

Customer ID	Transaction Time-Stamp	Items Bought
1	May 14 '09	30
1	May 21 '09	90
2	May 12 '09	10, 20
2	May 16 '09	30
2	May 23 '09	40, 60, 70
3	May 14 '09	30, 50, 70
4	May 14 '09	30
4	May 21 '09	40, 70
4	June 30 '09	90
5	May 11 '09	90

In the table 2.4 we have aggregated all the customer-sequences and arranged them chronologically based on the customer ID and the transaction time-stamp to represent the customer database in a customer-sequence manner which prepares for the next step of identifying the sequences which satisfy the minimum support.

Table 2.4: The Customer database expressed as a customer-sequence. [19]

Customer ID	Customer Sequence
1	[(30)(90)]
2	[(10 20)(30)(40 60 70)]
3	[(30 50 70)]
4	[(30)(40 70)(90)]
5	[(90)]

If we set the minimum support of a sequence as 25%, then we are requiring at least the support of two customers for any sequence. Then the only maximal sequences among all the other sequences are the following two sequences: [(30)(90)] and [(30)(40 70)] which meet with the minimum support defined by the user and as a result those two sequences are the sought after sequential patterns. As for the first sequential pattern [(30)(90)] it had gained the support of the customers 1 and 4. Although customer 4 had not bought the items 30 and 90 one after the other and instead the customer bought the items (40 70) between them, yet we still consider them as a sequential pattern since the goal is not to have patterns which are mandatory contiguous. As for the sequential pattern [(30)(40 70)] it had gained the support of the customers 2 and 4. Also, in the case of the customer 2 we can see that he/she had bought the item 60 beside buying the items 40 and 70, however the customer 2 still supports the sequential pattern due to the fact that the itemset (40 70) is a subset of the itemset (40 60 70). In order to show a sequence which has not satisfied the minimum support constraint we can consider the following sequence: [(10 20)(30)], since it was only supported by customer 2 among the other customers, which means that it lacks the support of an additional customer to be considered as a sequential pattern. Moreover, there other sequences that had met with the minimum support constraint, however they were not maximal sequences, which means that these sequences were part of other maximal sequences and therefore can not be used as a sequential pattern, and example of those sequences is as follows: [(30)], [(40)], [(70)], [(90)], [(30)(40)], [(30)(70)] and [(40 70)]. [19]

2.3.3 Sensor Association Rules and Sensor Sequential Patterns

Sensor Association Rules are a type of behavioral patterns which is used to acquire the temporal correlation between the sensor nodes in a WSN. By using these temporal correlations we can identify the set of sensors which detect events in the same time interval. A significant performance improvement is gained by utilizing Sensor Association Rules. The concepts of utilizing the sensor's behavioral patterns were presented by

Boukerche et al. [27] to be used in order to produce rules that would significantly enhance the WSNs performance and quality of service by introducing a new technique that reformulates the association rules mining problem and customizing it for the sensor's behavioral patterns. Moreover, This technique assists in WSN resource management which also manages to provide a solution for some of the shortcomings resulted by wireless communications. According to Boukerche et al. [27] the generation process of the sensor association rules follow a set of designed phases which is part of the Knowledge Discovery Process. The core phases in the generation process include the following:

- Providing a formal definition of the required sensor behaviorial patterns and sensor association rules.
- Subjecting the designated collected data to certain preparation and transforming procedures in order to fit the generation process. In [27], Boukerche et al. presented a representation structure called the Positional Lexicographic Tree (PLT). PLT's main task was to prepare the collected data for the mining process by compressing the data in a form that assists in mining the sensor behavioral patterns.
- Involving the transformed data into a data mining phase which results in producing the sensor association rules.

Boukerche et al. [27], have also introduced different data extraction methodologies which is utilized in collecting the data needed for the sensor association rules mining process.

In a continuation of the work performed with the sensor association rules in Boukerche et al. [27], Samarah et al. [74] have introduced a new reformulated version of the sequential patterns mining problem called Chronological Patterns which is customized to fit the needs of the sensor's behaviorial patterns where the order of detection is of special interest. According to Samarah et al. in [74], Chronological Patterns can be used as a prediction tool which would improve the performance of the WSN and eventually enhances the overall quality of service. In [74], Samarah et al. present a formal definition of the Chronological Patterns coupled with a new representation structure called the Chlorotical Tree (CT), which expedites the mining process of the Chronological Patterns.

For more details on the use of the chronological patterns (Sensor Sequential Patterns) as a prediction tool refer to section 3.5.1.

Chapter 3

Proposed Prediction-based Tracking Technique: Design and Implementation

In this chapter, we discuss our proposed design for the prediction-based tracking technique and we will be elaborating on all the sections involved in the tracking process and the mechanisms utilized to perform the sought after tasks. It's imperative to mention that we are adopting the network model for OTSNs as proposed in Xu et al. (PES) [1] where we will be using their work as our frame of reference due to the similarities in the problems being solved and since both techniques fall under the same category of object tracking, namely the prediction-based approach. However, it's also worth mentioning that our proposed technique has fundamental differences compared to PES, particularly in the main component of the system which is the prediction model, since in PES they rely on different types of heuristics to achieve accurate predictions, whereas, our prediction technique was designed to utilize data mining approaches namely sequential patterns.

This chapter is divided as follows: section 3.1 presents a formal definition of the problem statement related to the requirements of object tracking applications. The assumptions made in regard to the proposed tracking technique and the application requirements are presented in the section 3.2. In section 3.3 the basic object tracking schemes are presented and explained. In addition, section 3.4 evaluates the basic tracking techniques from the energy consumption perspective and introduces the energy consumption equations for each presented tracking scheme, including the ideal (theoretical) tracking scheme. In section 3.5 we introduce our prediction-based tracking technique which uses sequential patterns as the prediction tool (PTSP), where we show how the sequential patterns are generated from the movement patterns and used to create the prediction model, in addition we present the activation mechanisms and the missing object recovery mechanisms.

3.1 Problem Statement

In this section we will be stating the problem and presenting the requirements of applications designed to track moving objects. Moreover, we will be introducing the two main performance metrics which will be used to determine how efficient our proposed technique compared to other energy saving techniques. Also, we will be presenting the assumptions which we have maintained while designing our proposed technique and some basic OTSN techniques which will be used for comparison purposes.

3.1.1 Application Requirements

- The Wireless Sensor Network has S sensor nodes which tracks O moving objects.
- A *sampling duration* lasts for X milliseconds.
- The reporting of an object location is achieved by sending the Sensor ID where the object is being detected every T milliseconds to the application.

Now we can provide more formulated problem statement which will be based on the application requirements of an object tracking system.

Problem Statement: *Develop an energy efficient technique that adheres to the application requirements of an object tracking system which reduces the total energy consumption of the OTSN while maintaining tolerable missing rate level.*

It was clearly indicated in the problem statement that the two main metrics which will be used to evaluate the performance of the energy saving techniques are the **Total energy consumption** which will be used as to appraise the performance of the energy saving techniques for OTSN and the **missing rate** will represent the percentage of failed attempts to report a timely information regarding the movement of an object inside the network. These metrics will be further discussed in the following Performance Evaluation chapter (refer to chapter 4).

3.2 Assumptions

The following list represents the assumptions we will be using while presenting our proposed technique:

- We assume that the sensor nodes are static and the network topology including the position of each sensor node in the network is well known to the base station.
- Whenever a sensor node is mentioned in this technique we assume it's a logical representation of a group of nodes which work together in order to determine the attributes of the moving object. Therefore, the mentioned sensor nodes in this work are the cluster heads [1].
- The objects moving inside the network will be distinguishable where each tracked object will be assigned an object ID which will make each moving object recognizable to the network [6].
- We assume that the communications between the sensor nodes in the network and the base station will be based on the single-hop communications.

- To allow an active sensor node to wake-up a sleeping sensor node we assume the existence of a low-energy messaging channel [37] [38].
- We assume that synchronization is maintained between all the sensors in the network and the application.
- Due to its insignificance, we will not be considering the energy consumed as an overhead whenever a sensor node changes its mode of operation, such as the change from sleep mode and active mode [9].
- The sensor nodes located in the boundary of the network are outfitted with powerful equipment in terms of power supply, processing power and sensing components to be able to stay active longer than the inner nodes so that it can detect the entrance of any moving object and avoid any missing reports [6].

3.3 Basic OTSN techniques

The following OTSN techniques will be used to compare our proposed technique performance to their performance while tracking moving objects. These techniques are designed to satisfy the application requirement of a basic object tracking system.

- **Naive:** This is a basic object tracking technique where all the sensors in the OTSN are kept active all the time and therefore each moving object in the network will be detected by the sensor nodes and reported to the base station every T milliseconds, see figure 3.1. This technique will be used for comparison purposes as it represents the baseline where progress in energy conservation can be measured, since it's not possible to adjust this technique to produce better results in terms of energy savings [39] [43].
- **Scheduled Monitoring (SM):** In this technique all the sensor nodes in the network are allowed to stay in sleep mode and they change their status to active mode

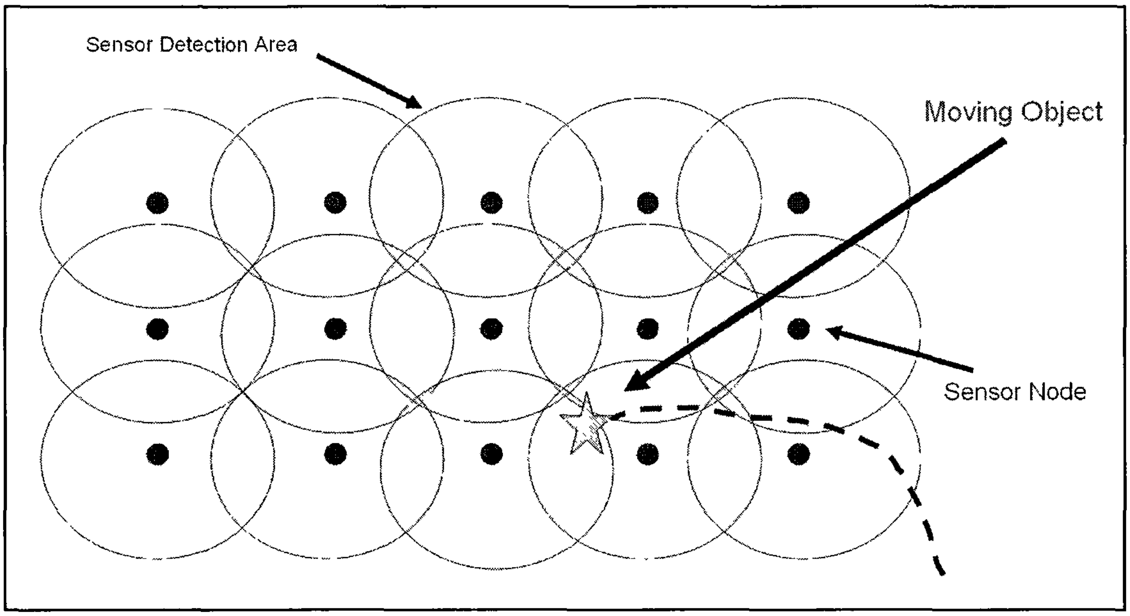

Figure 3.1: All sensors are kept active all the time in the Naive tracking technique.

for a brief period of time where they start sensing their monitored area and report their findings to the base station, given that both the sensor nodes and the base station are well synchronized. A major concept introduced in this technique is the fact that an OTSN network doesn't need to report its collected data all the time. Based on our application requirements, then in this technique all the S nodes are set to active mode for X milliseconds followed by a sleep mode period of (T - X) milliseconds, see figure 3.2. This alternating behavior of the network between sleep and active modes is set to take place during the whole network operating time. It's obviously clear that with the implementation of this technique a major savings in the consumed energy is accomplished compared to the Naive technique, since all the nodes in the OTSN network are scheduled to stay for very short time in active mode and continue to operate in sleep mode as long as possible. On the other hand, the main drawback of this technique is the fact that in order to avoid missing reports there a lot of nodes are being activated to participate in the object detection process while those nodes are not actually required [40] [41] [43].

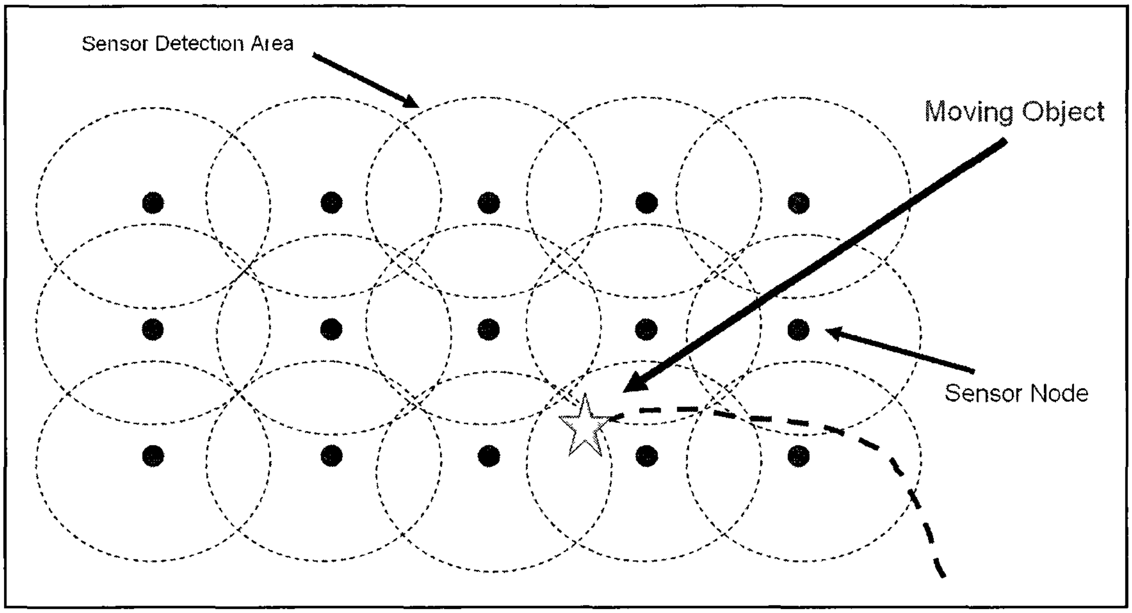

Figure 3.2: Scheduled Monitoring technique: All the sensors are kept in sleep mode during the (T - X) milliseconds.

- **Continuous Monitoring (CM):** This object tracking technique differs from the previous two techniques, since it doesn't involve all the sensor nodes in the OTSN network to track a moving object. As an alternative, this technique allows only one sensor node to stay in active mode continuously while it's detecting and tracking a moving object as long as this object stays inside the node's detection area and the rest of the sensor nodes in the network are kept in sleep mode. The current active node will continue to monitor the moving object in its monitoring area until it leaves its coverage area and begins to enter a neighboring node detection area and at that time the current active node will wake up the destination node to start monitoring the moving object, then the current active node will change to sleep mode while the destination node will become the current active node. The current active node sends its activation message to the destination node W milliseconds before the object enters the destination node detection area. Determining the duration of the

W milliseconds is based on the transmission rate and the moving object velocity which usually means a very small value for W. Hence, once a moving object reaches the edge of the detection area of the current active node it will be able to determine the destination node by calculating the velocity and direction of the object then a handover will take place where the current active node wake-up the destination node, see figure 3.3. Therefore, when an object starts moving between sensor nodes then they will have to work together in order to keep tracking the object movement and send their reports to the base station on the scheduled time for reporting. This technique key benefit is the fact that only one sensor node is used to track each moving object, which allows the rest of the noes in the network to maintain their sleep mode status as long as possible, hence preserving their energy. Nonetheless, in order to avoid any missing reports the current active node will have to keep its active mode status during the time when the moving object still inside its detection area, which eventually will lead to the depletion of the current active node energy. [42] [43].

3.4 Energy Consumption Analysis in OTSNs

When we analyze the performance of the aforementioned techniques we can determine that all those techniques weren't allowing any missing reports to take place. Therefore, the resulted missing rate level is 0%, while this could indicate a successful monitoring process, it comes on the expense of the network's energy consumption levels which is suffering badly due to the fact that both MCU and sensing components are being switched to active mode for long periods that mostly goes to waste. When we analyze the differences between the SM technique and the Naive technique we can deduce that if we lower the sampling frequency of all the senors in the Naive technique instead of keeping them active all the time, then the result will be a tracking technique which is exactly identical to the SM technique. Also, in a similar case, if we reduce the number of sensor

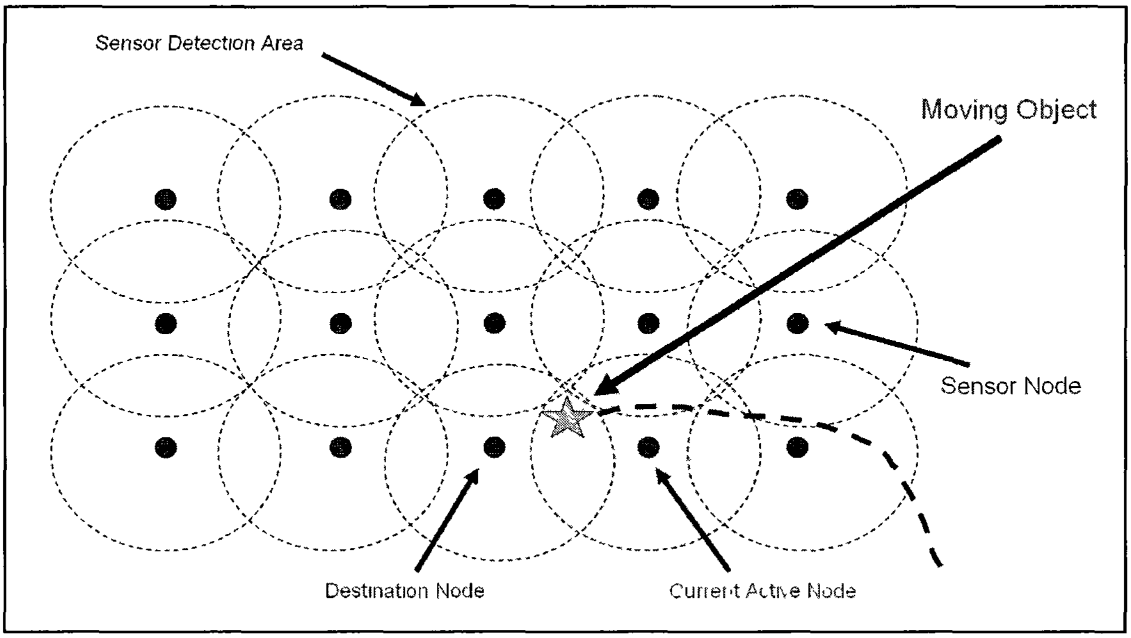

Figure 3.3: Continuous Monitoring technique: A handover is performed when the moving object reaches the detection area boundary of the current active node and starts entering the destination node's detection area.

nodes involved in the tracking of a moving object in the Naive technique to a single sensor node, then we will be creating a tracking technique with high similarities to the CM technique.

Furthermore, by analyzing both the SM and the CM techniques and how they contribute to energy conservation in OTSNs compared to the Naive technique, we can conclude that both techniques reduce energy consumption in a different way. While, the SM technique relies on having a low sampling frequency to preserve the OTSNs energy, the CM technique attempts to conserve its OTSN energy by reducing the number of actively tracking sensors to only one sensor. Hence, in order to design a tracking technique which will satisfy the tracking applications' requirements of receiving a timely updates regarding any moving object inside the OTSN while maintaining a low energy consumption level compared to other tracking techniques, then we need to lower the following two factors and keep them as minimum as possible, which consequently prolongs the OTSN network operational period:

- **Sampling Frequency.** (refer to 2.2.2 in the Background and Related Work chapter for more information).
- **Number of sensor nodes participating in object tracking.**

As a result of the impact of the previously mentioned factors, then we will attempt to design a tracking technique which possesses the ability to use the minimum number of sensor nodes to track a moving object and these nodes must not operate all the time in active mode, rather they should be able to switch to sleep mode to further enhance their energy preservation efforts. Therefore, what we are looking for is a prediction-based technique which allows the network to be able to identify which sensor node to activate and at what time, to avoid a missing report. However, having a perfect prediction technique is yet to be developed, hence we will accept a fair compromise where we can produce adequate and accurate predictions via an energy efficient object tracking technique while maintaining an acceptable level of missing reports which is represented by the application's tolerable missing rate.

In addition, based on how the introduced OTSN techniques perform their monitoring and reporting processes, it was derived that these techniques had different methods of how to monitor their detection areas, however they had similar procedures regarding how to report and communicate their gathered data. Thus, we will be considering the energy consumed mainly by the MCU and the sensing components in our work. The following table 3.1 [1] provides the energy consumption equations for each tracking technique mentioned previously and in addition to the hypothetical ideal tracking technique based on the tracking application's requirements. All the tracking techniques in table 3.1 have a 0% missing rate.

- E_{active}: The energy consumed by a sensor node per second while its MCU and sensing components are kept in active mode.

Table 3.1: **Energy consumption analysis in OTSN techniques. [1]**

Technique	Energy Consumption Equation	All Nodes?	Monitoring Type
Naive	$E_{active} \times TO \times S$	Yes	Continuous
SM	$(E_{active} \times X + E_{sleep} \times (T - X)) \times \frac{TO}{T} \times S$	Yes	Scheduled
CM	$(E_{active} \times TO) + (E_{sleep} \times TO \times (S - 1))$	No	Continuous
Ideal (In Theory)	$(E_{active} \times \frac{TO}{T} \times X) + (E_{sleep} \times (TO \times S - \frac{TO}{T} \times X))$	No	Scheduled

- E_{sleep}: The energy consumed by a sensor node per second while its MCU and sensing components are kept in sleep mode.
- ***TO***: The total time in seconds in which the OTSN has operated in a continuous manner.
- ***S***: Represents all sensor nodes in the network.
- ***X***: Represents the sampling duration in milliseconds. During X the sensor is kept in active mode.
- ***T***: Represents the period of time in milliseconds which takes place between each report and another sent by a sensor node to the base station.
- ***(T-X)***: It's the time in milliseconds which begins once a report to the base station was made by the sensor node and lasts until (X) the sampling duration starts. During *(T-X)* the sensor node is kept in sleep mode.
- $\frac{TO}{T}$: This is the total number of reports during the entire network's operational period.

After examining the energy consumption equations it becomes clear that the Naive technique is the most energy consuming tracking technique, due to the fact that it requires all the sensors in the network to remain in active mode all the time, despite that the number of reports to the base station doesn't exceed $\frac{TO}{T}$. Moreover, we can also conclude that the hypothetical ideal tracking technique is the most energy conserving technique, since it keeps all of the sensor nodes in the network in sleep mode for longer periods than the other techniques, thus saving more energy.

3.5 Proposed Prediction-based Tracking Technique (PTSP)

It was obvious from the energy consumption analysis that the Ideal technique was the best in terms of both the energy preservation and the quality of tracking, since it doesn't allow any missing reports and therefore its missing rate level is kept at 0%. However, in order to design the ideal tracking technique then this system must be able to determine the exact time and place an object will be at, so that it can activate the node which the moving object will moving under its detection area, so what we are looking for is a system that produces perfect and 100% accurate predications. Given that the development of such perfect predictions system is not possible then we can try to reach the next best thing which is a tracking technique that's capable of making good predictions and reducing the number both the sampling frequency and number of nodes involved in the tracking process.

Therefore, our proposed system (Prediction-based Tracking using Sequential Patterns (PTSP)) will be based on two stages as follows:

- **Stage 1: Sequential Patterns Generation.**
- **Stage 2: Object Tracking and Monitoring.**

The first stage involves building the prediction model where we will be collecting a huge log of data related to objects movements from the sensor nodes in the network, where it will be sent to the sink which will aggregate a large enough sum of data and store them in a database in order to produce the inherited behaviorial patterns of object movement in the monitored area. Based on this data the sink will be able to utilize data mining mechanisms in order to generate the sequential patterns which will be deployed by the sink to the sensor nodes in the network to allow the sensor nodes to predict the future movements of a moving object in their detection area.

In the second stage the actual tracking of moving objects starts and this stage has two parts: (1) *activation mechanism* , which entail the use of the sequential patterns to predict which node(s) should be activated in order to continually keep tracking the moving object; (2) *missing object recovery mechanism*, which will be used to find missing object in case the activated node is not able to locate the object in its detection area.

The key objective in our prediction technique is to keep in sleep mode for the longest possible period any sensor node that has no object moving in its detection area, and thus saving its energy. Moreover, in the case of a moving object in the vicinity of a certain sensor node then this sensor node will not be wake all the time and ought to switch to sleep mode as long as possible, while not impairing the tracking process, and this sensor node will be called the *current sensor.* Therefore, the current sensor will be utilizing the predication model to determine the next potential position of the moving object, and thus decide on which sensor nodes it will activate, called *destination sensors*, who will start waking up after a specified sleeping time in order to collaborate in the tracking of the moving object Also, after signaling the destination sensors with the activation message, the current sensor will switch to sleep mode and will wake up with the destination sensors to detect and track the moving object.

The current sensor switches to active mode for X milliseconds and during this time the current sensor senses its detection area and reports the findings to the base station by the end of X which also matches the end of T the time period between reports, figure 3.4 illustrates the relation between X, T and $(T-X)$. The last action the current sensor performs before switching to sleep mode is predicting the location of the moving object during the $(T-X)$ milliseconds and sending a message to the destination sensors to wake up after the $(T-X)$. Both the current sensor and the destination sensors will activate after the passing of the $(T-X)$ milliseconds and start tracking the moving object. In addition, once one of those active sensor nodes detects the object in its vicinity it will send an acknowledgment message (ACK) to the former current sensor, unless it's the current sensor, in order to become the new current sensor and starts the previously illustrated process, whereas the rest of the active sensor nodes will turn to sleep mode.

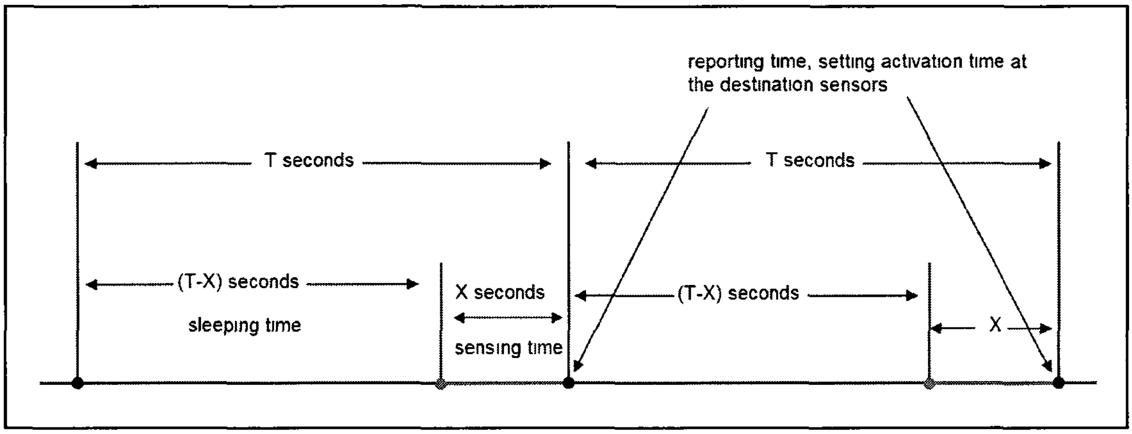

Figure 3.4: Demonstrating the relation between X, T and $(T-X)$ in the current sensor. [1]

It's imperative for any current sensor to be able to predict the next movement of the object inside its detection area, so that it can choose which sensor nodes to activate for a collaborative tracking, since the moving object may leave the detection area of the current sensor while it's in sleep mode during the $(T-X)$ milliseconds. Therefore, if the current sensor didn't activate any destination sensors then if the moving object leaves the area covered by the current sensor while it's sleeping and no destination sensors were activated then the application will not receive any data regarding the object's location although the object still inside the network. Hence, the main two questions to be answered are: (1) how to decrease the number of objects gone missing and (2) how to find a missing object.

As for the first question, the answer would be by using a prediction technique and collaborative tracking method, where the current sensor allows other sensor nodes to help in the tracking process of a moving object. Basically, the current sensor informs the destination sensors to activate at a certain time when both the current sensor and the destination sensors wake up at the same time. As a result, if the current sensor node could not find the object, then there is a higher probability that one of the destination sensors was able to detect the object in its vicinity. Then, the destination sensor will send an ACK message to the current sensor. Once the current sensor receives an ACK message from any of the destination sensors during a specified time it will change its status to a non-current sensor and it will switch to sleep mode, unless there is another object in its detection area. On the other hand, if the current sensor didn't receive any ACK messages from the destination sensors then a missing object is confirmed and the current sensor will initiate a missing object recovery process. So, to answer the second questions we need to determine which sensor nodes should be activated in order to cooperate in the hunt for the missing object. These questions will be answered in the following sections.

3.5.1 Sequential Patterns Generation and Prediction Model

Sequential Patterns Definition

In this section we will present a definition of the sequential patterns used in our prediction model following the definition provided by Samarah et al. [74] which is inspired by the definition of sequential patterns by Agrawal et al. [19].

Let S represent the set of sensors in a certain WSN. The list

$L = [(s_1, t_1), (s_2, t_2), (s_3, t_3), ..., (s_m, t_m)]$ denotes a list that contains the pairing of each sensor detection (represented by the sensor ID) and its equivalent time of detection, where $s_i \in S$ and $t_i <= t_{i+1}$ for all $1 <= i < m$. Therefore, the pair (s_i, t_i) represents the detection of a certain event by the sensor s_i which took place at the time t_i. [74]

The list shown in $O(L) = [s_1, s_2, s_3, ..., s_m]$ represent the list of sensors shown in L, except that it was ordered chronologically based on the time of detection recorded by each sensor. [74]

The pattern shown in $O = [s_1, s_2, s_3, ..., s_m]$ represents a sub pattern of the pattern

$O' = [s'_1, s'_2, s'_3, ..., s'_n]$, which can be interpreted as $O \sqsubseteq O'$, if $m <= n$ and a list of incrementing integers exists as $i_1, i_2, i_3, ..., i_m$, where $1 \leq i_1 < i_2 < ... < i_m \leq n$, and $s_k = s'_{ik}$ for $k = 1, 2, .., m$. Thus, O is designated as the "sub pattern", whereas O' is called the "super pattern". [74]

The database of sensor list is denoted by DS where for each list in the DS is coupled with a time called GT (Global Time), and GT provides the ordering criteria for each sensor in a list. Hence, by having DS as a database of lists and O as the sequential pattern, then the *support* of the sequential pattern O in DS can be described as the number of lists in DS where the sequential pattern O represents a sub pattern of the lists' sequential patterns. Thus, $Support(O) = |\{L \in DS | O \sqsubseteq O(L)\}|$. [74]

Therefore, the main goal in mining the sensors sequential patterns with a database of lists DS and a minimum support (*min_sup*), is to identify all the sensor sequential patterns which satisfies the user-defined minimum support and these patterns are called frequent sensor sequential patterns. In order to commence the patterns discovery process, the first phase begins with aggregating the data of the sensors' detections and arrange them in a chronological order as explained in the section 3.5.1 titled "Movement Patterns Acquisition" which will describe our methodology of generating the sequential patterns in order to use them in our prediction model. [74]

Sequential Patterns and Tri-Sensor Patterns

We have discussed sequential patterns earlier in this thesis (refer to section 2.3.2) and since we are using sequential patterns as a mean of creating our prediction model we are going to present a brief explanation of the type of sequential patterns we are using in our prediction model, namely the Tri-Sensor patterns which can be considered as a special form of the sequential patterns, since all the sequential patterns generated in the form of Tri-Sensor patterns are made of only three items, hence the prefix Tri in the patterns name "Tri-sensor patterns". Also, all the items in the tri-sensor patterns are of sensor IDs, and the ordering followed in these patterns is of great importance, since it can be represented as follows: $[SourceSensor, CurrentSensor, DestinationSensor]$, which represent the chronological ordering of the sensors' detection of a certain moving object in the network. Therefore, the *source sensor* represents the sensor ID of the sensor which detected a particular moving object o, before it moved to the next sensor designated as the *current sensor*, and later on the object o moves to the next sensor denoted by *destination sensor*.

Furthermore, in order to evaluate the statistical significance of a certain tri-sensor pattern, we have to calculate the *confidence* of each particular tri-sensor pattern, which can be viewed as the estimate of the probability $P(Y|X)$, which is the probability of finding the right-hand-side (RHS) of the pattern in sequences while these sequences also contain the left-hand-side (LHS) of the pattern. Therefore, the confidence of tri-sensor pattern $[sourcesensor, currentsensor] \implies [destinationsensor]$ is calculated by dividing the support of the pattern $([sourcesensor, currentsensor, destinationsensor])$ by the support of the pattern $[sourcesensor, currentsensor]$, which can be represented as follows:

$$conf(\text{[source sensor, current sensor]} \implies \text{[destination sensor]})$$

$$= \frac{supp(\text{[source sensor, current sensor, destination sensor]})}{supp(\text{[source sensor, current sensor]})}.$$

For further explanation of the concept of *support* and *confidence* please refer to the section 2.3.

Movement Patterns Acquisition

The first major part of our proposed technique is generating object movement patterns. In real world and depending on the terrain being monitored and types of objects that moves i.e. animals, enemy movements, etc ... certain paths tend to occur more frequently. For example, in a forest area, there maybe only 2 or 3 available routes for animals to reach a water source because other paths are blocked by heights or other obstacles. Our scheme relies on deploying the WSN and simply turning it on for the whole time to record movement patterns collected from sensors (basically the naive technique, see figure 3.5). As every object moves, its movement path and the sequence of sensors that detect it gets recorded into a database.

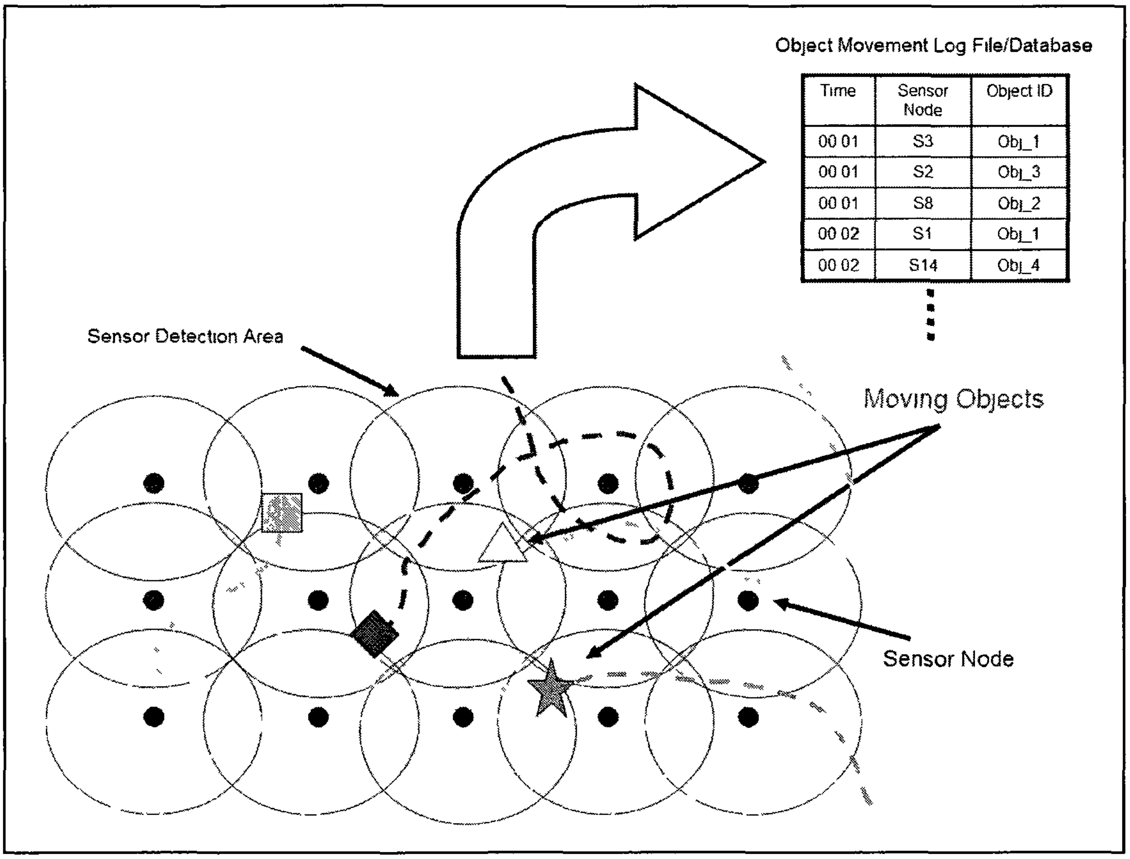

Figure 3.5: Recording objects' movements using Object ID along with the Sensor ID of the detecting sensor node and time, to produce Object Movement History Log File/-Database.

As shown in the table 3.2 the first stage of the movement patterns generation is started by recording all objects movements with their respective objects IDs and the ID of the sensors that detect each object along with the time of detection.

Table 3 2 Objects Movement Log

Time	Sensor ID	Object ID
00 01	S3	Obj1
00 01	S9	Obj4
00 01	S4	Obj2
00 01	S12	Obj3
00 02	S6	Obj1
00 02	S5	Obj2
00 03	S11	Obj4
00 03	S14	Obj3
00 04	S8	Obj1

After having sufficiently executed this monitoring and recording process, the next step is to analyze this data and convert it into movements patterns We use tri-sensor movement patterns A movement pattern simply records the *source sensor* of a certain object (the sensor where the object used to be before moving towards the current sensor), the *current sensor* and the sensor the object traveled to next, or as we call it the *destination sensor* For example, if an object moved through the following sequence of sensors S3, S6, S8, S7, We record an instance of the pattern S3, S6, S8 and pattern S6, S8, S7 Therefore, we can utilize the data collected in the object history movement log file (table 3 2) and transform it into sequences of sensors trigged by each object movement where each sequence of sensors will be transformed into tri-sensor patterns in the following form $[SourceSensor, CurrentSensor, DestinationSensor]$ and these patterns are generated in an overlapping manner where the *Current Sensor* becomes the *Source Sensor* and the *Destination Sensor* becomes the *Current Sensor*, whereas the *Destination Sensor* will be the next sensor in the sequence Figure 3 6 demonstrates how the tri-sensor patterns are generated in an overlapping manner from the sequence of detections by the sensor nodes of the OTSN for a particular moving object

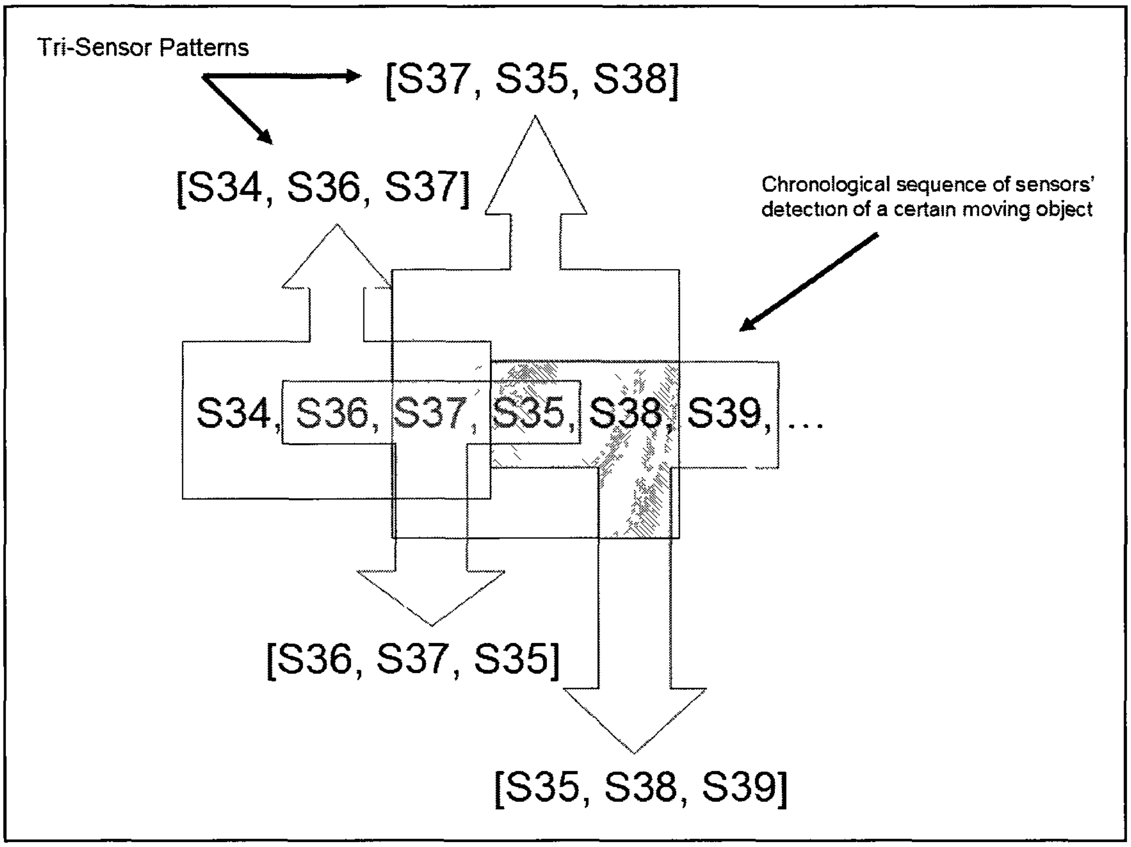

Figure 3.6: Tri-sensor patterns generated in an overlapping manner from the chronological sequence of sensors' detections of a particular moving object.

Table 3.3 shows how we aggregate the sequence of sensors' detections for a certain object in order to generate the tri-sensor patterns which will be used to produce the patterns for prediction. We would like to note that the sequence of sensors shown in table 3.3 are arraigned based on the chronological order of detection in the log file (table 3.2) for each particular object.

Table 3.3: Generating the tri-sensor patterns from the sequence of detections by sensor nodes for each object.

Object ID	Chronological Sequence of Sensors' Detection	Tri-Sensor Patterns Generated
Obj1	S3, S6, S8, S7, S5, S6, S4, S5,...	[S3, S6, S8], [S6, S8, S7], [S8, S7, S5], [S7, S5, S6], [S5, S6, S4], [S6, S4, S5],...
Obj2	S12, S15, S16, S17, S18, S19, S21, S20, S18,...	[S12, S15, S16], [S15, S16, S17], [S16, S17, S18], [S17, S18, S19], [S18, S19, S21], [S19, S21, S20], [S21, S20, S18],...
Obj3	S52, S51, S54, S53, S50, S48, S46, S47, S49, S51,...	[S52, S51, S54], [S51, S54, S53], [S54, S53, S50], [S53, S50, S48], [S50, S48, S46], [S48, S46, S47], [S46, S47, S49], [S47, S49, S51],...
Obj...	S..., S..., S...,...	[S..., S..., S...], ...

As the patterns accumulate, we can find the most repeated patterns and draw conclusions there. For example, if we found that the pattern S3, S6, S8 occurs too frequently, we can tell that if an object is at S6 coming from S3 then most probably it will head to S8. We call this process, sequential patterns generation. Here is an example of how the tri-sensor patterns get transformed into sequential patterns. Table 3.4 shows sequential patterns collected based on their occurrence from the generated tri-sensor patterns in the previous step shown in table 3.3.

Table 3.4: Aggregating Tri-sensor patterns based on their occurrence.

Source Sensor	Current Sensor	Destination Sensor	Occurrence
S77	S87	S86	739
S77	S87	S88	191
S96	S87	S86	24
S96	S87	S88	936
S92	S90	S93	571
S92	S90	S91	1052

We can tell that if an object has been detected at sensor S87 and coming from sensor S77 then it is more probable for that object to move to sensor S86 than to S88 since it occurred more. Likewise, if coming from S96 to S87, the strongest possibility is moving to sensor S88 since it occurred 936 out of 960 times, almost all the time. Applying simple mathematics we can derive the following sequential patterns in table 3.5 from the above patterns.

Table 3.5: Sequential patterns generated from tri-sensor patterns frequency of occurrence.

Source Sensor	Current Sensor	Destination Sensor	Confidence
S77	S87	S86	79.5%
S77	S87	S88	20.5%
S96	S87	S86	2.5%
S96	S87	S88	97.5%
S92	S90	S93	35.2%
S92	S90	S91	64.8%

Therefore, prediction is achieved using these sequential patterns. The higher the confidence, the better the chance of predicting the next sensor right. As we run our technique, we should also keep updating these patterns as new objects move. This makes our scheme adaptive to changing movement patterns which we like to call it **active learning**.

Furthermore, the sequential patterns generated in table 3.5 can also be represented using the implication form, which can be shown as follows:

$[S77, S87] \implies S86, Conf = 79.5\%$

$[S77, S87] \implies S88, Conf = 20.5\%$

$[S96, S87] \implies S86, Conf = 2.5\%$

$[S96, S87] \implies S88, Conf = 97.5\%$

$[S92, S90] \implies S93, Conf = 35.2\%$

$[S92, S90] \implies S91, Conf = 64.8\%$

It's also worth mentioning that these patterns are shown as examples, where in reality the number of occurrences and the diversity of the generated patterns are higher than what's presented in these demonstrations.

Sensor Sequential Patterns Deployment

In this step we will be embedding all the generated sensor sequential patterns to their respective sensor nodes which resembles the second and final step in building the prediction model of our tracking technique, however these sequential patterns will be continuously evaluated and updated to maintain high accuracy in terms of prediction abilities of our technique. Figure 3.7 shows an example of storing certain sensor sequential patterns in a sensor node.

3.5.2 Sensor Node Activation Mechanism

After the completion of the first stage of our tracking technique (Sequential Patterns Generation) then the second stage starts which is Object Tracking and Monitoring. In this section we will utilize the prediction model created previously, namely the sensor sequential patterns, in order to choose which sensor nodes to activate in order to continue tracking the moving object. However, since there are no 100% accurate predictions we will introduce other methods where more than one sensor node is activated instead of the destination sensor only.

(1) The first method is called (**Current sensor-Destination sensor**), which is the most energy conservative method. In this method the current sensor detects an object in its detection area and based on the embedded sensor sequential patterns the current sensor will decide which destination sensor it will activate, se figure 3.8. The decision on which sensor sequential pattern(s) to apply is based on what is the minimum confidence level the application would require from any sequential pattern before it could be used to predict the future movements of a moving object. Therefore, if the application sets the minimum confidence level at 70%, then any sequential pattern with a confidence

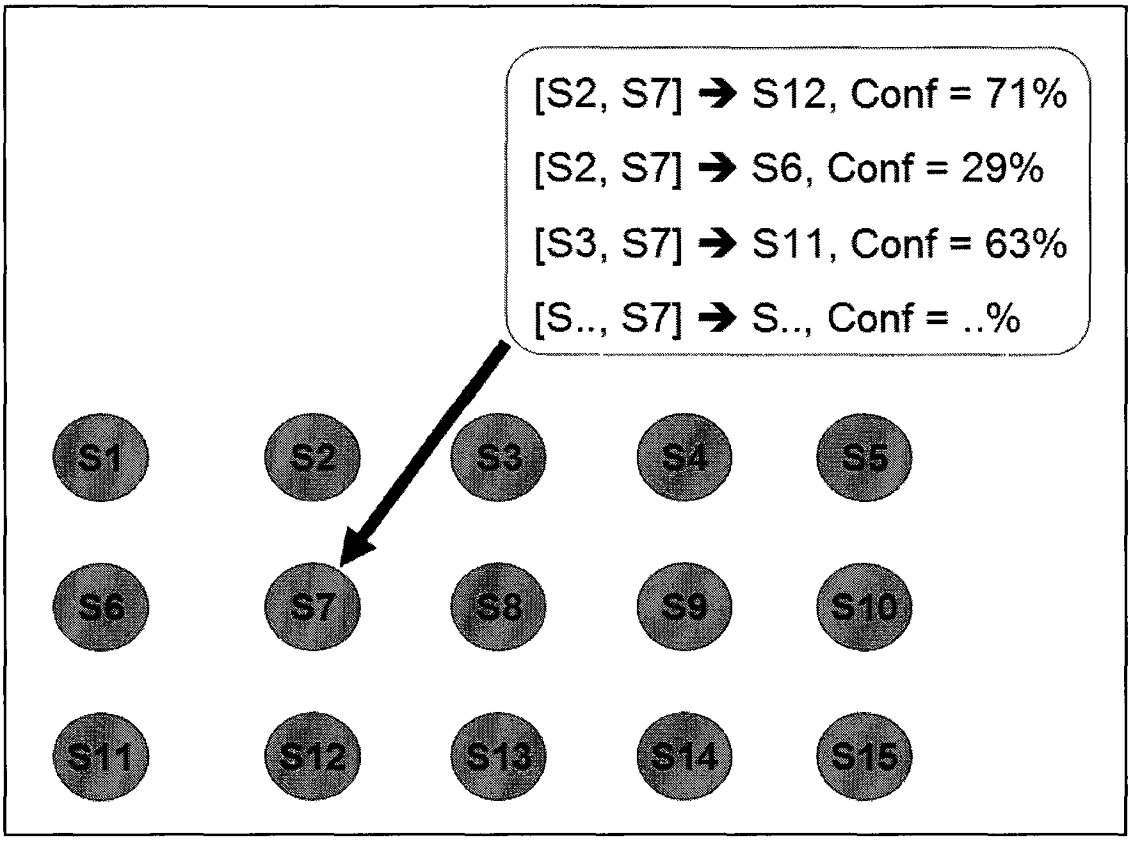

Figure 3.7: An example that shows certain sensor sequential patterns being stored in sensor node S7 in the network.

level lower than 70% will not be executed. Determining the most appropriate minimum confidence level requires a thorough analysis of the prediction model performance which will be discussed in future research. Also, in our simulation we have only implemented this activation mechanism, while the other two activation mechanisms will be the subject of future research.

(2) The second method is called (**Current sensor-Path sensor(s)-Destination sensor**) and in this method the current sensor is activating the sensors between the current sensor and the destination sensor, or as we call them the path sensors, in addition to the destination sensors. In this method, we are trying to tolerate any changes in the object's moving speed, so that if the object slows down before reaching the destination

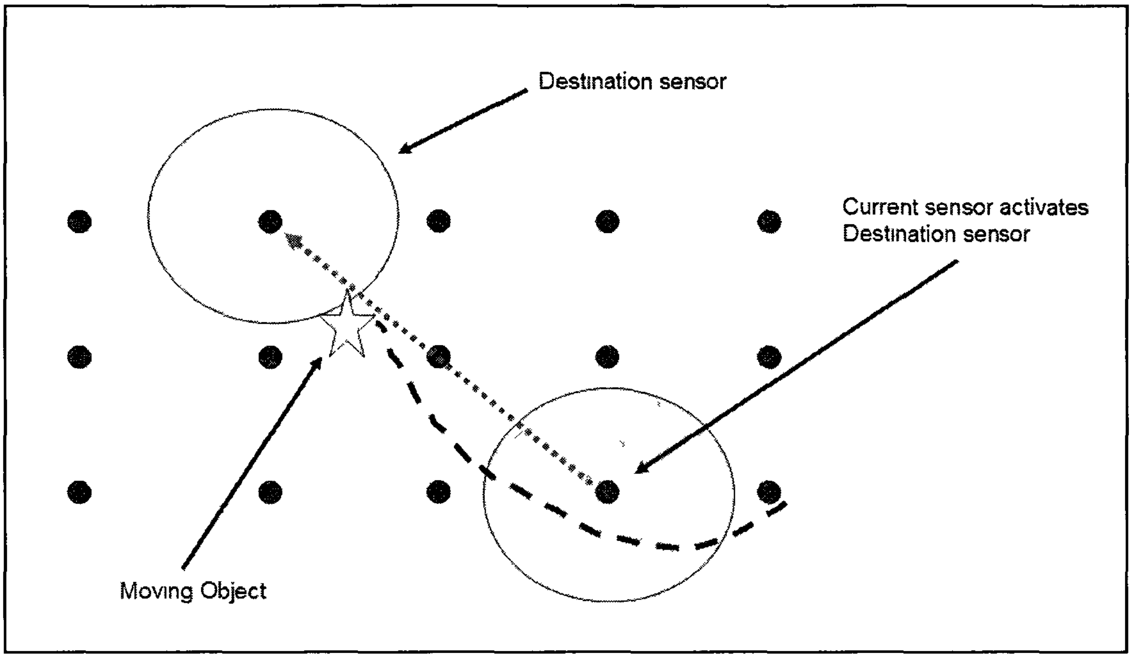

Figure 3.8: The current sensor is only activating the destination sensor based on the sensor sequential patterns.

sensor then the sensors in its path will be able to detect it and avoid a missing object, however if there are no sensor nodes between the current sensor and the destination sensor then this method will be similar to the first method, see figure 3.9. [1]

(3) The third activation method is called (**Current sensor-Neighboring sensors-Destination sensor**) and this is the most energy consuming activation method, since we will be activating all the neighboring sensors of the current sensor, path sensor(s) and the destination sensor. In this method we are trying to mitigate the effect of the changes in the direction and/or speed of the moving object which affects the accuracy of the predictions, see figure 3.10. [1]

We can conclude that using the third method the number of missing objects will be decreased dramatically compared to the other two activation methods. However, activating all the neighbors is an energy consuming method that should be avoided unless we are dealing with moving objects with very high probabilities of changes in

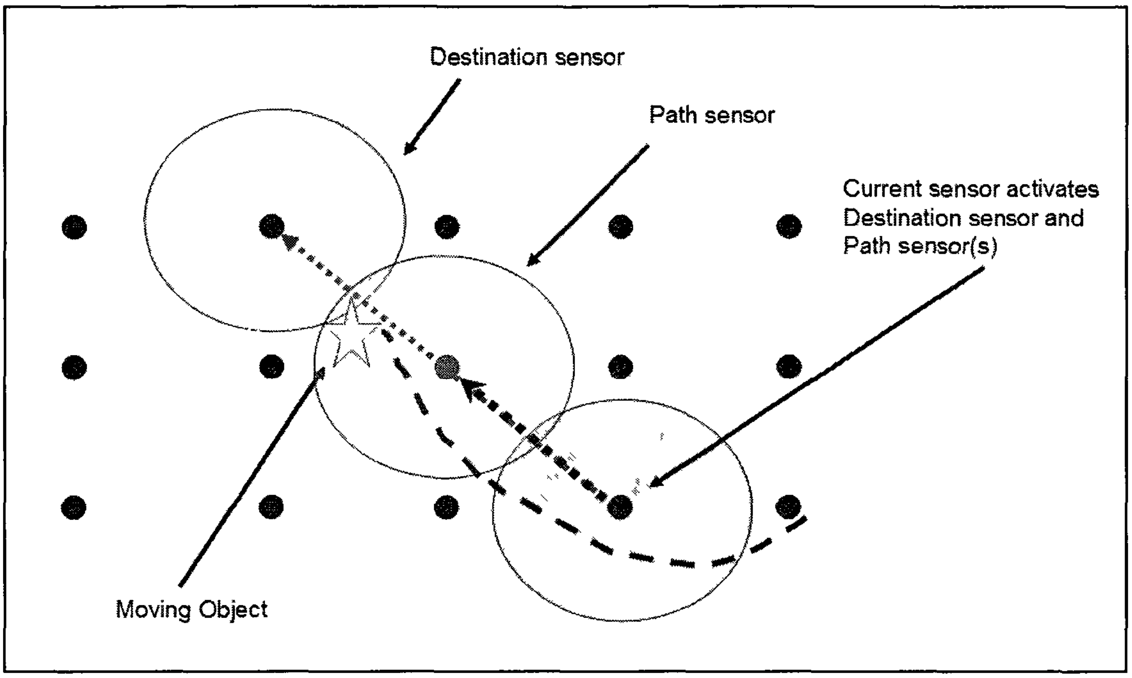

Figure 3.9: The current sensor is activating the path sensor(s) in addition to the destination sensor. [1]

both speed and direction. On the other hand, the (current sensor-destination sensor) and (current sensor-path sensor(s)-destination sensor) activation methods provide more energy conservative approaches. [1]

3.5.3 Missing Object Recovery Mechanism

Since we are using predictions to determine the future movements of a moving object, we also know that there are no prediction technique that could result in 100% accurate predictions. In addition, even if we use the most missing object conservative activation mechanism, which is the (current sensor-neighboring sensors-destination sensor), we still expect to have a missing object running in the network. Therefore, we need to develop a solution to find any missing object and return the network to the prediction tracking process (or as we call it the normal state), since any recovery technique will incur higher energy consumption than the normal prediction tracking technique.

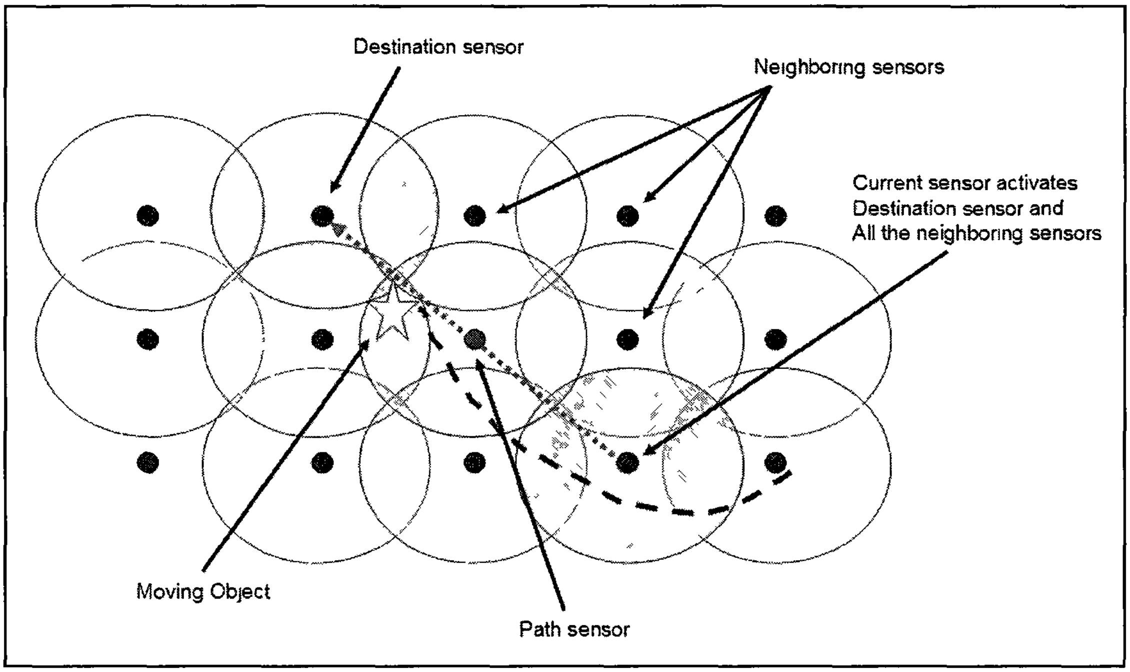

Figure 3.10: The current sensor is activating all the neighboring sensors in addition to the destination sensor. [1]

We have developed three recovery mechanisms in order to determine which recovery mechanism would generate lower energy consumption compared to the others.

- **Source Recovery Mechanism:** In this recovery mechanism the current sensor will activate all its neighboring sensor nodes if the object is not in its detection area and it didn't receive any ACK message from the destination sensor(s) after the passing of a certain timeout period.

- **Destination Recovery Mechanism:** This recovery mechanism is similar to the previous one, except that the current sensor will activate all the neighboring sensors of the destination sensor instead of the neighboring sensors of the current sensor.

- **All Neighbors Recovery Mechanism:** This recovery mechanism combines both previous recovery mechanisms since the current sensor will activate all its neighboring sensor nodes in addition to the destination sensor neighboring sensor nodes.

No matter what recovery mechanism used, if the missing object is not found then the current sensor will initiate the second step of recovery which is called *all network recovery* and in this step the current sensor will be sending an activation message to all the sensor nodes in the network to start looking for the missing object. If no sensor node finds the missing object during the second phase of recovery then the object must have left the network monitored region. In both recovery phases the current sensor uses an ultra low-energy messaging channel to activate the sensor nodes for recovery. The missing object recovery message includes the time at which the sensor nodes will activate and start looking for the missing object. During the first phase of recovery if any sensor node locates the missing object then it must inform the current sensor through an ACK message, in order to avoid the initiation of the second phase of recovery which involves all the sensor nodes in the network. In the second phase of recovery the ACK message is not required since if the object is not found then it must have left the network, and if it's found then the sensor node which located the missing object will automatically become the current sensor. [1]

Chapter 4

Performance Evaluation and Analysis

After having defined the problem we are tackling and laying out existing techniques and solutions that try to solve it and proposing our own method; we show in this chapter the simulation results of our proposed technique. In section 4.1, we formally define the influencing parameters and the comparison metrics we used in our experiments. Section 4.2 outlines the simulation software we implemented in details. It shows how we simulated movement patterns acquisition and how sequential patterns were derived from these patterns. It also shows the different parameters were incorporated into the simulation and how results were calculated and persisted. We finally show the performance results in section 4.3 and analyze the advantages and disadvantages of our work depending on different parameters and in comparison to other energy saving schemes.

4.1 Experiments Parameters and Metrics

In order to evaluate a method in a comprehensive manner, different scenarios and settings have to be considered. Factors relating to the objects being monitored and the wireless sensor network setting can greatly influence the results. The parameters we used in our experiments are listed below:

- **Number of Monitored Objects:** This is an important factor as we expect the energy consumption to rise with increasing number of moving objects.
- **Object Speed:** The movement behavior of object is an important factor. We will use different object speeds to find out how it affects performance.
- **Sensor Network Reporting Frequency:** The amount of time between each report to the base station.
- **Sensor Sampling Frequency:** The number of times a sensor samples per reporting instance.
- **Sensor Sampling Duration:** The amount of time taken for each sampling.

We will show different charts depicting performance under variations of the above parameters and we will thoroughly analyze their effect on the results.

We measure performance according to two main metrics:

- **Total Energy Consumed:** The amount of energy consumed by the whole network to monitor the moving objects which include the active and sleep modes during the simulation
- **Missing Rate:** The missing reports percentage to the total number of reports that should have been received by the application.

4.2 Developed Simulation Framework

We have implemented a software prototype that simulates the different energy schemes for the different scenarios we explained in the previous section. The software is written in C# and it fully implements the features we have discussed in the previous chapter.

We have used a database to store the experiments parameters and results. This database contains all the information needed to provide the performance evaluation charts.

4.2.1 Simulation Assumptions

The following assumptions were considered in the development of our simulation software and they include the assumption made in 3.2:

- The objects moving inside the network will be distinguishable where each tracked object will be assigned an object ID which will make each moving object recognizable to the network.
- We assume that the communications between the sensor nodes in the network and the base station will be based on the single-hop communications.
- To allow an active sensor node to wake-up a sleeping sensor node we assume the existence of a low-energy messaging channel.
- Due to its insignificance, we will not be considering the energy consumed as an overhead whenever a sensor node changes its mode of operation, such as the change from sleep mode and active mode.
- Since our approach aims at reducing the energy consumed by the MCU and the sensing components in the sensor node, we will not include the energy consumed by the radio components.
- We have only implemented the first activation mechanism in our simulation, which is the **Current sensor-Destination sensor**. In this method the current sensor only activates the destination sensor(s) and non of the path sensor nodes or the neighboring sensor nodes.

4.2.2 Sequential Patterns Derivation

The movement pattern acquisition was implemented by defining a set of *key paths.* A key path is a path which is usually taken by objects in the monitored terrain. As explained in the previous chapter, the key paths are terrain and object dependent. In our simulation we provide the following user interface to simulate a dynamic set of key paths.

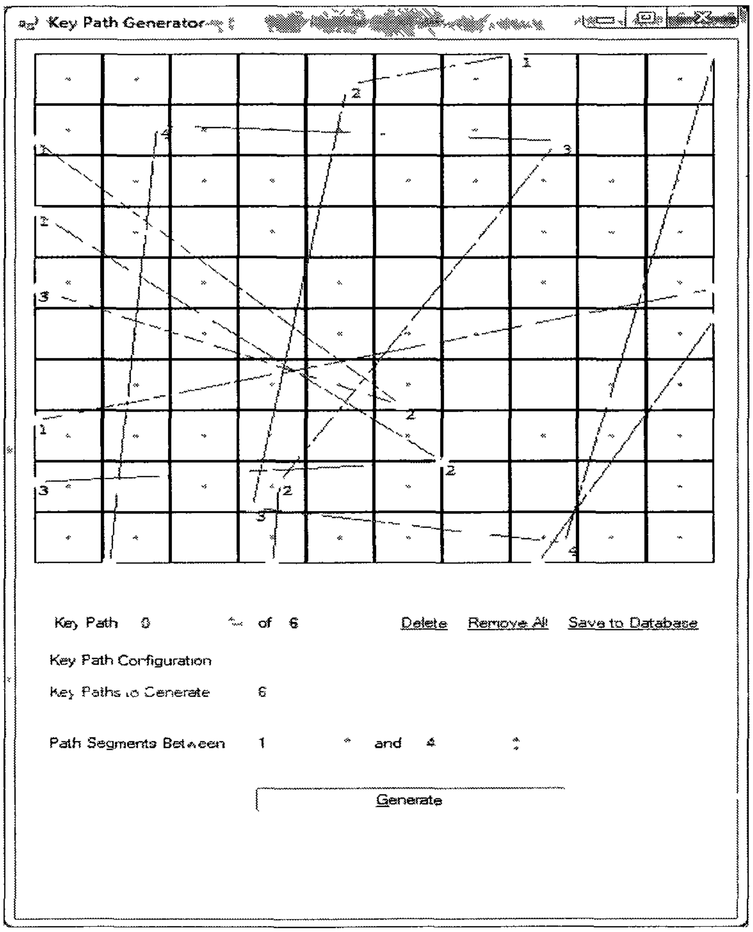

Figure 4.1: Key path generation user interface.

The user selects the number of key paths they wish to include in the monitoring area. The complexity of the path represented in the number of line segments is chosen too. given these inputs, the software generates random points and connects them forming the key paths. After generating the paths, the user can preview the paths and save them in the experiments database for upcoming simulation experiments.

After obtaining the key paths, the next step is to use the naive approach to capture object movement patterns. Figure 4.2 depicts this process.

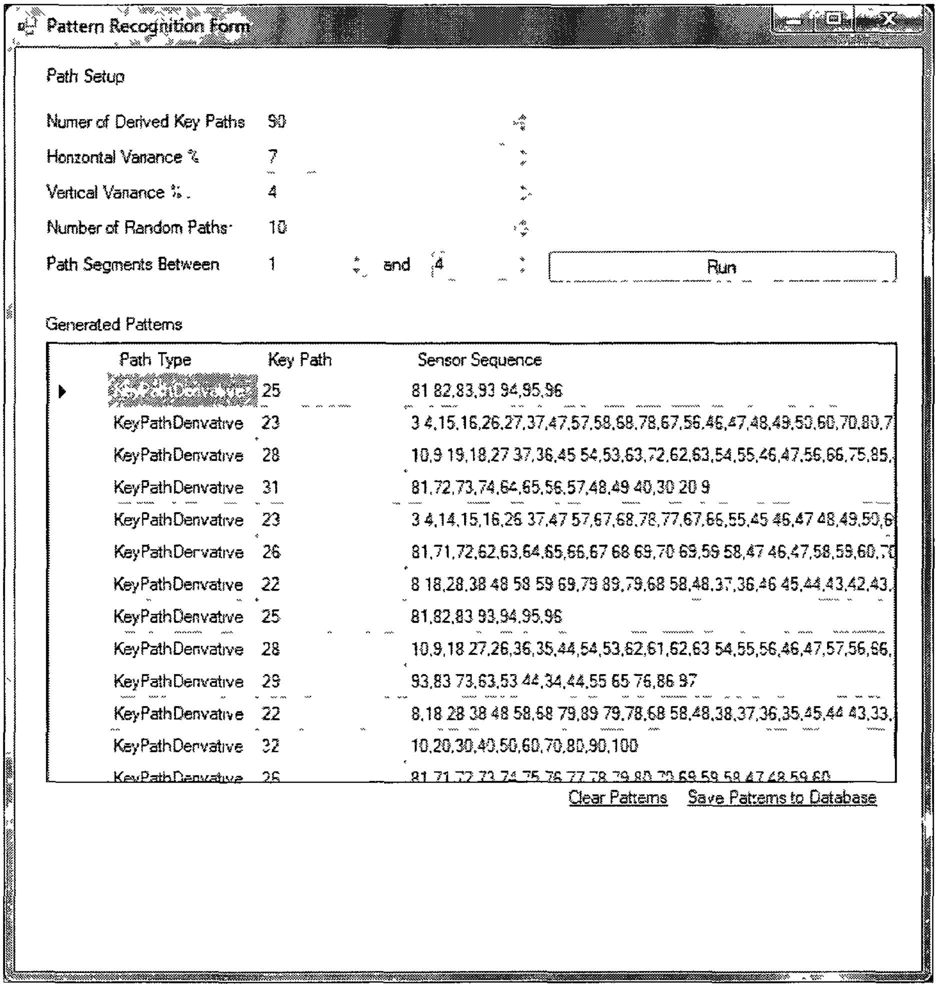

Figure 4.2: Pattern Recognition user interface.

The software simulates object moving along either *key path derivatives* or random paths. Naturally, not the same exact key path will be followed; rather a close path will be taken. We represent this by using horizontal and vertical variances. These values are represented as percentages of the monitored terrain width and height. A number of random paths can also be generated to represent real movement which sometimes does not follow the predefined key paths. The number of paths for each type is specified and the simulation is run. As the objects move through the paths, the sensor sequences that track the objects are recorded. This is shown in the grid in the same figure. For example, the first record shows that an object moving in a path close to key path number 25 was sensed by sensor sequence 81, 82, 83, 93, 94, 95, 96. Every tri-sensor sequence occurrence will be stored into a patterns table as figure 4.3 shows. These records will form the pattern repository from which sequential patterns will be derived.

id	current_sensor_id	source_sensor_id	destination_sen...	occurence
3090	59	60	58	625
3091	58	59	47	313
3092	47	58	46	245
3093	46	47	45	1610
3094	45	46	44	1706
3095	44	45	43	1659
3096	43	44	42	1495
3097	42	43	41	868
3098	41	42	42	983
3099	42	41	53	642
3100	53	42	54	666
3101	54	53	64	408
3102	64	54	65	408
3103	65	64	76	285
3104	76	65	67	87
3105	67	76	57	299
3106	57	67	58	335
3107	58	57	48	418
3108	48	58	38	865
3109	38	48	29	201
3110	29	38	19	192
3111	19	29	20	577
3112	20	19	10	550
3113	10	20	9	649
3114	9	10	8	719
3115	8	9	7	754
3116	7	8	6	762

Figure 4.3: Pattern records in the database.

Using the aggregated patterns and simple mathematics, sequential patterns for each sensor are derived. Each sequential pattern has a confidence which is based on the ratio of the number of occurrence of the pattern to the total number of other related patterns.

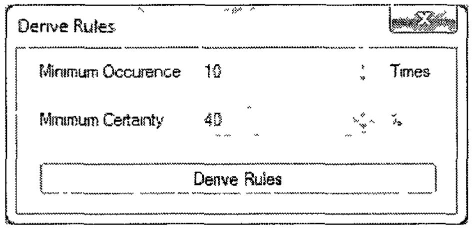

Figure 4.4: Sequential Patterns derivation user interface.

As depicted in figure 4.4, the user can select the minimum confidence for a sequential pattern to be included as well as the minimum number it occurred. Figure 4.5 shows the derived sequential patterns for a couple of sensors.

id	sensor_id	from_sensor_id	to_sensor_id	certainty
5213	7	18	6	1
5214	7	17	6	0.4667
5215	7	16	17	0.5
5216	7	8	6	0.9561
5217	7	6	8	0.619
5218	8	19	7	0.9286
5219	8	18	7	0.4545
5220	8	17	9	0.7143
5221	8	9	7	0.9742
5222	8	7	9	0.5667

Figure 4.5: Sequential Patterns records in the database.

4.2.3 Experiment Configuration.

We have conducted various experiments which dealt with the factors that affect the performance of our tracking technique. Therefore, we had to develop various parameters in our simulation to ensure that we get the required data for our analysis for each impacting factor. Figure 4.6 shows the main window of our simulator which include some parameters that influence the experiments behavior.

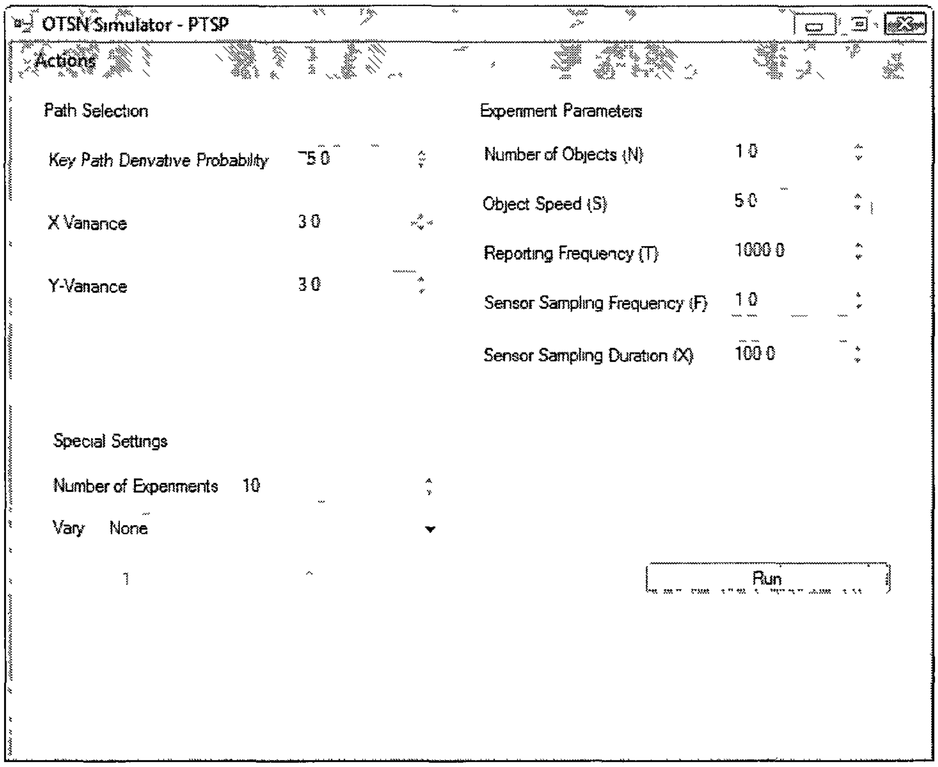

Figure 4.6: Main Window in the simulator which includes tunable parameters.

Path Selection

In this groupbox of parameters we can set the following:

- **Key Path Derivative Probability:** This parameter allows the user to set the percentage of key paths used among all the paths generated during simulation. For example, if we set this parameter to 75% then we will have 75% key paths generated in the experiment and 25% random paths.

- **X and Y Variance:** These parameters are used to create a variance in the way the key path is generated which will differ than how the key path is stored in the database. Thus, creating a slightly altered path than the original key path. The value of X and Y represent a percentage of the change made to the original key path, therefore the higher the value of X and/or Y the higher the changes made to the path from its original coordinates. X will represent the changes in the x-coordinates and Y represents the changes in the y-coordinates.

Experiment Parameters

This groupbox of parameters allow the user to change main factors that impact the performance of the tracking techniques:

- **Number of Monitored Objects:** Represents the network workload and can be used to test how a tracking technique behaves if the number of objects being tracked increases.

- **Object Speed:** This parameter is used to check how the speed of a moving object could impact the tracking process. The unit used is meter per second (m/s).

- **Reporting Frequency:** The amount of time between each report to the base station. This is not a impacting factor to the tracking process although it represents how often the application would receive the updates regarding the movements of objects in the network. The unit of measurement is milliseconds (ms).

- **Sensor Sampling Frequency:** The number of times a sensor samples per reporting instance. This is a crucial factor for the tracking process and increasing this number would result in a better tracking performance with a higher energy consumption, while reducing it will affect the quality of tracking and will reduce the active time of the sensor node and as a result a lower energy consumption. The minimum number of samplings is 1 and the maximum is $\frac{T}{X}$.

- **Sensor Sampling Duration:** The amount of time taken for each sampling which directly affects the tracking process, since if the sampling duration is high, then it's unlikely for the object to leave the sensor's detection area without the sensor knowing about it. However, a high sampling duration means higher active mode time and consequently increases the energy consumption levels. The unit of measurement is milliseconds (ms).

Special Settings

In this groupbox a special settings are available for the user to control the execution of the experiments:

- **Number of Experiments:** This control allows the user to run multiple experiments in an automatic manner without the need to run each experiment individually.
- **Vary:** This setting allows the user to choose the type of experiment the simulator will execute. The following is the available settings:
  - **None:** The experiment(s) will be executed without any changes to the main experiment parameters.
  - **Sampling Duration (X):** This setting will allow the user to set the changes in the sampling duration from a certain value to a certain value By a chosen increments.
  - **Sampling Frequency:** This setting allows the user to determine the changes in sampling frequency during the experiment, by using a minimum value (From:) and a maximum value (To:) and a certain increment (By:).
  - **Object Speed:** Similar to the two previous settings, this setting controls the increase of the object speed during the experiment.

4.3 Performance Evaluation Experiments

We will present a performance evaluation of PTSP compared to the other basic tracking techniques in terms of energy consumption. In addition, we will test PTSP performance against some changes in the impacting factors of OTSNs.

4.3.1 Simulation Settings

Our simulation experiments will be carried out in an OTSN of 100 logical sensor nodes in a 150×150 $meter^2$ monitored region. It's our assumption that each sensor node will have a coverage range of 15m. We have implemented the OTSN network based on the 2D grid topology [67]. The mobility model [35] implemented for the random paths is the Random Waypoint Mobility Model [14] [16], also the default speed is 5 m/s for each experiment, unless noted otherwise. The percentage of key paths to the random paths is set to 75% and 25% respectively. The X-variance (Horizontal-variance) and the Y-variance (Vertical-variance) is set to 5%. The number of path segments is set between 1 and 7 segments randomly chosen by the simulator. The minimum confidence is set to 50% in order to include a larger number of sequential patterns. We also assume that the sampling duration will be 100 ms and the sampling frequency will be more than the reporting frequency. The OTSN will send a report regarding the location of the moving object every 1000 ms to the application. Each simulation experiment will last for 140 seconds which will include more than one object and we have conducted 500 trails (i.e. experiments) in order to get an average value of the results, especially the energy consumption of the network and the missing rate. See table 4.1 for a summary of the simulation settings.

Table 4.1: Simulation Settings

Number of Sensors	100 sensor nodes
Monitored Region	150×150 $meter^2$
Sensor Coverage Range	15 m
Network Topology	2D Grid topology
Default Object Speed	5 m/s
Random Paths (Random Waypoint Mobility Model)	1000
Key Paths Percentage	75%
Random Paths Percentage	25%
X-variance Percentage	5%
Y-variance Percentage	5%
Path Segments	Between 1 and 7
Minimum Confidence	50%
Sampling Duration	100 ms
Time between reports	1000 ms
Simulation Duration	140 seconds
Number of trails	500

As for the energy consumption we have adopted the WINS energy consumption for sensor nodes used in PES [1], see table 4.2. We have only included the MCU and the sensing components energy consumption, since our proposed tracking technique focuses on reducing the energy consumption on those two components. The energy consumption for the radio component will be the subject of our future research.

Table 4.2: Sensor Node Energy consumption. [1]

Component	Mode	Energy Consumption (mW)
MCU	Active Mode	360
MCU	Sleep Mode	0.9
Sensing Component	Active	23

4.3.2 Basic Energy Consumption Analysis

In this experiment we examine the basic energy consumption of PTSP compared to the other tracking schemes, i.e. Naive, SM and CM, (see figure 4.7) where we can find that PTSP has outperformed all the other object tracking techniques. Compared to the

Naive technique all the other tracking techniques have shown very high energy savings that reached more than 90% of the energy consumption by the Naive technique The second best tracking technique was the CM, since it only needs one sensor node at a time to track a moving object unlike the SM which uses all its sensor nodes to locate the object In figure 4 8 we can see the same chart except that we have used a logarithmical scale to represent the energy consumption

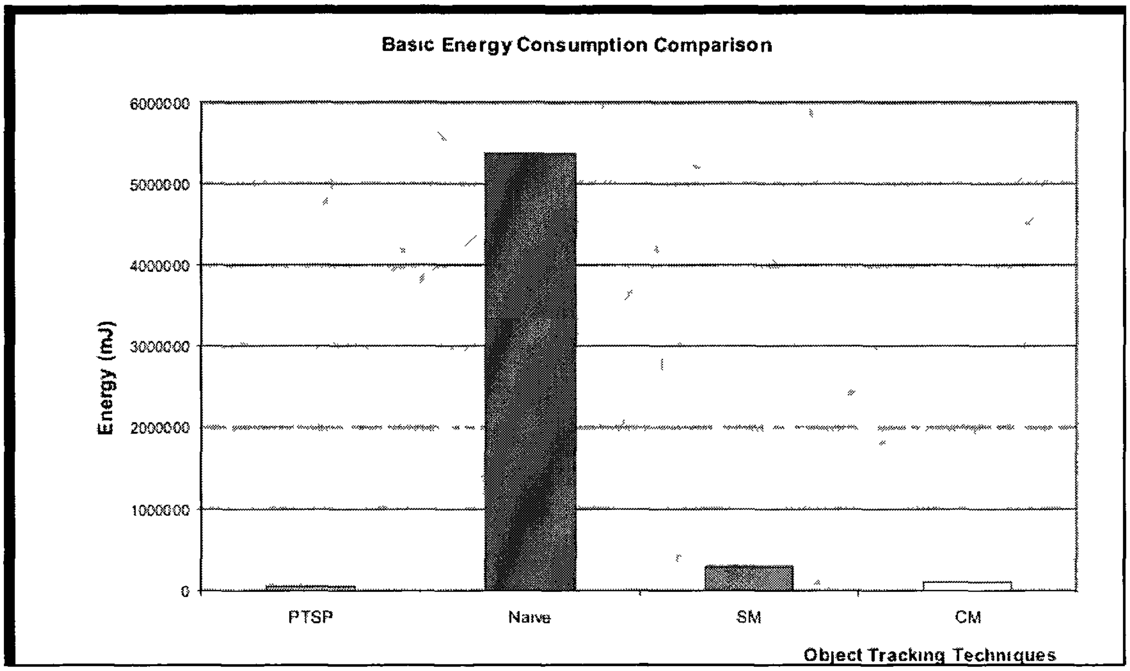

Figure 4 7 Basic energy consumption of each tracking techniques

4.3.3 Network Workload Analysis

In this experiment we have tested and compared PTSP to the other basic tracking schemes, namely the SM and CM, in regard to the network workload which is represented by the number of moving objects in the network The experiments were conducted by increasing

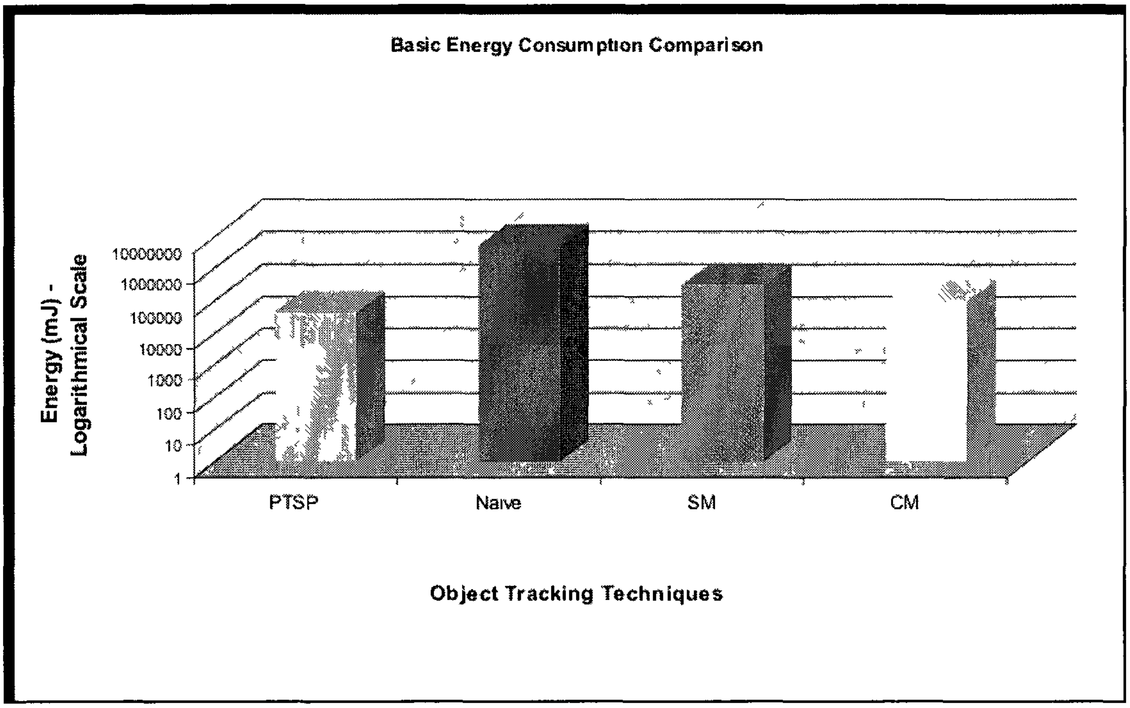

Figure 4.8: Basic energy consumption of each tracking techniques using a logarithmical scale for the energy consumption.

the number of moving objects from 1 to 10 objects. Since as expected the naive technique was not affected by the increase of the network workload we have removed its data from the chart in figure 4.9 since it had an enormously huge energy consumption level compared to the other schemes.

Moreover, we don't expect that PTSP to be the best tracking technique for all conditions. Therefore, we can see that the three techniques (PTSP, CM and SM) had maintained a very low level of energy consumption. These energy savings were accomplished by keeping the sensor nodes in the network in sleep mode as long as possible compared to the naive technique which keeps all the sensor nodes in active mode all the time, thus consuming huge amounts of energy in a futile manner. In addition, although the SM technique is shown in figure 4 9 consuming more energy compared to CM and PTSP while the number of moving objects is small.

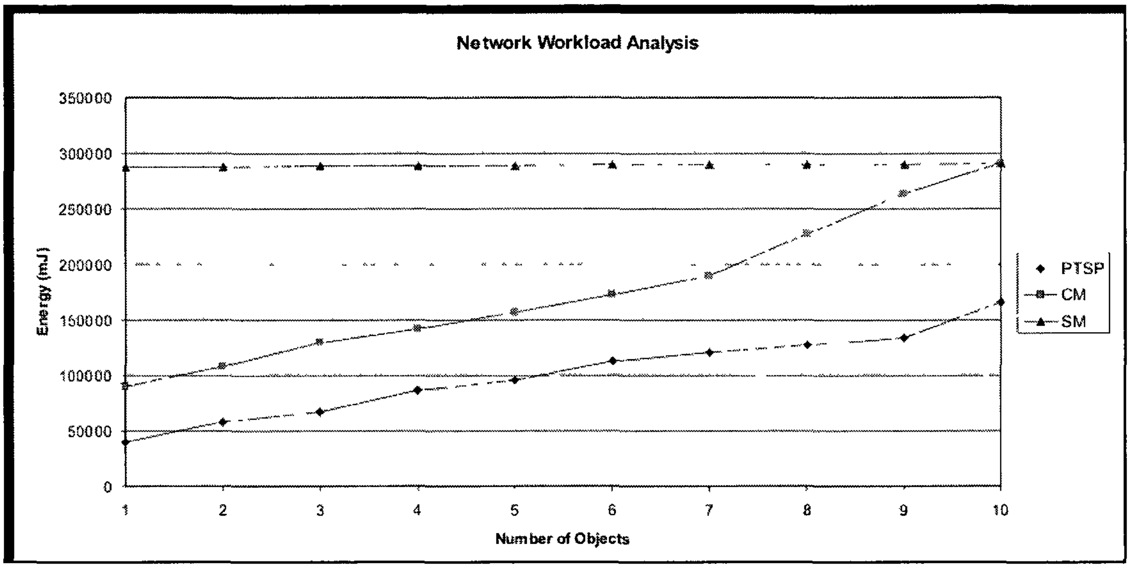

Figure 4.9: Network workload analysis.

Conversely, we can notice a dramatic reduction in the energy consumption gap between SM and both CM and PTSP when the number of objects increases to the point where CM matches SM in regard to the energy consumed when the object count reaches 10 objects. These results were due to the fact that SM is always activating all its sensor nodes regardless of the number of objects and the only increase of energy would be attributed to the extra reports sent to the base station. As a result, if the network has a large number of moving objects then the most ideal scheme would be SM. In the case of CM and PTSP the increase in the number of objects means an increase in the number of active sensor nodes and thus higher energy consumption levels. Also, it can be observed that the increase in the energy consumption was following a linear incremental pattern in regard to the increase of the number of objects. Therefore, for the following simulation experiments we will be using only one moving object to simplify the calculations.

As for the missing rate analysis we will only consider PTSP, since for the other basic tracking techniques (Naive, SM and CM) the missing rate is always 0%. We can notice in figure 4.10 that the missing rate levels is not impacted by increase in the number of moving objects as we have expected, since the missing rate is the ratio of the missing reports to the total number of reports, therefore this ratio is not affected by the increase in the number of objects since even though it's likely that the number of missing reports will increase, this number will be matched with an increase in the number of total reports, thus the ratio remains unchanged.

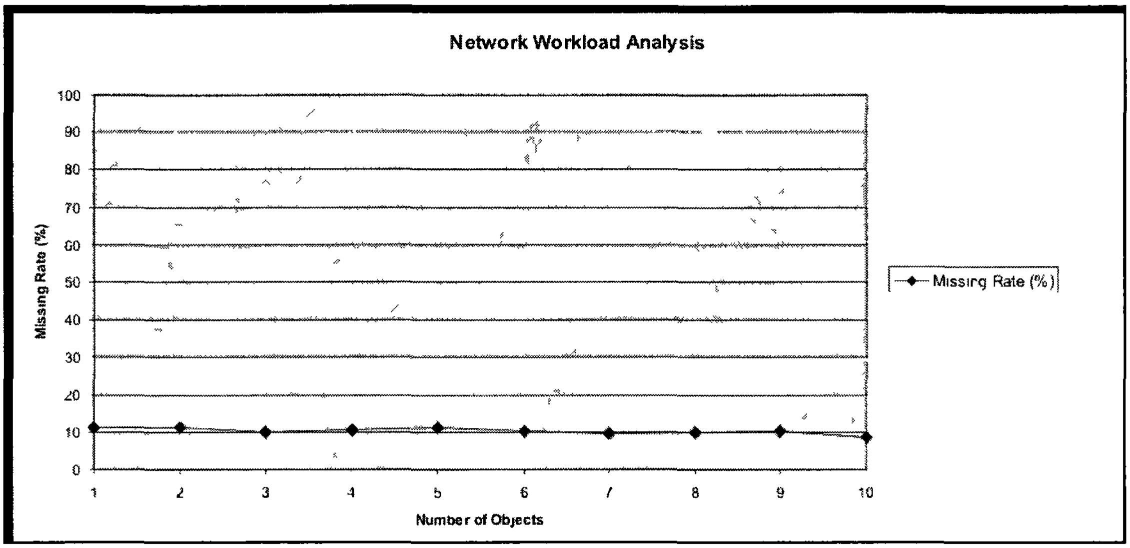

Figure 4.10: Missing rate levels against the network workload.

4.3.4 Object Movement Speed Analysis

This experiment evaluates how the changes in the speed of a moving object could affect the energy consumed by a tracking technique. Also, since the CM is better than the SM when there is a low number of objects in the network then we will only use the CM for comparison purposes against PTSP. As shown in figure 4.11 we can notice a linear growth in the energy consumption levels when the object speed also increases, which is resulted by the fact that when the object moves in a faster speed the prediction for the destination sensor will be harder and thus more recovery process is required and eventually an increase in the overall energy consumption of the network. We can notice that PTSP was outperforming CM when the object speed was below 30 m/s which is considered an excellent performance compared to the speed of the object, since tracking an object moving in a speed of 25 m/s for example is not an easy task, if energy savings is also a factor of the tracking technique.

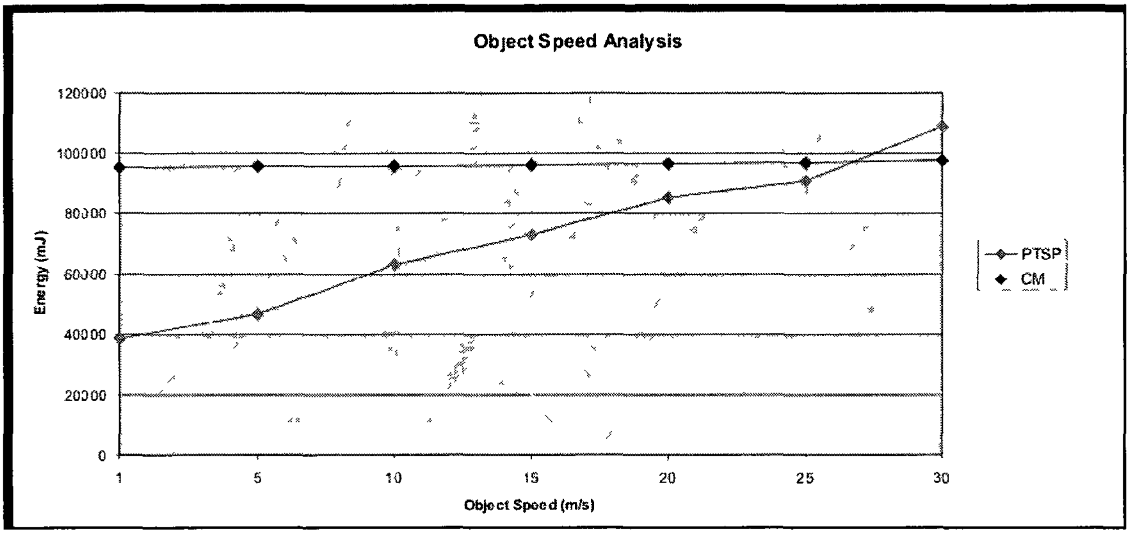

Figure 4.11: Object movement speed energy consumption analysis.

We can also notice that regarding the missing rate levels, PTSP has kept an acceptable level of missing rate, although it was increasing along with the increases in the object speed, see figure 4.12.

4.3.5 Sampling Duration Analysis

The sampling duration represent the time at which the sensor node is in active mode, since both the MCU and the sensing components are active and therefore this factor influences the energy consumption in our tracking technique. Since if the sampling duration is increased then as a result it increases the total time the sensor nodes are kept in active mode, which consequently results in an increase in the energy consumed by the network. Figure 4.13 shows that when the sampling duration is increased then it directly affects the energy consumption of PTSP, however it doesn't have any impact on CM, since CM doesn't use a sampling duration and instead

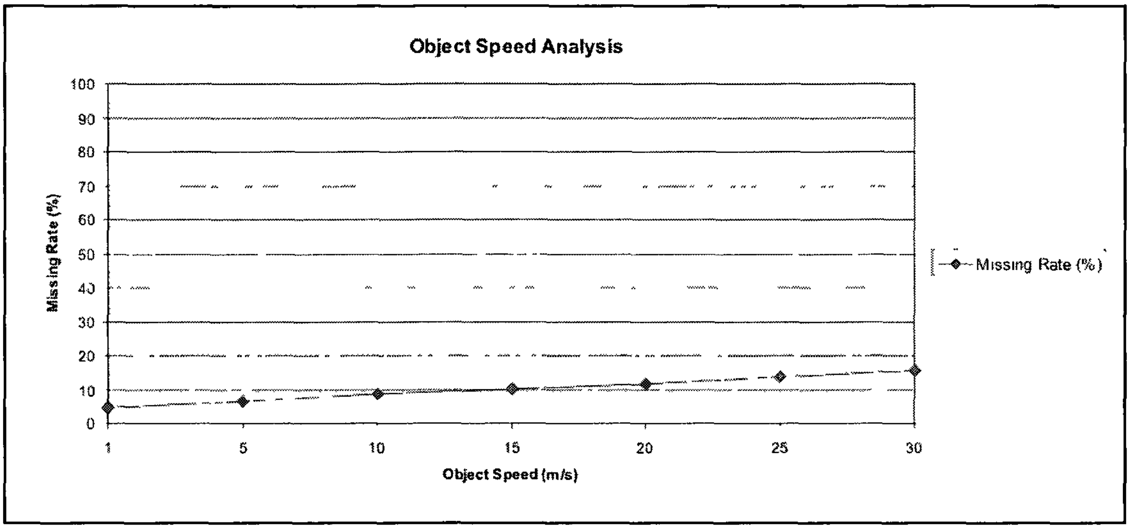

Figure 4.12: Object movement speed and missing rate analysis.

requires the sensor node which has detected an object to keep monitoring that object as long as the object is still inside its detection area, and to activate the destination sensor when the object is about the leave the current sensor detection area. Thus, CM doesn't depend on the sampling duration while monitoring moving objects. As for PTSP the increases in the sampling duration will increase the energy consumed by the network and also it will reduce the number of missing objects, since the current sensor is kept active longer than it used to be which means that there will be more assurance regarding the prediction of the future movement of the object. In addition, figure 4.14 shows that the missing rate will decrease linearly to become below 3% while the sampling duration increases, since the increase in the sampling duration results in lower number of missing reports and thus a lower missing rate.

4.3.6 Sampling Frequency Analysis

One of the most critical influencing factors is the sampling frequency, since this factor could affect the performance of tracking technique directly in terms of the quality of tracking and the energy consumption. We can define the sampling frequency as the number of samplings performed during a reporting period, which is

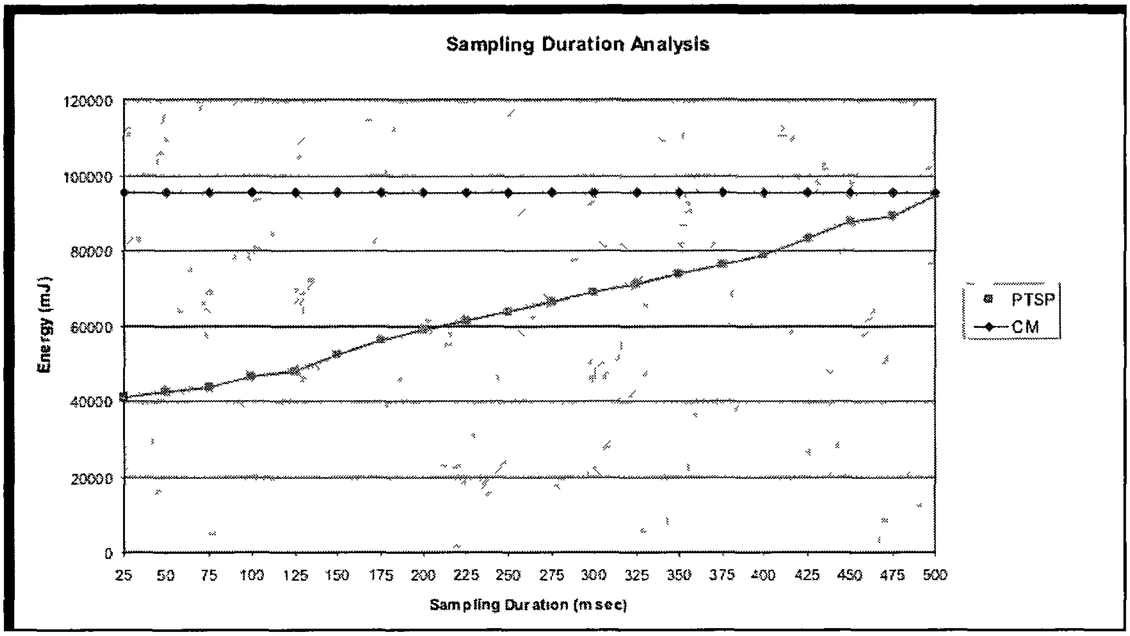

Figure 4.13: Sampling duration and energy consumption analysis.

the time between reports to the base station. It's imperative for the tracking technique to execute at least one sampling per a reporting period, so that the application could receive an updated information regarding the location of the moving object(s). When the sampling frequency is increased then this would mean that the sensor node will have a better information regarding the location of the moving object and as a result will be able to accurately predict it next movement, however these increase will incur a higher energy consumption levels, since the sensor nodes will be kept active for longer time. Therefore, similar to the sampling duration the sampling frequency represent bargain between more accurate predictions or less energy consumption. So, increasing the sampling frequency may result in better predictions, however it will also mean that we are keeping the sensor node active for longer periods which is mainly unneeded since there are no reports to be sent. Figure 4.15 shows that similar to the sampling duration there are a linear relation between the increments of the sampling

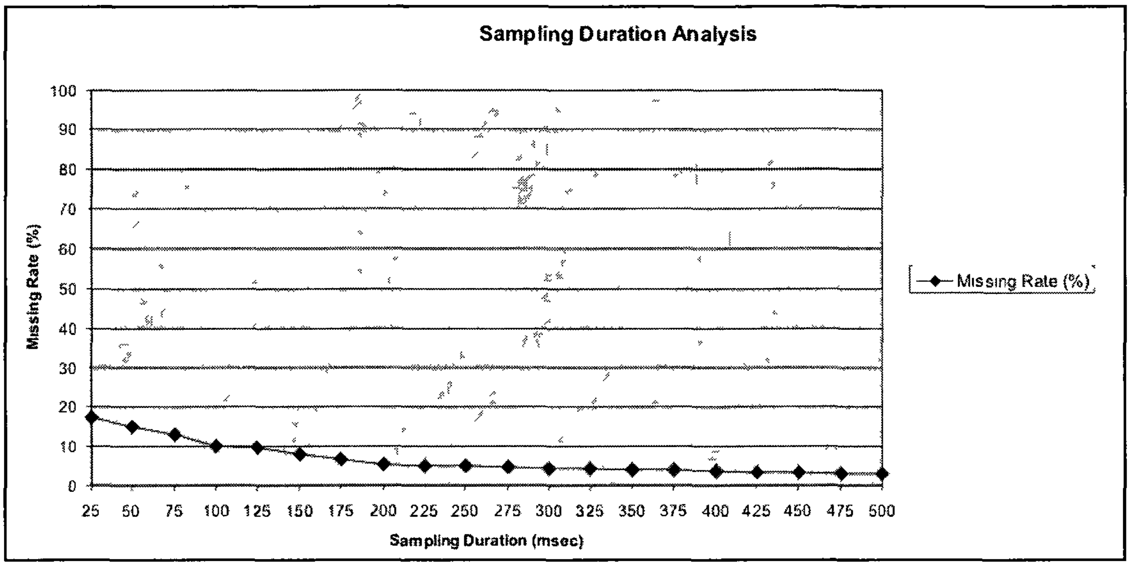

Figure 4.14: Sampling duration and missing rate analysis.

duration and the increase in the energy consumed by PTSP. We can also notice that CM is not affected by the increase in the sampling frequency since it doesn't employ this factor in its tracking scheme. In regard to the missing rate while increasing the sampling frequency we can observe that there has been a sharp decrease in the missing rate, see figure 4.16. PTSP missing rate has reached below 1% when the sampling frequency became 5 On the other hand, if we increase the number of samplings we will not achieve an optimal performance since any advantages made by having enhanced predictions will be negated by the cost added due to the increase in the energy consumption of the network. Therefore, it's recommended to tune the sampling frequency in a way that results in a better energy savings while maintaining low missing rate.

4.3.7 Recovery Mechanism Analysis

In this experiment we have evaluated the three previously mentioned recovery mechanisms in terms of energy consumption. We can notice in figure 4.17 that the first recovery mechanism is the best in terms of energy consumption which is based on activating the neighboring sensor nodes of the current sensor.

The next best recovery mechanism is the third one which combines the first two recovery mechanisms by activating all the neighboring

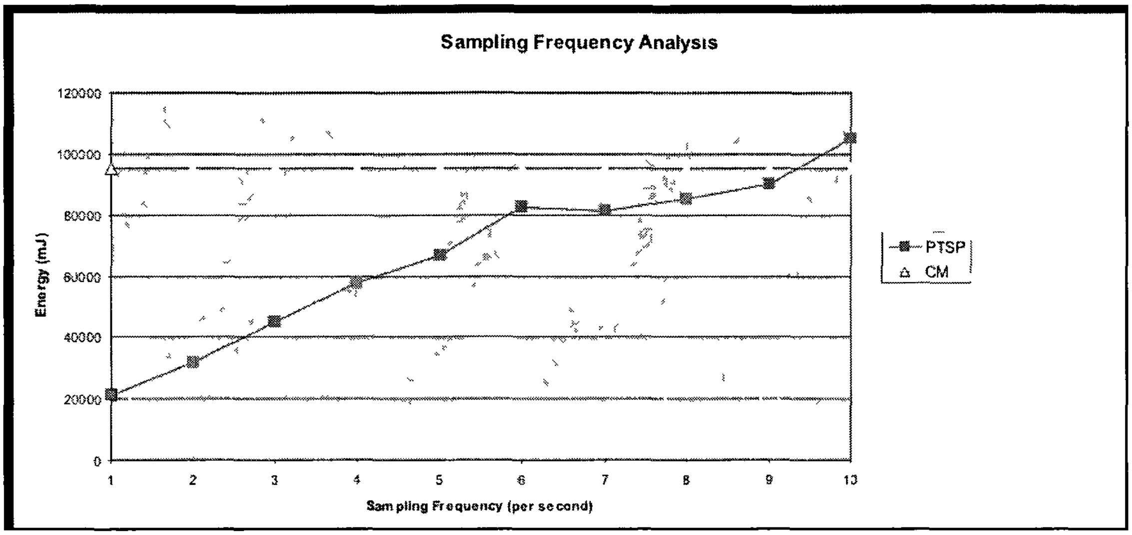

Figure 4.15: Sampling frequency and energy consumption analysis.

sensor nodes of both the current sensor and destination sensor, and we call it here the All Neighbors recovery mechanism. As for the worst recovery mechanism in terms of energy consumption, it was the second recovery mechanism called the destination recovery mechanism. The reason this recovery mechanism became the worst is due to the fact that this recovery mechanism required the network to go for the second phase of recovery more often compared to the other recovery mechanisms. Since the second phase of recovery involves activating all the sensor nodes in the network, then it incurs more energy consumption compared to the first phase of recovery. Therefore, we have chosen the first recovery mechanism (Source Recovery) as our recovery mechanism of choice in all the previous experiments, since it was the most energy conservative recovery mechanism.

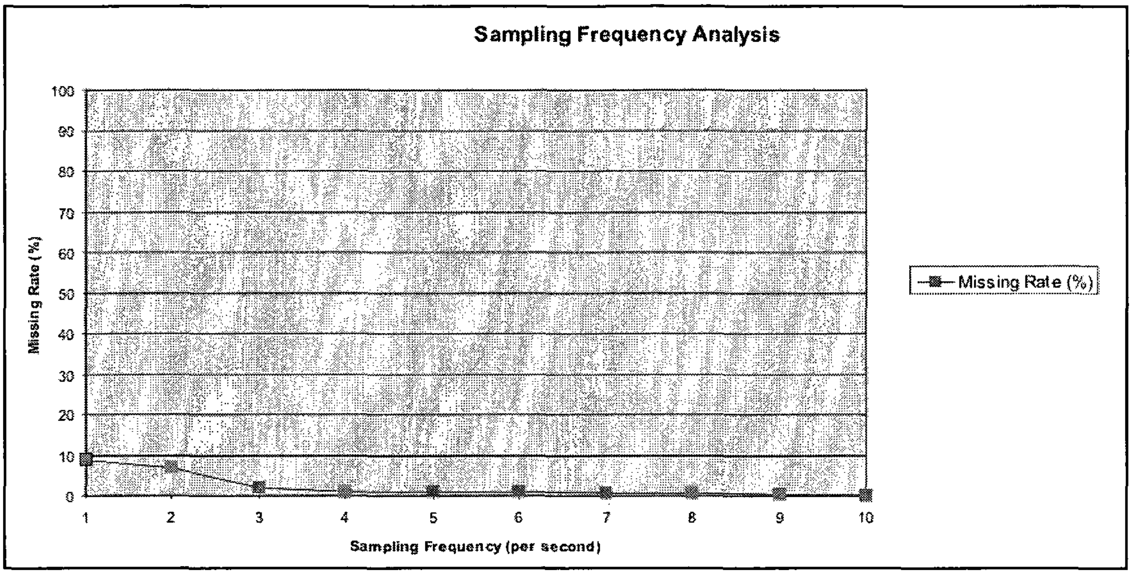

Figure 4.16: Sampling frequency and missing rate analysis.

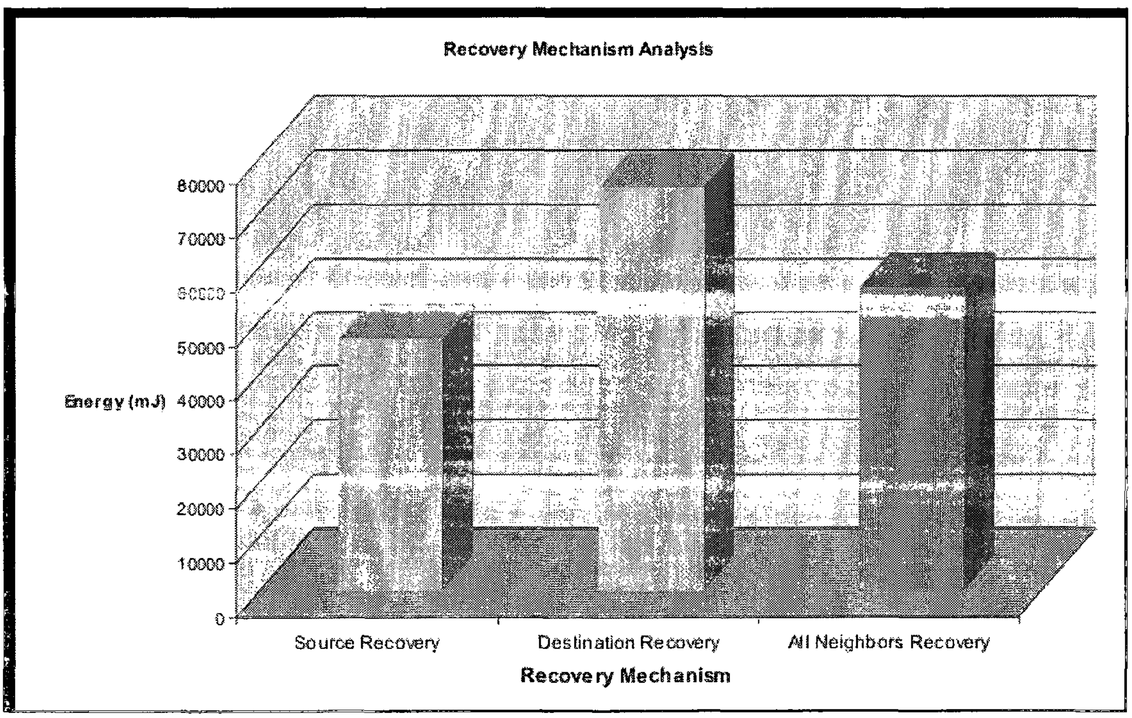

Figure 4.17: Recovery mechanisms and energy consumption analysis.

Chapter 5

Conclusion and Future Work

In this chapter, we will summarize the contributions made toward an effective and energy efficient prediction-based tracking technique which uses sensor sequential patterns as the basis of its prediction mechanism (PTSP). Based on many research projects in the object tracking sensor networks, we can be certain that there will always be a need for an energy efficient tracking scheme yet provides a high quality tracking data to its application. In the following sections we will begin summarizing our contributions and later on we will discuss the type of future work that could enhance our work and provide a better understanding to the complexities of developing an objects tracking technique.

5.1 Summary of Contributions

In this thesis, we have proposed an energy efficient prediction-based tracking technique which implements a novel approach in its prediction mechanism.

One of the key applications of the sensor networks which is widely adapted due to its huge number of implementations and usages, is the object tracking sensor network. Although, this application is known for its high demands of energy in order to perform its tasks in the best manner possible. Since the main obstacle that faces most tracking techniques in sensor networks is the fact that the sensor network suffers from a very limited power supply which in turn restricts the type of tasks to be accomplished by the sensor network. Therefore, the need to optimize the energy consumption in OTSNs is a fact the must be faced. Since most the energy savings research was focusing on minimizing the energy consumed by the radio component (RF radio) in the sensor nodes by reducing the number of messages transmitted and received, we are considering the energy consumed by the MCU and the sensing components in the sensor nodes which also attributed to a respectful amount of energy consumption. Therefore, we have developed an object tracking scheme that minimizes the time the sensor node stays in active mode (when both the MCU and the sensing component is active), thus generating a considerable amount of energy savings. In this thesis, our prediction based tracking technique (PTSP) utilizes the sensor sequential patterns to produce accurate predictions of the future movements of a certain object. These sequential patterns are continuously evaluated and updated in order to provide the prediction mechanism with the latest and most accurate predictions. In addition, the activation mechanism applies 3 different activation mechanism to assist in the tracking of the moving objects in which those mechanisms represent a tradeoff between the energy savings and the reduction of the missing objects. Moreover, we have introduced 3 different recovery mechanisms in regard to the neighboring sensor nodes to be activated. We have simulated our proposed tracking scheme (PTSP) along with 3 basic tracking schemes for comparison purposes. The results generated by the experiments were mainly testing the performance of PTSP and the other tracking schemes against two main metrics: (1) total energy consumed by the network during the simulation period, including the active and sleep mode energy consumption for each sensor node in the network, (2) missing rate, which represents a ratio of the missing reports to the total number of reports received by the application. Moreover, it was proven by the simulation results that PTSP outperformed all the other tracking schemes by keeping a low energy consumption levels while maintaining an acceptable level of the missing rate.

5.2 Future Work

In our developed tracking scheme we have studied the energy consumed by the MCU and the sensing components. Since we believe that the energy consumption contributed by the radio component should have an orthogonal affect to our simulations, then we propose for our future work to study the energy consumed by the radio component. Also, we will evaluate the impact of the 2 activation mechanisms which we didn't implement, namely (Current sensor-Path sensor(s)-Destination sensor) mechanism and (Current sensor-Neighboring sensors-Destination sensor). We will also investigate the possibility to produce more accurate sensor sequential patterns which also could be more adaptive to the changes made to the network. In addition, we will study the effects of the moving objects' direction and speed in producing more appropriate predictions. Finally, a fault-tolerance approach will be evaluated and studied in order to mitigate the effects of nodes failure while tracking a moving object.

Bibliography

[1] Y. Xu, J.Winter, and W.-C. Lee. "Prediction-based strategies for energy saving in object tracking sensor networks". In Proceedings of the International Conference on Mobile Data Management, pages 346-357, Berkeley, CA, Jan. 2004.

[2] H. Yang and B. Sikdar, "A Protocol for Tracking Mobile Targets Using Sensor Networks", Sensor Network Protocols and Applications, in Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, Alaska, May 2003, pp. 71-81.

[3] Akyildiz, I.F., Su,W., Sankarasubramaniam, Y., and Cayirci, E. "Wireless sensor networks: a survey". Computer Networks, vol 38(4), pp 393-422, March 2002.

[4] M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hmlinen, M. Hnnikinen, and T. D. Hmlinen, Ultra-Low Energy Wireless Sensor Networks in Practice, John Wiley & Sons, England, 2007.

[5] Karl H and Willig A. Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons Ltd, Chichester. 2005.

[6] Vincent S. Tseng and Kawuu W. Lin, "Energy Efficient Strategies for Object Tracking in Sensor Networks: A Data Mining Approach", Journal of Systems and Software, Vol. 80, No. 10, pp. 1678-1698, October 2007.

[7] Y. Xu, J. Winter and W.-C. Lee, "Dual Prediction-based Reporting for Object Tracking Sensor Networks", In Proceedings of the First Annual International Con-

ference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), 2004, pp. 154-163.

[8] S.-M. Lee, H. Cha and R. Ha, "Energy-Aware Location Error Handling for Object Tracking Applications in Wireless Sensor Networks", Computer Communication 30, 2007, pp. 1443-1450.

[9] Y. Xu and W.-C. Lee, "On Localized Prediction for Power Efficient Object Tracking in Sensor Networks", In Proceedings of the 1st International Workshop on Mobile Distributed computing, Providence RI, May 2003, pp. 434-439.

[10] M. Ilyas and I. Mahgoub. Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems. CRC Press, Boca Raton, 2004.

[11] Haenselmann, T. (2006-04-05). Sensornetworks. GFDL Wireless Sensor Network textbook.

[12] D. Culler, D. Estrin, and M. Srivastava. "Overview of sensor networks". Special Issue in Sensor Networks, IEEE Computer, 37(8):4149, 2004.

[13] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. "Energy aware wireless microsensor networks". IEEE Signal Processing Magazine, Vol. 19, Issue 2:4050, March 2002.

[14] F. Bai and A. Helmy, "A SURVEY OF MOBILITY MODELS". in Wireless Adhoc Networks. University of Southern California,U.S.A.: University of Southern California,U.S.A., 2004.

[15] V. Borrel, M. D. de Amorim, and S. Fdida. "On natural mobility models". In LNCS, International Workshop on Autonomic Communication (WAC), Athens, Greece, October 2005. Pages:243-253.

[16] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", in Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483-502, 2002.

[17] Chong CY and Kumar SP. "Sensor networks: Evolution, opportunities, and challenges". *Proceedings of the IEEE* 91(8), 12471256. 2003

[18] Masseglia, F., Teisseire, M., and Poncelet, P., "Sequential Pattern Mining: A Survey on Issues and Approaches", in Encyclopedia of Data Warehousing and Mining, nformation Science Publishing, Oxford University Press, pages:3-29, 2005.

[19] Agrawal, R., & Srikant, R. (1995). "Mining Sequential Patterns". In Proceedings of the 11th International Conference on Data Engineering, Taipei, Taiwan, 3-14.

[20] Michael W. Berry, Murray Browne (2006), Lecture notes in Data Mining, World Scientific Publishing.

[21] Daniel Larose, Discovering Knowledge in Data: An Introduction to Data Mining, Wiley, Hoboken, NJ, 2005.

[22] Bramer, M., 2007. Principles of Data Mining, Springer-Verlag, New York Inc.

[23] Kim, H., Kim, E., Han, K.: "An Energy Efficient Tracking Method in Wireless Sensor Networks". In: Koucheryavy, Y., Harju, J., Iversen, V.B. (eds.) NEW2AN 2006. LNCS, vol. 4003, pp. 278286. Springer, Heidelberg (2006)

[24] Cook, D. J. and Das, S. K. Smart Environments: Technologies, Protocols and Applications. Wiley-Interscience, 2005.

[25] Yang, T. A., Jain, D., and Sun, B. "Development of Emulation-Based Projects for Teaching Sensor Networks". J. Comput. Small Coll. 24, 2 (Dec. 2008), 64-71. ACM Press, New York, NY, USA.

[26] Boukerche, A., and Samarah, S. "A Performance Evaluation of Distributed Framwork for Mining Wireless Sensor Netwroks". Simulation Symposium, Annual, pp. 239-246, 40th Annual Simulation Symposium (ANSS'07), 2007

[27] Boukerche, A. and Samarah, S. "A Novel Algorithm for Mining Association Rules in Wireless Ad-hoc Sensor Networks". IEEE Trans. Parallel Distrib. Syst. 19, 7 (Jul. 2008), 865-877.

[28] Boukerche A., Samarah S., "A Novel data Mining Technique for extracting Events and Inter Knowledge based Information from Wireless Sensor Networks", Local Computer Networks, Annual IEEE Conference on, pp. 769-775, Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 2006

[29] Boukerche A., Samarah S., "An Efficient Data Extraction Mechanism for Mining Association Rules from Wireless Sensor Networks", Communications, 2007. ICC '07. IEEE International Conference on , vol., no., pp.3936-3941, 24-28 June 2007

[30] Bhatti, S. and Xu, J. 2009. "Survey of Target Tracking Protocols Using Wireless Sensor Network". In Proceedings of the 2009 Fifth international Conference on Wireless and Mobile Communications (August 23 - 29, 2009). ICWMC. IEEE Computer Society, Washington, DC, 110-115.

[31] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal, "Wireless sensor network survey", Computer Networks, Volume 52, Issue 12, 22 August 2008, Pages 2292-2330, ISSN 1389-1286

[32] Wireless sensor network, http://en.wikipedia.org/wiki/Wireless_sensor_network

[33] Sensor node, http://en.wikipedia.org/wiki/Sensor_node

[34] Association rule learning, http://en.wikipedia.org/wiki/Association_rule_learning

[35] Mobility model, http://en.wikipedia.org/wiki/Mobility_model

[36] Al-Ghanem, W. (2008). "Energy Efficient Cluster-Based Target Tracking Strategy". M.S. Thesis. Department of Computer Science and Engineering. Florida Atlantic University, Boca Raton, Florida. June 2008.

[37] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, and T. Tuan. "Picoradios for wireless sensor networks: the next challenge in ultra-low-power design". In Proceedings of the Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International , vol.1, no., pp.200-201 vol.1, 2002

[38] J. M. Rabaey, M. J. Ammer, J. L. da Silva Jr., D. Patel, and S. Roundy. "Picoradio supports ad hoc ultra-low power wireless networking". Computer , vol.33, no.7, pp.42-48, Jul 2000

[39] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan; "Energy-Efficient Communication Protocol for Wireless Microsensor Networks". System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on , vol., no., pp. 10 pp. vol.2, 4-7 Jan. 2000

[40] Manjeshwar, A.; Agrawal, D.P.;, "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks". Parallel and Distributed Processing Symposium., Proceedings 15th International ,pp.2009-2015, Apr 2001

[41] Manjeshwar, A.; Agrawal, D.P.; , "APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM , vol., no., pp.195-202, 2002

[42] Balasubramanian, S.; Elangovan, I.; Jayaweera, S.K.; Namuduri, K.R.; "Distributed and collaborative tracking for energy-constrained ad-hoc wireless sensor networks". Proceedings of the Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE , vol.3, no., pp. 1732- 1737 Vol.3, 21-25 March 2004

[43] Guang-yao Jin, Xiao-yi Lu and Myong-Soon Park. Dynamic Clustering for Object Tracking in Wireless Sensor Networks. UCS 2006, LNCS 4239, pp. 200 209, 2006.

[44] Q. Huang, C. Lu and G.-C. Roman, "Spatiotemporal Multicast in Sensor Networks", Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys), November 2003, pp. 205-217.

[45] C.-T. Lee, F. Yeong-Sung Lin and Y.-F. Wen, "An Efficient Object Tracking Algorithm in Wireless Sensor Networks", in JCIS-2006 Proceedings Advances in Intelligent Systems Research, October 2006.

[46] M. Walchli, P. Skoczylas, M. Meer and T. Braun, "Distributed Event Localization and Tracking with Wireless Sensors", in Proceedings of the 5th international Conference on Wired/Wireless internet Communications, May 23 - 25, 2007, Lecture Notes In Computer Science, vol. 4517. Springer-Verlag, Berlin, Heidelberg, 247-258.

[47] M. Abdelkader, M. Hamdi, and N. Boudriga, "Voronoi-Based Sensor Network Engineering for Target Tracking Using Wireless Sensor Networks", New Technologies, Mobility and Security, 2008. NTMS '08. , vol., no., pp.1-5, 5-7 Nov. 2008

[48] S. C. Chuang, "Survey on Target Tracking in Wireless Sensor Networks." Dept. of Computer Science - National Tsing Hua University, (5/26/2005).

[49] Y.-S. Chen and S.-Y. Ann, "VE-Mobicast: A Variant-Egg-Based Mobicast Routing Protocol in Wireless Sensor Networks", in Proceedings of the 40th IEEE International Conference on Communications (IEEE ICC2005), May 2005, vol. 5, Seoul, Korea, pp. 3020-3024.

[50] Y.-S. Chen and Y.-J. Liao, "HVE-mobicast: A Hierarchical-Variant-Egg Based Mobicast Routing Protocol for Wireless Sensornets", in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC2006), April 2006, vol. 2, Las Vegas, NV, USA, pp. 697-702.

[51] H.-W. Tsai, C.-P. Chu and T.-S. Chen, "Mobile Object Tracking In Wireless Sensor Networks", Computer Communications 30, 2007, pp. 1811- 1825.

[52] H.T. Kung and D. Vlah, "Efficient Location Tracking Using Sensor Networks", Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE , vol.3, no., pp.1954-1961 vol.3, 20-20 March 2003

[53] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, "Efficient In-Network Moving Object Tracking in Wireless Sensor Networks", Department of Computer Science and Information Engineering - National Chiao Tung University, 2004.

[54] W.-P. Chen, J.C. Hou and L. Sha, "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks", in Proceeding of 11th IEEE International Conference on Network Protocols (ICNP'03), Atlanta, Georgia, USA, pp. 284-294, November 2003.

[55] Y. Zou and K. Chakrabarty, "Target Localization Based on Energy Considerations in Distributed Sensor Networks", in Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, Alaska, pp. 51-58, May 2003.

[56] E. Olule, G. Wang, M. Guo and M. Dong, "RARE: An Energy Efficient Target Tracking Protocol for Wireless Sensor Networks", Parallel Processing Workshops, 2007. ICPPW 2007. International Conference on , vol., no., pp.76-76, 10-14 Sept. 2007

[57] Z. Wang, H. Li, X. Shen, X. Sun and Z. Wang, "Tracking and Predicting Moving Targets in Hierarchical Sensor Networks", IEEE International Conference on Networking, Sensing and Control, 2008, 1169-1173, ICNSC 2008.

[58] G.-y. Jin, X.-yi Lu, and M.-Soon Park, "Dynamic clustering for object tracking in wireless sensor networks", in Ubiquitous Computing Systems, 2006, pp.200-209.

[59] W.-R. Chang; H.-T. Lin and Zong-Zhi Cheng, "CODA: A Continuous Object Detection and Tracking Algorithm for Wireless Ad Hoc Sensor Networks", Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE, pp.168-174, 10-12 Jan. 2008.

[60] S. Pino-Povedano and F.-J. Gonzalez Serrano, "Distributed tracking and classification of targets with sensor networks", Software, Telecommunications and Computer Networks, 2008. SoftCOM 2008. 16th International Conference, pp.213-217, 25-27 Sept. 2008.

[61] C.-Y. Chong, F. Zhao, S. Mori and S. Kumar, "Distributed Tracking in Wireless Ad Hoc Sensor Networks", in Proceedings of the Sixth International Conference on Information Fusion (FUSION 2003), Cairns, Australia, pp. 431-438, July 2003.

[62] S. P. M. Tran and T. A. Yang, "OCO: Optimized Communication and Organization for Target Tracking in Wireless Sensor Networks", Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06), pp. 428-435, 2006.

[63] F. Zhao, J. Shin and J. Reich, "Information-Driven Dynamic Sensor Collaboration for Tracking Applications", IEEE Signal Processing Magazine, 19 (2), pp.61-72, 2002.

[64] R. Shorey, A. L. Ananda, M. C. Chan, and W. T. Ooi, "Mobile, wireless, and sensor networks: technology, applications, and future directions", Published [Piscataway]: IEEE Press; Hoboken, N.J., Wiley-Interscience, pp173-196, c2006.

[65] A. Rapaka and S. Madria, "Two Energy Efficient Algorithms For Tracking Objects In A Sensor Network", Wireless Communication Mobile Computing, 7:pp.809-819, 2007.

[66] Microsoft Corporation, C#, http://msdn.microsoft.com/en-us/vcsharp/aa336809.aspx

[67] Hock Guan Goh, Moh Lim Sim and Hong Tat Ewe, "Energy Efficient Routing for Wireless Sensor Networks with Grid Topology", Embedded and Ubiquitous Computing, in IFIP International Federation for Information Processing, pp. 834843, 2006.

[68] R. AGRAWAL, T. IMIELINSKI, AND A. SWAMI, "Mining association rules between sets of items in large databases", In Proceedings of the 1993 ACM SIGMOD international Conference on Management of Data (Washington, D.C., United States, May 25 - 28, 1993). P. Buneman and S. Jajodia, Eds. SIGMOD '93. ACM, New York, NY, 207-216.

[69] Piatetsky-Shapiro, G. (1991), Discovery, analysis, and presentation of strong rules, in G. Piatetsky-Shapiro & W. J. Frawley, eds, Knowledge Discovery in Databases, AAAI/MIT Press, Cambridge, MA.

[70] Jochen Hipp, Ulrich Gntzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule mining - A general survey and comparison. SIGKDD Explor. Newsl. 2, 1 (Jun. 2000), 58-64.

[71] Agrawal, R. and Srikant, R. 1994. "Fast algorithms for mining association rules". In Proc. 1994 Int. Conf. Very Large Data Bases (VLDB94), Santiago, Chile, pp. 487499.

[72] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I. 1996. Fast discovery of association rules. In Advances in Knowledge Discovery and Data Mining, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.), AAAI/MIT Press, pp. 307328.

[73] R. Srikant and R. Agrawal. "Mining sequential patterns: Generalizations and performance improvements". In Proceedings of the 5th international Conference on Extending Database Technology: Advances in Database Technology (March 25 - 29, 1996). P. M. Apers, M. Bouzeghoub, and G. Gardarin, Eds. Extending Database Technology, vol. 1057. Springer-Verlag, London, 3-17.

[74] Samarah, S. and Boukerche, A. "Chronological Tree- A Compressed Structure for Mining Behavioral Patterns in Wireless Sensor Networks". Journal of Interconnection Networks 9(3): 255-276 (2008).

[75] S. Samarah, M. Al-Hajri, A. Boukerche, "Object Tracking Sensor Networks: A Survey". To be submitted.

[76] S. Samarah, M. Al-Hajri, A. Boukerche, "An Energy Efficient Prediction-based Technique for Tracking Moving Objects in Wireless Sensor Networks". To be submitted.

[77] M. Al-Hajri, S. Samarah, A. Boukerche, "Recovery Techniques for Tracking Missing Objects in Wireless Sensor Networks". To be submitted.

[78] S. Samarah, M. Al-Hajri, A. Boukerche, "A Predictive Energy Efficient Technique to Support Object Tracking Sensor Networks". Accepted by the *IEEE Transactions on Vehicular Technology 2010.*