An Energy-Efficient Target Tracking Protocol Using Wireless Sensor Networks

by

Adel Mohammad Shafiei

Supervisor: Prof. Azzedine Boukerche

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

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Target tracking using Wireless Sensor Networks (WSNs) has drawn lots of attentions after the recent advances of wireless technologies. Target tracking aims at locating one or several mobile objects and depicting their trajectories over time. The applications of Object Tracking Sensor Networks (OSTNs) include but not limited to environmental and wildlife monitoring, industrial sensing, intrusion detection, access control, traffic monitoring, patient monitoring in the health-related studies and location awareness in the battlefield. One of the most rewarding applications of target tracking is wildlife monitoring. Wildlife monitoring is used to protect the animals which are endangered to extinction. Road safety applications are another popular usage of wildlife monitoring using WSNs.

In this thesis, the issues and challenges of energy efficient wildlife monitoring and target tracking using WSNs are discussed. This study provides a survey of the proposed tracking algorithms and analyzes the advantages and disadvantages of these algorithms. Some of the tracking algorithms are proposed to increase the energy efficiency of the tracking algorithm and to prolong the network lifetime; while, other algorithms aim at improving the localization accuracy or decreasing the missing rate. Since improving the energy efficiency of the system provides more alive sensors over time to locate the target; it helps to decrease the missing rate as the network ages. Thus, this study proposes to adjust the sensing radius of the sensor nodes in real time to decrease the sensing energy consumption and prolong the network lifetime.

The proposed VAriable Radius Sensor Activation (VARSA) mechanism for target tracking using wireless sensor networks tackles the energy consumption issues due to resource constraints of the WSNs. VARSA reduces the radio covered area of each sensor node to only cover the Area of Interest (AoI) which is the location of the target in tracking applications. Thus, VARSA aims at decreasing the sensing energy consumption which leads to encreasing the network life time. In addition, VARSA decreases the missing rate over time as it provides more alive sensors to detect the target compared to previous activation algorithms as the network ages. VARSA is compared to PRediction-based Activation (PRA) and Periodic PRediction-based Activation (PPRA) algorithms which are two of the most promising algorithms proposed for sensor activation. The simulation results show that VARSA outperforms PRA and PPRA. VARSA prolongs the lifetime of the network and decreases the missing rate of the target over time.

Acknowledgements

I would like to express my sincere thanks to my supervisor Professor Azzedine Boukerche for his guidance and support. His aspiring guidance and his passion for science along with his constructive criticism and illuminating views provided me the facilities to learn and grow. Without his supervision and constant help this dissertation would not have been possible. He has given me the opportunity to be a member of PARADISE research laboratory and to be surrounded by enthusiast knowledgable researchers in this laboratory.

I am deeply grateful to the members of PARADISE research laboratory for several discussions and inspiring comments. My special thanks go to Amir Darehshoorzadeh for his suggestions and valuable comments on this work. My gratitude also goes to Mahmood Salehi on whom I could count to tackle implementation and technical issues.

Last but not the least, my deepest appreciation goes to my wife and my parents for their endless love and support.

Glossaries

WSNs	Wireless Sensor Networks	**VANETs**	Vehicular Ad Hoc Networks
MANETs	Mobile Ad Hoc Networks	**OTSN**	Object Tracking Sensor Network
GPS	Global Positioning System	**AoI**	Area of Interest
NS2	Network Simulator 2	**NA**	Naive Activation
PA	Periodic Activation	**RA**	Randomized Activation
CGA	Coverage Guarantee Activation	**PRA**	PRediction-based Activation
PPRA	Periodic PRediction-based Activation	**VARSA**	VAriable Radius Sensor Activation
CH	Cluster Head	**CM**	Cluster Member
GH	Grid Head	**GM**	Grid Member
RSSI	Received Signal Strength Indicator	**ToI**	Time of Arrival
AoA	Angle of Arrival	**RFID**	Radio Frequency Identification
TI	Tracking Interval	**DC**	Duty Cycle
MSE	Mean Squared Error	**THT**	Tri-Hexagon Tiling

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Overview

Recent advances in wireless technologies such as Bluetooth, 802.11/WiFi and WiMAX provided the sensor nodes the ability to communicate and exchange data with each other wirelessly. A Wireless Sensor Network (WSN) consists of hundreds of such tiny low-power micro-sensor nodes with wireless communication and limited processing capabilities. One of the monitoring applications using WSN is target tracking which is locating one or several mobile targets and depicting their trajectory over a tracking time. Target tracking applications has lead to the emerging use of large scale WSNs.

1.1.1 Wireless Sensor Networks

Wireless Sensor Network (WSN) is a set of tiny low-power micro-sensor nodes with limited processing and communication capabilities. A WSN is deployed to gather the environmental information such as seismic, acoustic, magnetic, InfraRed and video data [97]. Sensors are equipped with radio components (radio transceiver and antenna), battery, electrical circuits and interfaces. The limitation of energy resources is the most challenging issue in all applications of WSNs.

In radio components, selecting the data rate is a compromise of energy efficiency and communication speed. In addition, frequent use of radio components may deplete the energy of the sensor nodes. Sensor activation algorithms select which subset of sensor nodes to keep awake and for how long to improve the energy efficiency while the minimum requirement of the application is met. All the components of the sensor nodes even the processor are able to switch between sleep and active mode to save more energy. Sensor activation has drawn lots of attentions in recent studies in order to maximize the efficient utilization of energy. The main aim of these approaches is to prolong the lifetime of the sensor network while fulfilling the requirements of the application.

A variety of proximity sensor nodes are commercially available. While some of the wireless sensor nodes provide the distance of the object to the sensor and/or its moving direction; binary proximity sensors only output one bit concerning the presence of the target in the visibility area of the sensor. The visibility area of the sensor node is defined as the area in which the sensor is able to detect the target. This visibility area is modelled as a circle with radius R_s in this thesis. Binary proximity sensors are significantly tiny, cost-effective and energy efficient. In addition, the location error has shown to be acceptable for non-critical applications such as wildlife monitoring. This location error is at most equal to the sensing radius of the sensor node which is normally less than 15 meters. In the other hand, smart sensors need a specific number of sensors to cover a location in order to be able to locate the target but one binary proximity sensor can locate the target.

Network resiliency is the ability of the network to fulfill its predefined task while there are some node failures. Deploying high-density sensor networks to improve the network resiliency and localization accuracy is economically feasible using these inexpensive sensor nodes. Thus, we take advantage of binary proximity sensors in our proposed solution for energy efficient target tracking.

Sensing energy consumption of a sensor node is a factor of the radius of the circular area under radio coverage of a this sensor [9]. The ability to tune the sensing radius of a sensor node in real-time is provided in recent active sensing technologies. The novel capability of recent proximity sensors to adjust their sensing radius results in more accurate localizations and less energy consumption for sensing. Selecting the best sensing radius to wake up the sensor with and/or adjust it dynamically is an intrinsic challenge due to the trade-off between energy efficiency and tracking quality. Thus, proposing a new scheduling algorithm which takes advantage of this new capability of sensor nodes to adjust their vicinity area dynamically might be a considerable progress in the field of target tracking in WSNs.

The topology of a wireless sensor network might vary from star to tree or mesh connections. Researchers have proposed lots of applications for WSNs in the past decade [72] [6]. These networks might be used to aggregate environmental monitoring data such as temperature, sound, pressure monitoring or physical monitoring data such as structural monitoring at a server node or a gateway. Another application of WSNs is healthcare and patient monitoring [98] [96].

1.1.2 Target Tracking Sensor Networks

One of the most rewarding applications of WSNs is target tracking which is an aim at locating one or several mobile objects and depicting their trajectories over time. A tracking sensor network is a sensor network which is used to track one or multiple moving objects within its visibility range. In a target tracking WSN, the entire network nodes collaborate in sensing and the gathered data is aggregated in a sink node, which uses the reported data to estimate the trajectory of one or several mobile objects called targets. The target could be an animal, a vehicle, a robot or a person, which is moving under the coverage area of the network. The target tracking algorithm might track a malicious moving object while ignoring other objects in the tracking field [87]. Some applications of target tracking might include but not limited to environmental and wildlife monitoring [72], industrial sensing, intrusion detection [6] [25], access control, traffic monitoring, patient monitoring in the health-related studies [98] [31] and location awareness in the battlefield. Among all applications of target tracking, wildlife monitoring has drawn tremendous attention in recent years to protect animals, which are endangered to extinctions or warn vehicles about an upcoming animal trespassing a roadway. It is reported that there has been 115000 deer-vehicle collisions in Pennsylvania, USA causing 400 million dollars in 2013.

Target tracking applications can operate in two different modes: surveillance and tracking [59]. In surveillance mode, the presence of the target in the tracking field is not determined yet. So, a complete or partial coverage of the network is required to trigger the tracking mode which keeps tracking the target initially detected in the surveillance mode. Authors of [116] suggest to always keep the nodes in the borders of the tracking area active to ensure the detection of any intrusions. Tracking mode keeps tracking of the target and aggregates the sensed data at a sink node.

Target tracking in WSNs inaugurate sever challenges due to the energy limitations of the sensor nodes and the required live performance of tracking, specially for high-speed targets. Several researches have been conducted to enhance the energy consumption of the sensor nodes considering their limited processing capabilities [9] [64] [114]. The proposed approaches are either an asset to decrease the consumed energy for communication [106] [74] or to decrease the sensing energy consumption while assuring the coverage of the Area of Interest (AoI) [100] [120]. Area of Interest is defined as the locations of the targets in the field of target tracking. Besides energy efficiency enhancement, the tracking quality is to be assured. Hence, some algorithms aim at increasing the tracking quality by addressing the localization issues [68] [109] [114]. Even though there are some limited studies in decreasing the sensing energy consumption; this field is still an open research area due to the new advances in sensing technologies.

Some sensors provide an approximate distance of the target to the sensor and/or its angle or direction of movement; but binary proximity sensors just provide one bit output regarding the presence of the target in radio coverage of the sensor. These tiny sensors have shown to be promisingly power efficient while their tracking quality is comparable to more complicated sensors [66]. Thus, we have decided to use binary proximity sensors. In addition, inexpensive sensors provide us with an opportunity to deploy a high-density sensor network, which is more tolerant to losing some nodes due to power exhaustion or environmental damages and can improve the localization accuracy.

Target can be equipped with a wireless transceiver to provide certain forms of information regarding its behaviour or identification [117]. The proposed tracking algorithms for these active targets are called cooperative [88] [75]. Non-cooperative target tracking algorithms do not use any information exchange between sensor nodes and the target [79] [111]. In this study, we investigate a passive target which is not equipped with any communication components and information exchange between the target and the sensor node is not possible.

A sensor node can be in one of these four operating modes: transmitting, receiving, idle or sleep. The first three states are called active modes. Energy consumption decreases from transmitting to sleep state. Moreover, the sensing module of the node might be activated in each of the aforementioned active states or not which also has a considerable impact in energy consumption. The problem of sensing energy consumptions of the sensor nodes has been addressed in the literature by covering the AoI using active sensors and sending the other sensor nodes to sleep [94]. This approach is highlighted as sensor activation [94]. As an example, Figure 1.1 shows a deer passing a tracking area indicated by a rectangle. The sensors near the deer trajectory are activated while other sensor nodes are in sleep mode in order to preserve their energy resources. The proposed algorithms for sensor activation are categorized into six groups: Naive Activation (NA), Periodic Activation (PA), Coverage Guarantee Activation (CGA), Randomized Activation (RA), PRediction-based Activation (PRA) and Periodic PRediction-based activation (PPRA). Sensor activation may also be referred as sensor scheduling.

Some sensor activation algorithms activate a subset of nodes [107], called a cluster,

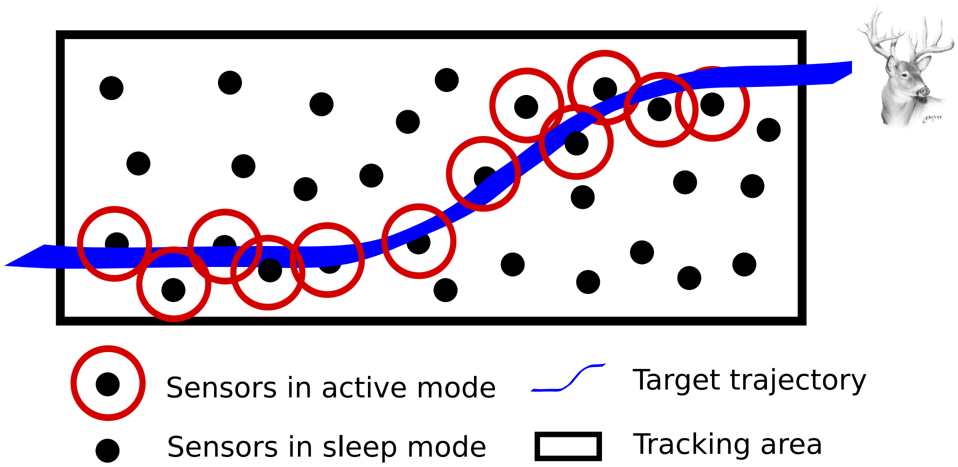

Figure 1.1: Sensor Scheduling in Tracking Sensor Networks

while others activate one node [43]. The cluster can be devised before the start of tracking as a static cluster or can be formed dynamically when the target enters the surveillance area [57] [108] [39]. Cluster activation algorithms are not energy-efficient as they need to activate a number of nodes at each interval. In addition, these algorithms need to exchange information between the cluster head and cluster members, which is an overhead in communication energy consumption. Two comprehensive surveys of the clustering algorithms for target tracking in wireless sensor networks are provided at [1] and [34].

Some filter-based algorithms have been proposed to improve location estimation. Applying Kalman filter and its extensions [103] [119] or particle filters [104] [70] for a tracking algorithm is shown to increase the localization accuracy of a tracking algorithm; but these techniques are computationally hard considering limited computation capabilities of binary proximity sensors.

1.2 Motivations for Target Tracking Using Wireless Sensor Networks

Energy limitations of the sensor nodes and the required live performance of tracking specially for high-speed targets are the main challenges of target tracking in WSNs. Even though switching the target nodes between sleep and awake has shown to be considerably energy efficient, the tracking quality is to be assured. While there are comprehensive researches to address localization issues [91] [26] or optimizing the communication energy consumption [42] [106]; minimizing the sensing energy consumption is still an open research area.

Recent advances in active sensing technologies provide the opportunity to adjust the sensing range of each sensor dynamically which can lead to deploy a more energy efficient algorithm with better tracking quality. Thus, proposing a new activation algorithm which takes advantage of this new capability of sensor nodes to adjust their visibility area in real time might be a considerable progress in the field of target tracking in WSNs.

1.3 Problem Statement

The problem of target tracking using WSNs requires to address several challenges due to the energy limitations of the sensor nodes and high probability of missing targets. Even though switching the target nodes between sleep and awake has shown to be considerably energy efficient, the tracking quality is to be assured. Selecting the best sensing radius to wake up the sensor with and/or adjust it in real time is also an intrinsic challenge due to the trade-off between energy efficiency and tracking quality.

Thus, we have summarized the problem of efficient sensing for target tracking using WSNs as follows. Given a binary proximity wireless sensor network, implementing a sensor activation algorithm to decrease the sensing energy consumption and to prolong the network lifetime with an acceptable localization accuracy. This scheduling algorithm should select which nodes to keep awake and with what sensing radius based on the current state of the network.

1.4 Research Objectives

The main objective of this thesis is to propose a new efficient sensor activation technique for target tracking using wireless sensor networks. We argue that the significant energy saved in sensing enhances the tracking quality of the system as the network ages. Thus, the proposed algorithm is an asset to both prolong the network life time and to improve the tracking quality.

A novel tracking system is designed and evaluated in this thesis. The proposed tracking system include the sensor nodes deployment, initialization, tracking, location prediction, localization and information aggregation a the sink node. The entire system is elaborated in details in the following sections. In addition, the proposed tracking algorithm requires to be error resilient; hence, a recovery mechanism is required to overcome the node failures or prediction errors.

Another aim of this thesis is to provide a comparative analysis for current sensor activation algorithms. Extensive simulations are conducted to study different classes of sensor activation algorithms using Network Simulator 2.35 (NS-2.35).

1.5 Contribution

Some of the contributions of this thesis are but not limited to:

- We have categorized and compared different sensor scheduling techniques for target tracking in WSNs.
- A novel sensor scheduling mechanism is proposed.
- We have designed a novel target-tracking algorithm, which includes localization, prediction model, sensor scheduling, recovery and data aggregation at the server node. Our objective is to decrease the sensing energy consumption by keeping the sensor in sleep mode or decreasing its sensing radius; while keeping localization accuracy within acceptable range.
- The performance of different sensor activation algorithms is compared through extensive simulations using Network Simulator 2 (NS-2.35).

1.6 Thesis Organisation

The remainder of this thesis is organized as follows:

- Chapter 2 surveys the proposed algorithms for target tracking. This chapter discusses the sensor node deployment algorithms. Then, this chapter summarizes the proposed localization techniques. Afterwards, sensor activation algorithms are elaborated. A thorough analysis and comparison of the these sensor activation algorithms are provided in this chapter.
- Chapter 3 provides a comprehensive methodology including the research model, research hypothesis, network model and target mobility model. Then, the simulation

software and the network settings are elaborated. This chapter aims at making this study reproducible in other research studies.

- Chapter 4 describes the proposed VAriable Radius Sensor Activation (VARSA) algorithm in details. It elaborates different states of VARSA, target movement model, prediction method and the applied localization technique and routing algorithm.
- Chapter 5 outlines the simulation setup and the evaluation metrics. Then, it provides the parameter tuning including adjusting the tracking interval, duty cycle of tracking and the rate of reducing the radio covered area of each sensor node. Then, this chapter provides a comprehensive comparison of VARSA with two other promising sensor activation algorithms.
- Chapter 6 concludes this thesis. It also provides some future research directions for extending this study.

Chapter 2

Related Works

2.1 Introduction

This chapter highlights various promising algorithms that have been proposed for target tracking using WSNs. Target tracking in WSNs has been studied with different flavours. Some of the proposed target tracking algorithms are assets to improve the energy efficiency of the tracking system [106] [100] [54]. Energy-efficient sensor deployment [19], computing and processing energy consumption, communication energy consumption [24] and sensing energy utilization are some of the the challenges to be addressed in WSN tracking algorithms [97]. In the other hand, some algorithms try to enhance the localization accuracy and tracking quality [73] [109] [68] [11]. The problem of having faulty nodes in the network has been also well-investigated in the literature [50] [23] [49] [51] [95] [22] [99] [30]. Obstacles might interfere the communication of the nodes in the network; however, a cooperative algorithm is proposed in [2] to overcome this challenge. Routing the sensed data to a sink node is another research field in target tracking applications [8] [12] [16] [105] [33] [29].

To better understand the problem of target tracking in WSNs and the proposed solutions in the literature, various literature review and survey articles have been published recently [27] [10] [61] [94] [53]. Some of these studies discussed the proposed algorithms for localizations; while other papers study the energy consumption enhancement of the tracking algorithms.

Some of the localization improvement techniques are presented for smart sensors and might need several nodes to cover an area in order to locate the target, while, other algorithms are proposed for binary proximity sensors. To begin with, this chapter provides an introduction to the sensor deployment techniques for target tracking using WSNs; then, it elaborates the localization improvement techniques. Afterwards, energy consumption enhancement techniques are discussed. This chapter also elaborates how the Kalman filter and its extensions and particle filters are used to improve the location estimation accuracy of the tracking algorithms.

2.2 Sensor Deployment

Sensor deployment strategy plays a crucial role in the performance of the tracking algorithm. The place of the sink node can be steady or mobile [92]. Steady sink node always remain at the same location and therefore has access to an infinite sources of energy. In the other hand, mobile sink node needs to track the target and reach its location at the shortest possible time.

The sensor nodes can be deployed randomly or in a deterministic fashion. Deterministic algorithms define the exact coordinates of the location of the sensor node at the time of deployment, while random strategies, also called nondeterministic, assign a probability for each location to host a sensor node. Nondeterministic algorithms facilitate the random sensor deployment and makes it more cost-effective specifically for difficult-to-access environments [21]. In the other hand, deterministic sensor deployment can assure the level of coverage in the network [20]. An *k-coverage* in the network reveals that each point in the network is at least covered by k sensor nodes [93]. Uniform Random sensor deployment and two of the deterministic sensor deployment techniques, Grid and Tri-Hexagon Tiling (THT), are elaborated in the following sections.

2.2.1 Uniform Random

Uniform random deployment of the sensor nodes scatters the sensor nodes in the tracking field randomly. The probability of the presence of a sensor node in an specific location is the same as the other locations in the tracking field. This deployment strategy is cost-effective as the sensors can be thrown out from an airplane [93]. It is shown in [77] that uniform deployment of sensor nodes outperforms the deterministic sensor deployments in a mostly sleeping sensor network. The state of art tracking algorithms using WSNs utilized the network as a mostly sleeping sensor network since it only covers the AoI.

2.2.2 Grid

A grid-based sensor deployment layout can be a unit square, equilateral triangle or a hexagon. Each grid point in the network reveals the location of a sensor node. This sensor deployment technique can be used to assure a certain level of coverage or connectivity in the network. However, deploying sensor nodes in specific locations is a high-cost mission and it can be even infeasible in some circumstances. For instance, square grid sensor deployment provides 2-coverage, 3-coverage and 4-coverage areas in the network [93]. Figure 2.1 shows the square grid sensor deployment in a rectangle tracking field.

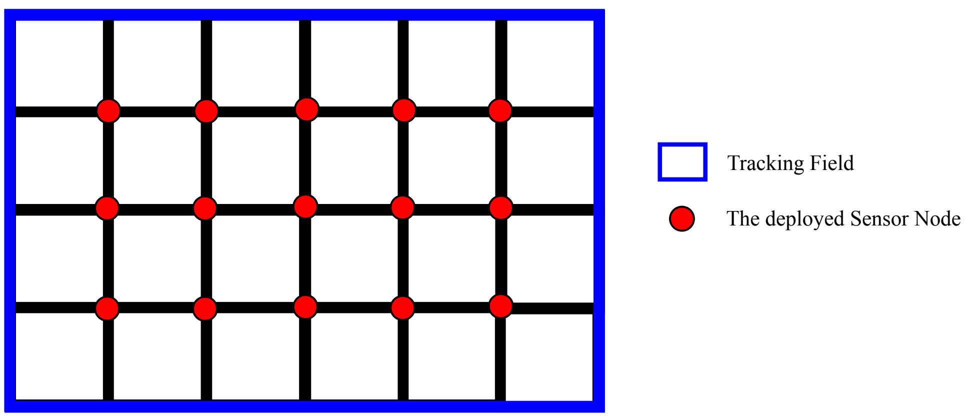

Figure 2.1: Square Grid Sensor Deployment

2.2.3 Tri-Hexagon Tiling (THT)

Tri-Hexagon Tiling aims at covering the whole sensor network without any uncovered area using hexagon and triangle grids. In this technique, the tracking field is covered by hexagons aligned in a grid line and then, the uncovered area in the tracking field is covered by the triangle grids. THT provides areas of 2-coverage, 3-coverage and 6-coverage in the network [93]. THT is not still practical for real applications. Figure 2.2 represents how a combination of triangle and hexagon grids can be used to select the position of the sensor nodes in the tracking field.

2.3 Sensor Nodes Localization

Localization techniques for sensor nodes tackle the challenges of finding the location of a sensor node in the tracking area [26] [36] [3] [45]. Due to the high-cost and unfeasibility

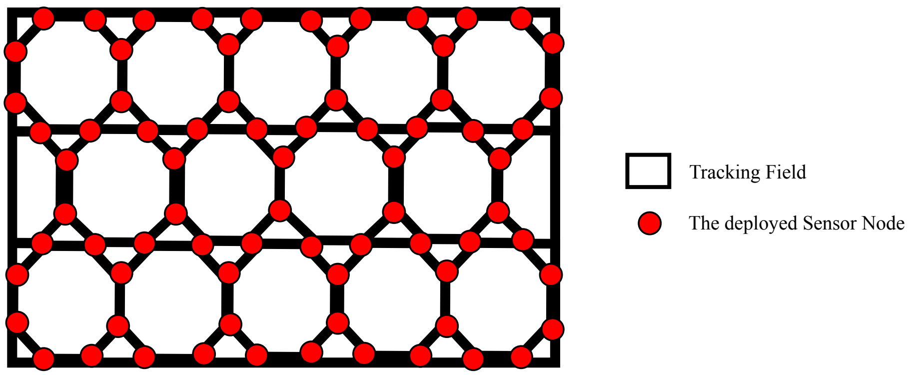

Figure 2.2: THT Sensor Deployment

of equipping all sensor nodes with a GPS module, only some of the nodes are aware of their location in a sensor network. These nodes are referred to as *beacons*. Other sensor nodes use a localization technique to estimate their location by finding their distance to these beacon nodes.

There are two well-known techniques to estimate the distance of a sensor node to a beacon: Received Signal Strength Indicator (RSSI) and Time of Arrival (ToA) [36]. RSSI uses the power of the received signal to estimate the distance between the receiver and the sender. In the other hand, ToA takes advantage of the time which a message transfers between the source to the destination node to estimate the distance of the two nodes.

Some other approaches in the literature argue that all sensor nodes should be able to locate themselves using a GPS module; however, a scheduling algorithm should be designed to select which sensors should activate their GPS module while the others send their GPS module to sleep mode to save the energy [85]. These techniques are not cost-effective as the price of implementing GPS modules in a tiny proximity sensor is considerably high for tracking in a large tracking area with a dense deployed sensor network.

After finding the distance, a localization technique should be used to find the location coordinates of the sensor nodes. Trilateration and Multilateration are two of the widely deployed localization techniques [26]. Trilateration localization technique locate the target based on its distance to three beacon sensor nodes. In case of having accurate measurements, the location of the sensor can be exactly derived using its distance to three beacons. However, due to the measurement errors and uncertainties, Trilateration is not a practical localization technique [36]. Thus, Multilateration technique uses an optimization method to solve a set of equations derived from the distance of this sensor node to several beacons in order to estimate the location of the node [101].

The Angle of Arrival (AoA) can also be used to determine the location of the sensor node. Receiving three or more wireless signals from different beacons, a sensor node can calculate its location [89]. The limitation of AoA is the estimation errors caused by multi-path reflections.

2.4 Target Localization

In this section, the proposed localization techniques to estimate the location of the target and its trajectory using the data aggregated from binary proximity sensors are discussed. Then, we justify the localization technique that we take advantage of in VARSA.

2.4.1 Centriod Localization

Centriod localization approximates the location of the target as the average of coordinates of all sensor nodes which has the target in their radio coverage area [73]. Equations 2.1 and 2.2 represent the estimated x and y coordinates of the target respectively, where n sensors located at (x_i, y_i) has detected the target at time t_0.

$$x_{target,t_0} = \frac{\sum_{i=1}^{n} x_i}{n}, \tag{2.1}$$

$$y_{target,t_0} = \frac{\sum_{i=1}^{n} y_i}{n}, \tag{2.2}$$

2.4.2 Weighted Centriod Localization

Authors of [38] argue that if the target stays longer within the coverage area of a sensor node; it might have been closer to that sensor node. The algorithm weights the coordinates of different sensors which detected the target over time based on the duration that they have detected the target. Then, it depicts the target trajectory as the best line which fits these weighted points. It also refines the last estimated trajectories using the current accumulated data to best represent the actual trajectory of the target.

2.4.3 Arc Estimation

When a target passes from the radio coverage of one sensor to another, target location is estimated as the middle point of the arc which is an intersection of the circles representing the visibility area of the two sensor nodes in [109]. Each sensor node generates a *True* bit if it detects the target and a *False* bit if there is no target in its visibility area. Sensor nodes do not report any information to the tracker unless their state changes. Each time a report is received at the sink node, it finds the arc which the target is crossing and the target location is estimated as the centre of that line. When a sensor reports the start of sensing the target, the target is in a point within the circle with the centre of node location and radius of its sensing radius. In addition, the target cannot be in the visibility area of all nodes with *False* output and is within the visibility area of nodes with *True* output. Thus, the sink node estimates the location of the target at the time of state change as the centre of the arc which is on the border of the sensing range of the new detector of target and has no parts in the visibility area of the sensor nodes which did not detect the target at the transition time.

The smaller the arc with the estimated location of target is, the more precise the estimation that we might have using the arc estimation method. Thus, the algorithm weights the smaller arcs such that the depicted trajectory is closer to the points in smaller arcs [109]. This weight is calculated using

$$w = \frac{|circle|}{|arc|} \tag{2.3}$$

where $|arc|$ is the length of the arc which includes the estimated location of the target and $|circle|$ is the length of the circle indicating the radio covered area of the sensor.

2.4.4 Piecewise Linear Segments

The proposed approach in [102] suggests to send the time flags that the target has passed from one sensor to another to the sink node. Then, the sink node finds the trajectory of target based on these transition times. The sink node finds a region F which contains the visibility area of all sensor nodes which detected the target at the current tracking interval unless the parts which contains the visibility area of the sensor nodes which did not detect the target.

$$F = \bigcap_{i \in I} S_i - \bigcup_{i \in Z} S_i \tag{2.4}$$

where S_i is the radio covered area by sensor i, the subset of sensor nodes which did not detect the target at the current tracking time interval is I and the subset of sensors which detected the target at this tracking time interval is Z. Limiting the possible location of the target by region F, the algorithm prefers the simplest way to estimate the trajectory of the target within this region. Since the linear estimation has the least complexity, the algorithm aims at locating the target using a piecewise linear trajectory with fewest number of segments. Figure 2.3 shows the region F which binds the location of target and the piecewise linear estimation of the target trajectory.

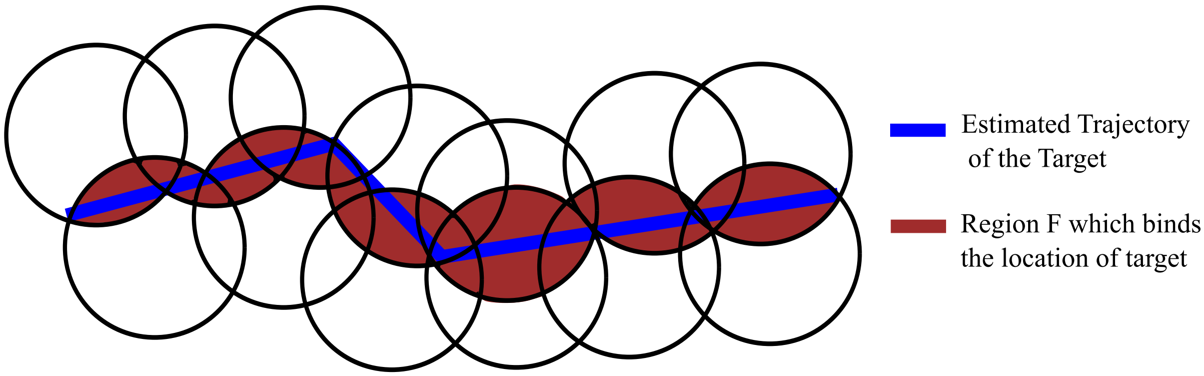

Figure 2.3: Piecewise Linear Estimation of the Target Trajectory [102]

2.4.5 Tanget and band Method

Tanget and band methods are proposed in [68] and [67] for sparsely deployed networks, when there might be several coverage holes which are not covered by any sensors. This approach provides a technique to estimate the location of the target while there is no information from its location using the gathered data from past and the current data. Even though it might not provide a good live performance; but it is shown to be effective for noncritical applications which are not designed for live operations.

Tangent method

Tangent method is a localization technique which uses an estimation of the distance that the target has travelled within visibility area of a specific node to estimate the trajectory of the target. This technique measures the time that the target is moved through the visibility area of a node. Then, the algorithm calculates the distance, d_i, that the target might have travelled in the visibility area of sensor s_i assuming constant velocity of the target and linear movement during staying at one node's visibility area. Given three consecutive distances, (d_i, d_{i+1}, d_{i+2}), the trajectory of the target can be estimated as a line while transferring through the visibility area of three sensors (S_i, S_{i+1}, S_{i+2}). Figure

2.4 represents how the location of the target can be estimated while the target is not within the visibility of any sensor nodes in the network.

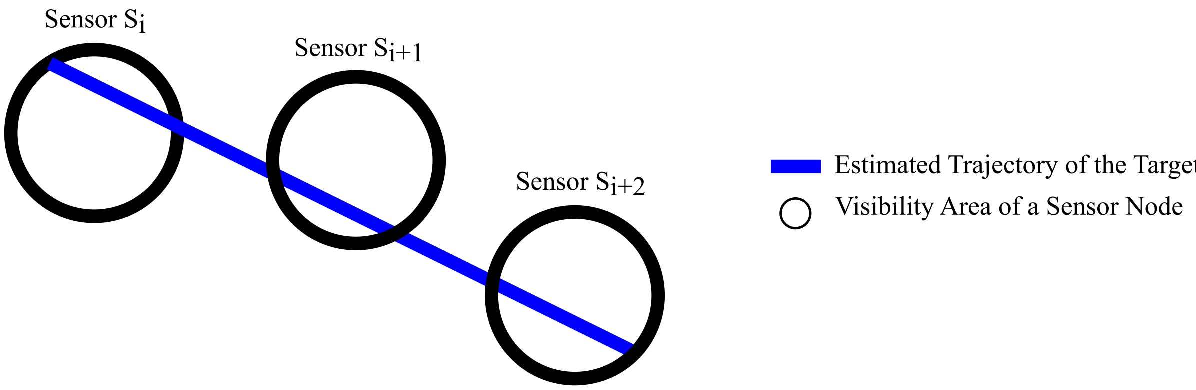

Figure 2.4: Tangent Method Localization Technique [68]

Finding a line which passes three sensor nodes visibility area within a certain distance is not always feasible due to measurement errors. In addition, the target might not always move linearly. Thus, the authors suggested to find an approximate common tangent to three circles with the centre of the three nodes and tangent to the three chords (d_i, d_{i+1}, d_{i+2}). When four consecutive nodes detect the target, one line can be estimated for the target trajectory using the first three nodes and another linear trajectory can be found for the last three sensors. These two lines intersect at a point. The algorithm picks the line from the first three sensors as the trajectory of the target for before the intersection of the two lines and the trajectory after the intersection is estimated to be on the second line achieved from the sensed data of the last three nodes. Figure 2.5 demonstrates how the algorithm can find the trajectory of the target.

Band Method

Band method estimates the distance that the target travels between two nodes as a product of a constant velocity assumed for the target and the time difference from the moment the target leaves the visibility area of one node to the moment that target enters the visibility area of another node [68]. This distance, called d_i^{out}, provides a band between two sensors for the target trajectory. Then the algorithm takes advantage of d_i^{in}, which is the distance that the target travels within the visibility area of one node, to further decrease the band on the trajectory of the target.

Figure 2.6 can further explain the proposed localization algorithm. Authors of [68] argued that if the target travels a distance d_i^{out} from Sensor S_i to Sensor S_{i+1}, the

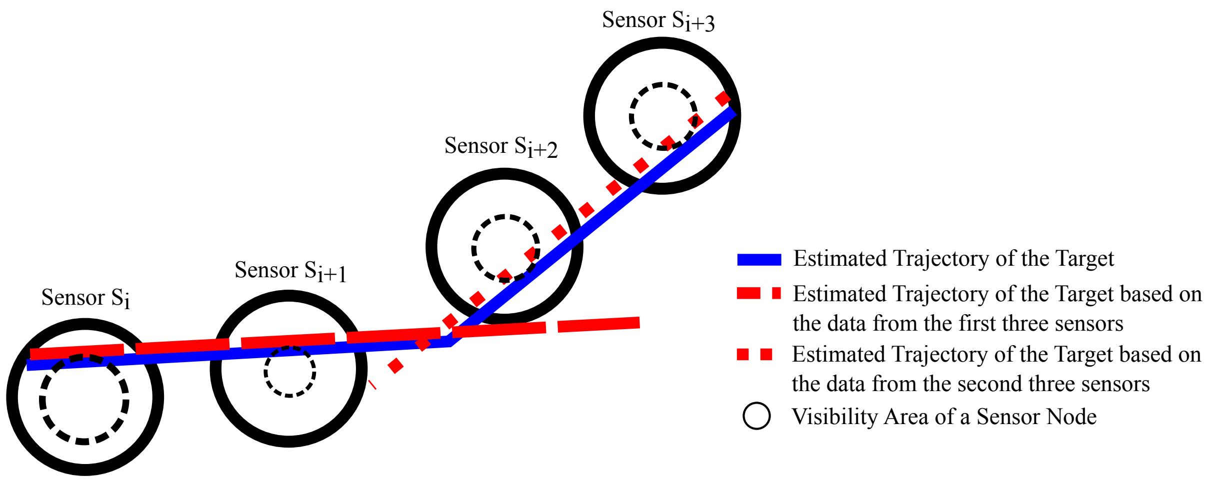

Figure 2.5: Estimating Target Trajectory using Tangent Method [68]

trajectory of the target is between AB and CD. Thus, the target exits Sensor S_i from Arc AB and enters Sensor S_{i+1} within Arc CD.

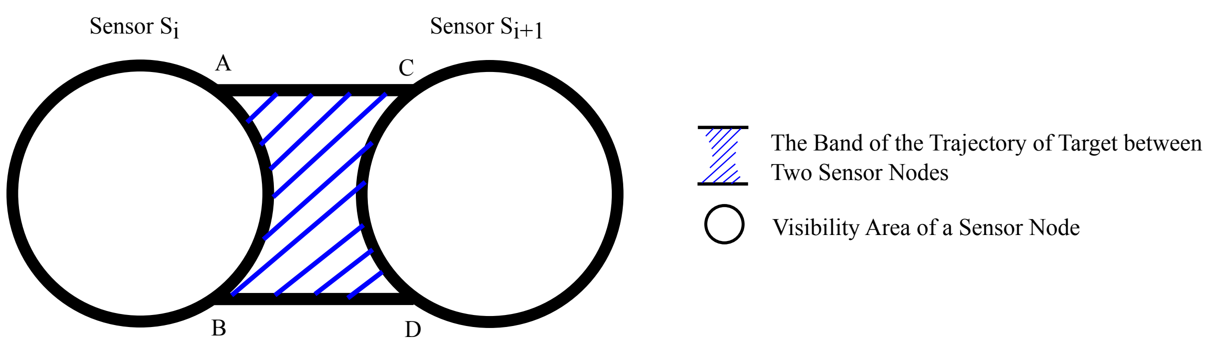

Figure 2.6: Limiting the Feasible Target Trajectory between Two Sensors using Band Method [68]

When the trajectory band between the two sensors are found, the algorithm determines the band for the target movements within visibility area of the current sensor node. For each sensor, there is an arc limiting the area that the target can enter the visibility area of a node and another arc for limiting the exit location of the target. In Figure 2.7, the entrance point of the target is limited by Arc EF and its exit point is limited by Arc AB. Assuming the target entered the visibility area of the node from point E, it can exit from G or H traversing d_i^{in} in the sensors radio covered area. If the entrance location moves toward F, the exit point moves toward I and J. Thus, the feasible exit region is on either Arc HI or Arc GJ. Finally, given these two feasible arcs for exit points of the target and Arc AB for feasible exit point of the target from Sensor i, the possible band for the target trajectory can be reduced to the area surrounded by $IKBD$ as shown in Figure 2.7.

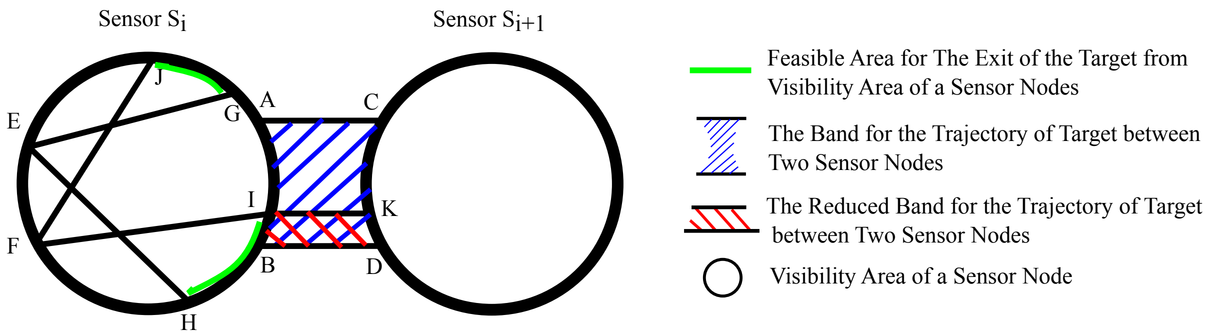

Figure 2.7: Decreasing the Feasiible Trajectory of Target between Two Sensor Nodes using Band Method [68]

2.4.6 Polygon Representation of Location

It is suggested to indicate the location of the target with a hexagon in [114]. The tracking area is divided to several hexagons and each sensor node covers one of them. Instead of providing an exact coordinates of the target, the algorithm returns a set of areas which are shown with symbols representing a hexagon.

2.5 Comparison of the Target Localization Techniques

The proposed localization techniques for target tracking in proximity WSNs are summarized in Table 2.1. As indicated in the table, some of these applications need time synchronization between sensor nodes. Some algorithms require to have overlapped coverages to locate a target; while some other approaches locate the target with one active node. In addition, tangent and band methods are able to locate the target using the data over time even though there is no information from the current location of the target. This lack of information could be due to network holes where no node covers those areas. Velocity should be considered constant in some of these techniques while others have no constraint in the changes of the velocity of the target.

Table 2.1: Localization Techniques for Target Tracking in Proximity WSNs

Technique	Sync.	Overlapped Coverage	Needed Sensors for Localization	Constant Velocity
Centriod [73]	No	Yes	1	No
Weighted-Centriod [73]	No	Yes	1	No
Arc Estimation [109]	No	Yes	3	No
Piecewise Linear Segments [102]	Yes	Yes	2	No
Tanget Method [68] [67]	No	No	0	Yes
Band Method [68] [67]	Yes	No	0	Yes
Polygon Localizarion [114]	No	No	0	No

2.6 Sensor Activation Techniques

The probability that a sensor node detects a target in its visibility area is low as the sensing range of a sensor node is much smaller than the tracking area. In addition, even if the target is close to the sensor node, its information might be redundant as there might be more sensors covering the same area simultaneously [81]. Thus, a sensor activation mechanism should be defined to decide which sensors to wake up and the duration of the activation. Sensor activation is the technique of efficiently choosing a subset of one or more sensors in the area of interest to be in active mode with an appropriate sensing radius; while the other sensor nodes are sent to sleep. The most challenging issue in sensor activation is to pick an appropriate size of coverage; since size of the covered area in the network is a compromise of energy efficiency and tracking quality. In addition, the interval of tracking might be adjusted based on the application requirements and/or the velocity of the target to provide an acceptable tracking quality with minimum energy consumption.

The idea of activating the sensors in the area of interest while other nodes are in sleep mode has drawn lots of attentions in the recent literature [62] [84] [41]. The proposed tracking algorithms can be classified into six categories: Naive Activation, Periodic

Activation, Coverage Guarantee Activation, Randomized Activation, Prediction-based Activation and Periodic Prediction-based Activation. The rest of this section elaborates these techniques in more details.

2.6.1 Naive Activation (NA)

This mechanism is entitled as Naive Activation as this is the most immature technique in which all the sensor nodes are always active [38]. In this approach, the entire network is always sensing and communication active and the location report is sent to the sink node after each tracking interval. If the entire tracking area was initially covered by the sensors at the time of sensor deployment, the target is never lost unless there is a node failure or a blockage. However, the entire network exhausts its energy pretty soon after the start of tracking due to the high sensing energy consumption. Naive sensor activation provides high location accuracy and almost zero missing rate while the network is not dead, but it is not energy-efficient. The network lifetime is very short and all the sensor nodes exhaust their energy pretty soon after start of tracking. Short life time and energy inefficiency of NA leads the researchers to propose more energy efficient algorithms; even though NA provides high localization accuracy and almost zero missing rate.

2.6.2 Periodic Activation (PA)

PA suggests to alternate the entire network between active and sleep mode periodically [38]. This technique is shown to be effective and more efficient than Naive target tracking. PA is able to save a considerable amount of energy by sending the idle sensors to sleep; however, it might deactivate the sensor nodes in the AoI which leads to increasing the missing rate and location inaccuracy. Selection of interval time of tracking is a compromise between energy consumption and tracking quality as increasing the interval of tracking leads to a worse location accuracy and better network life time.

PA activation techniques might deploy a fixed tracking interval to switch the sensor nodes between sleep and active or they might use an adaptive tracking time interval. Adaptive tracking time interval aims at increasing the energy efficiency by increasing the period of tracking and its duty cycle when the target is not near a sensor node or when its velocity is high. But, if the target moves fast or it is near a tracking area, the tracking time interval is decreased to assure the tracking quality of the algorithm.

The tracking time interval can also vary depending on the operational purposes of the sensor node. The sensor activation algorithm proposed in [69] considers two different distributions for tracking time interval for the sensors in current AoI and sensors far from this area. The algorithm divides the tracking area by a virtual grid. One sensor in the grid is called Greed Head (GH) and the rest of the sensors are Grid Members (GMs). GH is responsible for the surveillance state. A virtual grid is in Surveillance state when there is no target in this grid; So, only the GH is active to detect any target entering this grid. Then, if the GH finds a target in the grid, it activates more sensor nodes within the grid and the activated GMs collaborate to locate the target. This approach combines the fixed and adaptive tracking time interval for periodic activation.

2.6.3 Coverage Guarantee Activation (CGA)

Coverage guarantee activation techniques ensure the k-coverage of the whole tracking area [71]. A cost is assigned to each sensor to determine if it needs to be active or not. This cost is an exponentially increasing function of the remained energy of the sensor node. The set of sensors which cover a point with the minimal cost are activated in each tracking interval. After each tracking interval, the cost is calculated again and the active subset of nodes in the network might change.

CGA aims at maximizing the network lifetime while ensuring the k-coverage of all the points in the network. Thus, it does not take the behaviour of the target into consideration and regardless of the location of the target, the whole network should be covered by the sensor nodes. In addition, all sensor nodes need to be synchronized to update their cost functions and a huge amount of data transmissions are required at each time interval to update the cost of neighbouring nodes so that a node can decide to wake up or stay sleep.

2.6.4 Randomized Activation (RA)

In this approach, each node is activated with a probability of P at each time interval of tracking. It has been proposed in [91] to increase the probability of awaking the sensors closer to the current location of target. Assuming the limited velocity for the target, the displacement of target is limited in each tracking time interval and RA is able to locate the target by increasing the probability of activating sensor nodes in the AoI. The proposed algorithm is distributed and the current detector transmits control packets to neighbour nodes to increase their probability of waking up.

2.6.5 PRediction-based Activation (PRA)

Prediction-based sensor activation techniques were the first approaches in the literature to cover the area of interest while the rest of the sensor nodes are sent to sleep mode for energy efficiency [106] [121]. PRA activates a cluster of nodes in the predicted AoI in each tracking time interval. This cluster consists of a cluster head and cluster members. The current cluster head decides which detector to wakeup for the next time slot as the cluster head. Each cluster might be a set of one or more nodes. A cluster might include one or more sensor nodes.

Size of the cluster is a trade-off of energy efficiency and localization accuracy. The minimum size of the cluster to satisfy a Mean Squared Error (MSE) for the target location can be estimated as suggested in [65]. MSE is the average of the squared errors on the estimated location of the target and its actual location. A hierarchical PRA tracking algorithm is suggested in [90] which uses a super-node with extra capabilities in each cluster. This super-node is responsible for data fusion and transferring the sensed data to the super-node in the next predicted cluster. An energy cost function is defined in [118] to decide which node to activate in each tracking interval. This cost is a function of the residual energy of the sensor node, its initial energy and the number of times that it has been scheduled for sensing.

Prediction Method

Several prediction mechanisms have been proposed in target tracking literature [64] [103] [56]. These prediction models can be classified into three categories based on the data used for prediction: Spatial, Temporal and Spatio-Temporal.

- **Spatial Prediction:** Spatial prediction techniques estimate the next location of the target using the data collected at the current time from the neighbour nodes [64].

- **Temporal Prediction:** It is proposed to estimate the next location of the target based on the data of one node over time in Temporal Prediction method. Thus, each current node predicts the future location of the target using its sensing history [103].

- **Spatio-Temporal Prediction:** Some researchers have proposed to take advantage of both of the aforementioned prediction techniques to estimate the location of the target using Spatio-Temporal data [56].

Movement Model

There are various movement models to be used for predicting the next location of the target.

- Linear Model: Linear model for the target movement assumes that the direction and velocity of the target is constant in a short interval of time. This model simplifies the calculations and avoids unnecessary processes at the sensor nodes considering the limited processing capability of nodes. The velocity of the target is estimated as 4.2 and 4.3 in a planar area.

$$V_{x,t_i} = \frac{x_{t_i} - x_{t_{i-1}}}{t_i - t_{i-1}}, \tag{2.5}$$

$$V_{y,t_i} = \frac{y_{t_i} - y_{t_{i-1}}}{t_i - t_{i-1}}, \tag{2.6}$$

  This approach predicts the x and y coordinates of the next location of the target after a tracking time interval using Equations 2.7 and 2.8 respectively.

$$x_{t_{i+1}} = V_{x,t_i} \times (t_{i+1} - t_i) + x_{t_i}, \tag{2.7}$$

$$y_{t_{i+1}} = V_{y,t_i} \times (t_{i+1} - t_i) + y_{t_i}, \tag{2.8}$$

- **Average Movement Model:** It is suggested to estimate the velocity of the target as the average of the velocities of target during the past time in [106]. This approach is an overhead in communication energy consumption since the velocity records should be transferred to the current detector.

- **Weighted Average Movement Model:** This movement model assigns more weights to recent velocity data compared to the data collected previously [106]. This approach does not only consume more energy for communication; but it also needs time synchronization between different sensor nodes.

- **Pattern Recognition Movement Models:** The last category of the movement models applies artificial intelligent techniques to model the movement of the target. S. Samarah et al. generate a sequence of $[Lastdetector, Currentdetector, Nextdetector]$ to train the system using artificial intelligent techniques in order to predict the location in [100]. Authors argue that targets might follow a similar movement pattern specially in wildlife monitoring where there is limited sources of water on specific locations for animals. Computation complexity of artificial intelligent techniques may not best suit the sensor nodes limitations. It is also suggested in [32] to use pattern recognition techniques in order to find a temporal relation between the outputs of different sensor nodes.

Wake-up Mechanism

The wake-up mechanism determines the size and location of the cluster of nodes to be activated in each time interval. Three different approaches have been proposed to select which nodes to activate [115].

- **Heuristic Destination:** In this approach approach, one node is only awaked at each tracking time interval. The selection of which node to wake up can be based on their closeness to the predicted location of the target and their remained energy [46]. The probability of missing the target is higher than other approaches but the consumed energy is considerably less than other algorithms and the increased missing rate can be compensated using some recovery mechanisms.

- **Heuristic Route:** Heuristic Route activation aims at decreasing the missing rate of the target by waking up all the sensor nodes on the route from the current detector of the target to the next predicted location of target. This approach is based on considering prediction errors on the velocity of target and precise prediction for the direction of movement.

- **Heuristic All-NBR:** All the neighbours of the nodes in the route from the current detector to the predicted sensor to locate the target at the next tracking time interval are waked up. This wake up mechanism can overcome minor error on both velocity and movement direction predictions.

After predicting the next location of the target, cluster head requires to wake up next predicted cluster members. CET algorithm selects a node in the direction of the target movement as the cluster head and more nodes are activated based on their closeness to the target location by the cluster head [106]. Some approaches propose to only wake up one node. It is also proposed to consider both the closeness of the predicted location of the target to a sensor node and the sensor's residual energy to select the next sensors to activate among different candidates in [46]. Activating one node provides an energy efficient technique which is susceptible to higher missing rates. Assuming an error on absolute value of the estimated velocity, it is proposed to activate all nodes in the route from the current detector to the next predicted detector. In addition, the neighbours of this route might be activated to leave a room for direction estimation mistakes.

Some of the proposed sensor activation approaches have considered the problem of sensor scheduling as an activation problem [54]. These algorithms aim at optimizing one of the performance metrics such as coverage, energy consumption or tracking quality. Tracking WSN is a random process; thus, target movement and sensor activation is modelled by two Markov chains with one step transition probability matrix in [54]. At each tracking time interval, the algorithm decides which nodes to activate based on user-defined objectives.

To further improve the energy efficiency, sensing coverage area of the sensor nodes might be adjusted during network deployment to provide minimal average of the sensing radius [121]. In [9], authors took advantage of Voroni-Laguerre diagram to discern redundant radio covered area. Then, the radio covered area of the network is decreased by either sending more nodes to sleep mode or decreasing the sensing radius of the current active nodes.

In [115], authors proposed to activate a cluster of nodes ahead of the moving target using PRA. The proposed dual prediction algorithm calculate the next location of the target at both the cluster head and cluster members. If the sensed data is analogous to the predicted locations, the report will not be sent to the cluster head to avoid unnecessary transmissions. However, the location history should be transmitted from the current node to the next node.

Extended Kalman Filter for Target Tracking

Kalman Filter (KF) was first proposed by R.E Kalman as a recursive solution to solve discrete-data linear filtering problem. This mathematical formulation for estimating the state of a process minimizes the squared mean error. The recursive mechanism of KF uses the past and present state to better estimate the future state of a system. This estimation improvement is achieved using a feedback control by estimating the process state at a time and improving the estimation using the noisy measurement.

In target tracking algorithms, the accuracy of the location predictions can be improved by applying Kalman filter and its extensions [4] [83] [119] [5]. EKF is used to update the estimation having the data sensed by the sensor. In addition, EKF can be used to predict the next location of the target in advance to activate a subset of sensor nodes in that area. Thus, the main two components of EKF is *time update* equations and *measurement update* equations. Time update equations are used to predict the next state of the system in advance to have a priori estimate of the next location of the target. Measurement update equations provide the system the ability to improve the measured state of the system using the priori estimate and measurement function [110]. This enhanced estimation of the state of system is called posteriori estimate.

The state of the target motion can be represented by its coordinates and its velocity vector. Thus, the target state, denoted as $X(K)$, can be shown as

$$X(K) = (x(k), v_{x(k)}, y(k), v_{y(k)}), \tag{2.9}$$

where $x(k)$ and $y(k)$ are the coordinates of the location of target at time t_k and the horizontal and vertical components of the velocity of the target are denoted by $v_{x(k)}$ and $v_{y(k)}$ respectively. The target movement is modelled as Equation 2.10

$$X(K+1) = F(\Delta t_k)X(k) + w(k, \Delta t_k), \tag{2.10}$$

where $F(\Delta t_k)$ is the state transition matrix, $w(k, \Delta t_k)$ is the process noise and Δt_k is the tracking interval time, $t_{k+1} - t_k$, as devised in [82]. Assuming a linear model for the target movement, A can be derived as

$$A = \begin{pmatrix} 1 & \Delta t_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t_k \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{2.11}$$

Assuming to have one active tasking sensor at each tracking time interval, the measurement model is given by:

$$Z_i(k) = h_i(X(k)) + v_i(k), \tag{2.12}$$

where h_i represents measurement function depending on $X(k)$ and measurement noise is denoted by $v_i(k)$. Measurement noise and process noise are independent white zero-mean Gaussian probability distributions. EKF predicts the next state of system, $\hat{X}(k+1|k)$, given the estimate of $\hat{X}(k|k)$ of $X(k)$ at time t_k as

$$\hat{X}(k+1|k) = A\hat{X}(k|k), \tag{2.13}$$

Given the covariance of the estimated state $P(k|k)$, the predicted state covariance can be calculated using:

$$P(k+1|k) = AP(k|k)A^T + Q, \tag{2.14}$$

Assuming sensor j is activated to locate the target, the predicted measurement of sensor j is:

$$\hat{Z}_j(k+1|k) = h_j(\hat{X}(k+1|k)), \tag{2.15}$$

The difference between the actual measurement of sensor j and the predicted state is called innovation:

$$\gamma_j(k+1) = Z_j(k+1) - \hat{Z}_j(k+1|k)), \tag{2.16}$$

Covariance of innovation is:

$$S_j(k+1) = H_j(k+1)P(k+1|k)H_j^T(k+1) + R_j(k+1), \tag{2.17}$$

where $H_j(k+1)$ is the Jacobian matrix of h_j at time t_{k+1} with respect to the predicted state $\hat{X}(k+1|k)$. Then, the estimation is updated as:

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)\gamma(k+1), \tag{2.18}$$

with the covariance matrix:

$$P(k+1|k+1) = P(k+1|k) - K(k+1)S(k+1)K^T(k+1), \tag{2.19}$$

In [103], Q is considered as:

$$Q = q \begin{pmatrix} \frac{\Delta t_k^3}{3} & \frac{\Delta t_k^2}{2} & 0 & 0 \\ \frac{\Delta t_k^2}{2} & \Delta t_k & 0 & 0 \\ 0 & 0 & \frac{\Delta t_k^3}{3} & \frac{\Delta t_k^2}{2} \\ 0 & 0 & \frac{\Delta t_k^2}{2} & \Delta t_k \end{pmatrix}, \tag{2.20}$$

where q depends on the process noise and can be changed dynamically during the tracking. When the target is moving faster or its direction is changing rapidly, q might be increased.

Error Resiliency

The ability of the designed tracking algorithm to overcome node failures and prediction errors is defined as error resiliency. The designed tracking system should be able to cope up with target lost instances. An innovative prediction error avoidance and error correction algorithm is proposed in [78]. If a sensor node notices the change of direction or velocity of the target before going to sleep, it recalculates the next predicted location of the moving object to avoid prediction errors. This step of making the tracking system survivable is referred as error avoidance. But if there is no time to change the next radio covered zone, error correction mechanism activates more neighbour nodes to include the target. In addition, the activated sensor node starts a timer and in case of failing to sense the target until the timer is expired, error correction mechanism initiates a recovery mechanism from the current detector. This recovery mechanism expands the radio covered area in the network to locate the target.

Sensor nodes are prone to failures due to lack of energy resources or physical damage. A survivable target tracking algorithm is proposed at [106]. This approach uses the network graph to select the next area of network to cover in order to locate the target. This network graph is a disconnected graph with no crossing edge. This graph consists of several enclosed polygonal regions, called Face. Members of a Face are called face neighbours. The proposed wake up mechanism has three steps. In the first step, the two nearest sensors to the predicted location of the target at the next tracking interval are wake up. If the algorithm fails to locate the target at the first step, all sensor nodes within a Face are waked up. The third step aims at locating the target by waking up all the sensor nodes in the neighbour Faces.

The recovery mechanisms are categorized into three different categories based on the nodes which initiates the recovery to locate the missed target in [100].

- **Source Recovery:** Source Recovery mechanism suggests to initiate the recovery from the last detector which was able to locate the target. This detector node wakes up all it neighbours to include the target.

- **Destination Recovery:** This recovery mechanism expands the radio covered area of the network from the next predicted node to locate the target. This node activates all its neighbours to find the mised target.

- **All Neighbour Recovery:** All Neighbour recovery activates both the neighbour nodes of the last detector node and the neighbour nodes of the predicted sensor to be the next detector.

All these recovery techniques activate all sensor nodes in the network if the activated neighbour nodes in the first step can not locate the target. The performance of these three recovery mechanisms are compared in [100]. This study reveals that *Destination Recovery* is the most energy-efficient recovery mechanism. Source recovery mechanism is the second rewarding technique in terms of energy efficiency. All Neighbour recovery requires the most energy compared to the other two recovery techniques.

2.6.6 Periodic PRediction-based Activation (PPRA)

Periodic Prediction-based Activation algorithms are identical to Prediction-based techniques but the nodes in the area of interest switch into sleep mode periodically and at the next time slot, the sensors in the predicted location of the target are awaked. PPRA algorithms decrease the energy consumption of the sensor nodes by sending the current detectors to sleep mode; even though the target might be in their visibility area.

Time interval of this switching between sleep and active modes can be adjusted in real time. H. Jamali et al. proposed to adjust the interval of tracking dynamically using a lookup table, which binds each average velocity of the target to a tracking interval, for achieving the least energy consumption [64]. During the run time, each node calculates the average velocity of the target and searches a pre-defined table to find the correspondent tracking time interval. Then, current node wakes up three nodes based on their closeness to the predicted location of the target. If one of these three nodes does not sense the target, more neighbour nodes will be awaked to locate the target. In addition, tracking interval might be tuned based on the current velocity of target or its average velocity collected over the time. More weights can be given to the current velocities. In [120], both tracking time interval and size of the cluster is adjusted in real time. Members of the cluster are selected based on their residual energy and the communication energy needed to report the location to the sink node. In this approach, each node must update its information about the residual energy of its neighbours which is an overhead on communication energy consumption of the system.

It is shown in [60] that a constant fixed tracking interval can be used for sensor activation in OTSNs to achieve minimal energy consumption. However, an innovative PPRA solution using variable adaptive tracking interval is suggested in [63]. This adaptive algorithm is designed to tackle the localization challenges of targets which move with variable accelerations. Tracking interval adapts the acceleration of the target to ensure the tracking quality. It is also proposed in [112] to adjust the tracking time interval in real time based on the required tracking accuracy and the cost of activating a sensor node for that tracking interval.

A variable sensing radius using RFID (Radio Frequency Identification) to track multiple targets is suggested in [43]. This technique computes a cost metric depending on the distance of the target to the sensor node and the required activation time of the sensor. Then, the sensor with the least cost is activated. This algorithm adjusts the sensing radius of the RFID reader based on the estimated location of the target to the centre of the sensor node.

2.7 Comparison of the Sensor Activation Techniques

The proposed sensor activation techniques are summarized in table 2.2. These algorithms use different target movement models to predict the next location of the target. In addition, some of these algorithms aim at increasing the prediction accuracy using Kalman filter or its extensions.

Table 2.2: Sensor Activation Techniques for Target Tracking in WSNs

Ref	Activation Technique	Prediction Model	Filter
[71]	CGA	-	-
[64]	PPRA	Spatial	-
[56]	PPRA	Spatio-Temporal	-
[114]	PRA	Spatio-Temporal	-
[115]	PRA	Spatial	-
[38]	RA	Spatial	-
[103]	PRA	Temporal	EKF
[100]	PPRA	Temporal	-
[119]	PRA	Temporal	STKF
[120]	PPRA	Spatial	-
[106]	PRA	Temporal	-
[46]	PRA	Temporal	-

2.8 Summary

In this chapter, the proposed algorithms for efficient target tracking using WSNs are categorized. We have also provided an analysis of the well-known sensor deployment, localization and sensor activation algorithms for target tracking using WSNs. In addition, it is shown that the consumed energy for sensing in a sensor node depends on the radius of the covered area by the sensor. Active sensing technologies provide the opportunity to adjust the sensing range of each sensor in real-time which can lead to deploy a more energy-efficient algorithm with better tracking quality. Thus, we take advantage of this novel capability of sensor nodes to further decrease the sensing energy consumption and prolong the network life time in VARSA.

Chapter 3

Methodology

3.1 Introduction

A crucial characteristic of an outstanding computer science research is reproducibility [47]. Reproducibility and repeatability of a research should be guaranteed by providing a comprehensive methodology of the research so that identical results can be regenerated in an independent experiment. Therefore, the results confidence is assured. This chapter attempts to provide a thorough modeling and experimental setup information. In addition, this methodology provides a fine configuration to analyze and evaluate various target tracking algorithms. This scientific method to come up with a solution to overcome energy efficiency challenges of target tracking using WSNs is derived from [47].

3.2 Research Model

Our research in target tracking using WSNs is targeted to enhance the energy efficiency of the tracking networks. We have developed several testable hypothesis as described in Section 3.4. Then, we inferred the consequences of the proposed approach. The suggested algorithm has been evaluated using extensive simulations. To provide a narrow confidence interval, fifty iterations of experiments has been conducted and the reported results are the average of recursive experiments. As the consistency is obtained from the experimental results, the proposed algorithm is proved to improve the energy efficiency of tracking using WSNs.

To come up with the examined hypothesis, a thorough knowledge of the current state of the art was needed. Hence, we have studied the available tracking algorithms to decrease the energy consumption for sensing during the tracking. Afterwards, we have formulated the tracking problem. Then, the tracking system and the target behaviour is modelled. Experimental results revealed that the proposed approach provides a more energy efficient tracking algorithm compared to two other promising tracking algorithms.

3.3 Problem Statement

We consider the problem of activating a subset of sensor nodes in the AoI for enhancing the sensing energy consumption; while the tracking quality is assured. Given N binary proximity sensors, $\{S_1, S_2, ..., S_N\}$ located as denoted by coordinates (x_i, y_i) in a two dimensional tracking area, the algorithm decides which sensor S_i to be in active mode and its correspondence sensing radius R_i over the time. Sensor nodes are homogeneous and they all have the same capabilities with a limited battery supply. There is also one sink node located at coordinates $(0, 0)$ with unlimited power supply which is responsible to gather the location information of the target and depict its trajectory.

The objective of this sensor activation algorithm is to prolong the network life time and decrease the target missing rate over time by keeping more sensors alive. We aim to prolong the network life time by decreasing the energy consumption for sensing. Hence, the algorithm decides which nodes to activate for how long and with what sensing radius in real time.

3.4 Research Hypothesis

In Chapter 2, a classification of the proposed approaches to address the problem of target tracking using WSNs is presented. In this chapter, we derive some hypothesis to be investigated in this thesis. We have analyzed, evaluated and compared these hypothesis using simulations in the following chapters. In our qualitative study, some of the sensor activation techniques has shown to outperform others. Hence, the following hypothesis are derived to be tested in our simulation study:

Hypothesis 1: *Predicting the location of the target after a predefined time interval and activating a cluster of sensors close to the predicted location of target while other sensor nodes are in sleep mode can considerably improve the energy efficiency of the tracking algorithm*

During testing this hypothesis, we have examined if activating the nodes in the AoI which is called PRediction-based Activation, *PRA*, can provide an energy efficient OTSN by sending redundant sensor nodes to sleep mode.

Hypothesis 2: *Alternating the detector nodes between sleep and active modes can reduce the sensing energy consumption and prolong the network lifetime while tracking quality is assured. Nodes in the AoI can also be switched between sleep and active modes without a high decrease in the localization accuracy*

This hypothesis represents that deploying a periodic activation mechanism to schedule the sensor nodes is an asset to decrease the sensing energy consumption. This hypothesis is to evaluate the Periodic PRediction-based Activation, *PPRA*, techniques. Tracking the target continuously is not required and the trajectory of the target can be derived providing the discrete informations about the location of the target.

Hypothesis 3: *Sending the appropriate sensing radius to wake up with to the next predicted detector in the network aims at decreasing the sensing energy consumption in target tracking using WSNs. The sensing radius can also be further reduced after the initial detection of target by tuning the sensing radius of sensor nodes in real time to improve the energy efficiency of target tracking using WSNs.*

Hypothesis 3 suggests that the the predicted sensor node does not require to be activated with its maximum sensing radius. Predicted location information of the target can help the predicted sensor to wake up with an appropriate sensing radius to locate the target. In addition, the radio covered area in the network can be further decreased to improve the energy efficiency of the network. The energy consumption enhancement can improve the localization accuracy over time as it provides more alive sensor nodes to locate the target as the network ages.

All the aforementioned hypothesis are evaluated using extensive simulation results taking into account a variety of evaluation metrics. These metrics represent the energy consumption and the localization accuracy of the tracking algorithms.

3.5 Modeling

Modeling is the process of extracting relevant features of a phenomenon in order to simplify the study of its behaviour [47]. The model should represent the measurable and observable characteristics of the phenomena and the consequences of any changes in the system. In our study, we need a model for the tracking network which consists of several individual sensor nodes and a model for the target movement.

3.5.1 Network Model

The problem of sensor activation for target tracking using wireless sensor networks has been evaluated in a homogeneous network in which all sensor nodes are identical. Sensor nodes are able to adjust their sensing radius in real time. In addition, we assume that each sensor node can communicate with other sensor nodes within its transmission range wirelessly. A binary output is provided by each sensor node regarding the presence of the target in its visibility area. This visibility area is a circular area with radius R_s surrounding by the sensor node which is radio covered by this sensor. We have considered ideal sensors which output $true$ when the target is closer than R_s to the sensor node and outputs $false$ otherwise. This sensing model is called Boolean disk model [80]. In reality, the probability that a target is sensed depends on the distance of the target to the sensor node and the environmental noise. In our analysis this probability is defined as:

$$p_{(d)} = \begin{cases} 1 & d <= R \\ 0 & d > R \end{cases} \tag{3.1}$$

where d is the distance of the target to the sensor node and R denotes the sensing radius of the sensor node.

Each sensor can be in one of these three modes: active, communication active and sleep mode. In active mode, both the sensing and communication modules of the sensor is active and it is able to locate the target in its visibility and send a report containing location information of the target to the sink node. Communication active nodes are collaborating in forwarding the data to the sink node; however, their sensing module is inactive and they cannot sense the target in their visibility area. Sleep nodes are completely deactivated and they only listen to their low power radio receiver. These sleep nodes might be awaked using a low power paging radio channel by other nodes in their transmission range as devised in [58]. It is worth to mention that idle nodes in the network consume almost as much energy as active nodes for sensing but the communication energy consumption is less. The deployed sensor nodes are aware of their locations using a GPS-less self positioning method as proposed in [44]. Neighbour location informations are aggregated at each sensor node at the beginning of the tracking time to augment the routing performance.

In our experimental studies, we have evaluated VARSA and two other well-known algorithms considering 600 sensor nodes randomly placed in an area of $200m \times 200m$. Then, we have also evaluated VARSA, PRA and PPRA for 200, 400 and 800 nodes to examine the effect of the density of the deployed sensor nodes in the performance of the algorithm. Sensor nodes have 0.15 J energy resources at the deployment time and the communication energy consumption model is implemented according on the model used in [120]. Each sensor node is able to communicate with its neighbours within 25 meters. Transmission range of each sensor node is adjustable [14]; however, we have considered the a fix transmission range for all the sensor nodes. Visibility area of a sensor node is considered as a circle with maximum radius of 15 meters which can be adjusted in real time. Table 3.1 outlines the network settings. The power consumption of idle nodes is $40microW$; however, it does not consist of the consumed energy for sensing which is based on its sensing radius and is $50microW$ when the node is scheduled for sensing with its maximum feasible sensing radius.

The sink node, where all the sensed data is aggregated at and the trajectory of target is depicted, is always located at coordinates $(0, 0)$. There is no limitation of the energy resources for the sink node and its energy is not calculated as a sensor node in the simulation experiments.

Table 3.1: Network Settings

Parameters	Values
Radio Propagation Model	TwoRayGround
MAC Layer	IEEE 802.11
Antenna	OmniAntenna
Surveillance Area	200×200
Number of Nodes	200, 400, 600, 800
Initial Energy	0.15 J
Maximum Sensing Power	50 microW
Receiving Power	50 microW
Transmission Power	100 microW
Idle Power	40 microW
Max Sensing Radius	15
Transmission Range	25

We have implemented the greedy routing algorithm for communication between different sensor nodes including sending the location report to the server node; however, the proposed approach is independent of the routing algorithm and it can work properly using any communication protocol.

3.5.2 Target Mobility Model

A single target, which is a deer, is being tracked. We use Random Waypoint mobility model for the target; however, we have manipulated the bonnmotion-2.1 mobility generator to best represent the deer movement as described in [37].

In this thesis, we focus on enhancing wildlife monitoring applications using WSNs. Thus, the proposed algorithm is evaluated by tracking a single target with deer movement behaviours. We have applied the deer movement habitudes in RandomWaypoint movement in Bonnmotion-2.1 mobility generator code based on the the information acquired from [37].

We have also evaluated VARSA, PRA and PPRA using a Pursue movement model to investigate the effect of the target movement model in the performance of these algorithms. The Pursue can simulate a deer when it is running from a hunter or a criminal running from the cops [35].

Bonnmotion Mobility Generator

Bonnmotion is a Java based mobility generator which can be used to create and analyze several mobility scenarios. The generated scenarios can be converted to an OTcl script which is compatible with ns-2 requirements. Some of the scenarios which Bonnmotion can generate include the Random Waypoint model, Random Walk model, Gauss-Markov model, Manhattan Grid model, Reference Point Group Mobility model, Disaster Area model, Pursue model and Random Street model. We have selected the target mobility model to be Random Waypoint as discussed in Section 3.5.2. Then, the algorithms have been also evaluated using Pursue movement model.

Random Waypoint Model

Random Waypoint is a mobility model in which the target picks its destination after some pause times and moves toward it [7]. At a specific time after the start of the simulation, skipped time, the target picks a destination and moves toward it with velocity v_t where $v_{min} < v_t < v_{max}$. When the target arrives at a destination, it remains at this location for t_{pause} where $t_{min} < t_{pause} < t_{max}$. Then, the target selects the next waypoint. There are some attraction points in the simulation area and the probability of deciding to move to an attraction location is higher than moving to any other location in the simulation area. This model prevents the target to stay in the middle of the simulation area for most of the times. The number of these attraction points can vary based on the simulation requirements.

In our simulations, we have modelled the target mobility with a Random Waypoint model. The duration of the simulation is 10 days and the skipped time after the start of the simulation is set to be 30 seconds. This initial skip time enables us to start tracking the target when topology is stable and the system has passed the initialization step. The maximum velocity of a deer is $20m/s$ and there is one pause time of 4.5 hours in each day for sleeping in mobility behaviours of the target. Both horizontal and Vertical, 2D, movements are enabled at the same time. Target mobility parameters are summarized in Table 3.2.

Table 3.2: Target RandomWaypoint Mobility Parameters

Parameters	Values
Mobility Model	Random Waypoint
Mobility Area	200×200
Minimum Velocity	18 m/s
Maximum Velocity	20 m/s
Maximum Pause Time	120 s
Minimum Pause time	0 s
Mobility Duration	10 days

Pursue Mobility Model

The main usage of Pursue mobility model is to simulate some mobile nodes which are pursuing a single target. However, we used only the mobility information of the target node in this study. Pursue mobility model can effectively show the mobility of cops pursuing a criminal [35] or a deer escaping from hunters. The location update equation of Pursue mobility model depends on the last location of the target and a random vector and it does not depend on any area of interest in the tracking field. Hence, the target is trying to scape without any pause time during the movement and the probability of moving the target to any direction is the same. However, the amount of randomness of the movements are limited to best represent the pursuing situation. The duration of the simulation is considered to be an hour since the pursuing situation usually does not last long. The Pursue mobility parameters are summarized in Table3.3.

Table 3.3: Target Pursue Mobility Parameters

Parameters	Values
Mobility Model	Pursue
Mobility Area	200×200
Minimum Velocity	18 m/s
Maximum Velocity	20 m/s
Pause Time	0 s
Mobility Duration	1 hour

Evaluation of VARSA using Pursue mobility does not only reveal its performance for wildlife monitoring but it also determine its applications for high mobility patterns without any pause time such as pursuing.

3.6 Experimental Environment

Computer simulation is a tool to study the phenomena which can not be evaluated in a real testbed due to the unfeasibility or high cost of deploying the required testbed. Thus, we have conducted extensive simulation experiments to study and compare the practical performance of target tracking algorithms for WSNs. These experiments are used to evaluate both the current available algorithms and the novel tracking algorithm proposed in this thesis.

3.6.1 Simulation Software

Network simulation is essential to enhance the computer networks as new network protocols and models have to be evaluated and studied using simulations before deployed. There are several simulation tools to study the behaviour of the network such as Network Simulator, OPNET, OMNet++, SWiMNet [17] [18] and GloSim. Network Simulator (ns) is a discrete event object oriented simulator which is used to evaluate computer networks. ns is written in C++ with an OTcl interpreter. Several versions of ns, categorized as ns-1, ns-2 and ns-3, have been released.

We have implemented the proposed algorithm in Network Simulator 2 (ns-2.35) which is tested and validated in Paradise Research Laboratory. In addition, it is one of the most widely used network simulators. ns-2 supports the simulation of wired and wireless networks. The nodes can be static and mobile. There are also comprehensive documentation and tutorials for ns-2 [52]. The simulation script and the setup is to be written using OTcl. OTcl is an script language which is used to define the network topology, schedule the events and configure the protocol. The topology consists of the nodes positions, interface configurations and mobility of the nodes. In addition, communication channel, radio range, number of nodes, simulation time, initial energy of nodes and the model for energy consumption is defined in the OTcl script. New objects are initiated in the interpreter. The core of ns-2 is in C++. The routing protocol and agents are written in C++ and is assigned to nodes in the OTcl script. Figure 3.1 shows how the OTcl interpreter interacts with the C++ libraries to create different objects and run the implemented protocol. The simulation results are carried out during a simulation run. In summary, C++ is used to provide the researcher the ability to manipulate bytes, packet headers and implement algorithms with a short run-time [52]. C++ is fast but changing the code needs more time. On the other hand, OTcl is slower but can be changed faster and easier. OTcl is responsible for configurations which is required once at the start of the simulation, hence, the run-time is not crucial. A thorough manual for ns-2 can be found in [52].

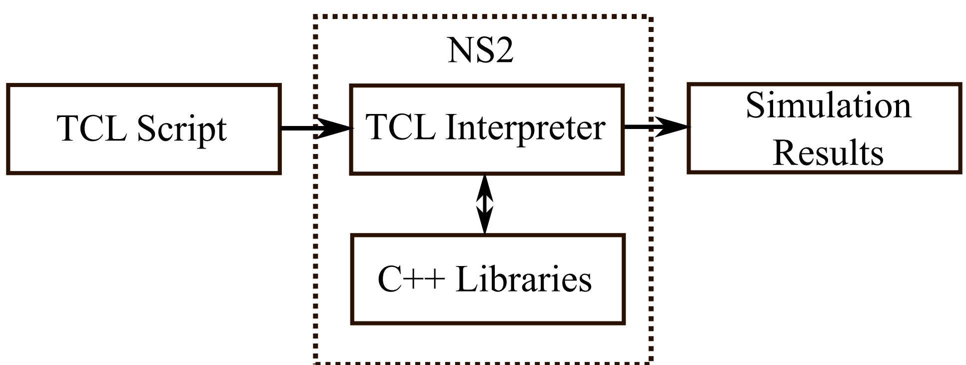

Figure 3.1: The Interaction of OTcl Script and C++ Code in ns-2

Chapter 4

Prediction-based VAriable Radius Sensor Activation Using Wireless Sensor Networks (VARSA)

This chapter proposes a novel target tracking algorithm using WSNs. A robust Object Tracking Sensor Network (OTSN) algorithm should address the problem of energy limitation of a sensor node. In addition, tracking quality is to be assured. Hence, VARSA attempts to resolve the conflict of interest between energy consumption and localization accuracy by deducting the extra radio covered area where the target is not located while assuring the coverage of the AoI. This compromise is achieved by tuning the sensing radius of a sensor node in real-time. It is also important to design a survivable algorithm which is able to cope with sensor failures. Therefore, a recovery mechanism has been devised for our tracking algorithm.

This chapter is structured as follows. The sensor deployment strategy is discussed. Then, VARSA has been elaborated precisely in Section 4. This section explains different states of the algorithm along with the movement model assumed for the target and the prediction mechanism.

4.1 Sensor Deployment

It has been shown in [93] that deploying the sensor nodes randomly in the network field is comparable to more sophisticated sensor deployment algorithms such as square grid and pattern based Tri-Hexagon Tiling (THT) node deployment. Coverage, energy consumption and the delay of these sensor deployment algorithms has been evaluated in [93]. THT outperforms the other two node deployment strategies in terms of energy consumption and delay and square grid provides a better coverage of the network field. However, random deployment is still preferable to avoid the planning overhead [93]. As discussed in Section 2.2, grid-based and THT sensor deployment techniques are not practical for target tracking applications. Hence, we have deployed the sensor nodes with a uniform random probability distribution in the tracking field.

4.2 Algorithm Description

The proposed VAriable Radius Sensor Activation (VARSA) mechanism for target tracking using WSNs has been devised in four states: initialization, tracking, radius adjustment and recovery. The proposed tracking system is in initialization state at a predefined time just after the start of tracking, called initialization time. Tracking state is initiated after the initialization time in which only the selected node to detect the target is active. When a sensor detects the target, it adjusts its visibility area by trying smaller radius of detection to save more energy. This state is called radius adjustment. Even though prediction error should be avoided; a recovery mechanism is needed for the worst case scenario when the target is lost. Recovery state tackles the error resiliency issues of VARSA. These states are elaborated in more details in the following sections.

4.2.1 Initialization State

Initialization state begins after deploying the sensor nodes in the tracking area. All the sensor nodes acquire the address and the location coordinates of the sensor nodes in their transmission range. These information are listed in a lookup table in the sensor node to be used for routing or recovery purposes. Each node preserves a lookup table containing all its neighbour node addresses and location coordinates. These information helps the implemented greedy routing protocol to aggregate the target location data at a sink node. As the sink node is always located at coordinates $(0, 0)$, each sensor node can effectively relay the packets in a short path having the location of its neighbours.

All the sensor nodes are in active mode to find the initial location of the target in initialization state. This sink node uses data fusion techniques to draw the trajectory of the target based on the received information over time. This state lasts until the end of the initialization time when all the sensor nodes except the closest sensor to the estimated location of the target are sent to sleep. During this step, each detector reports the location of the target as its coordinates to the sink node if it detects the target and the sink node estimates the location of the target using the Centriod localization technique. Centriod localization technique estimates the location of the target at time t_0 as the average of the location coordinates of the sensor nodes which detected the target at the same time.

4.2.2 Tracking State

In tracking state, all sensors except the closest sensor to the predicted location of the target are sent to sleep. At the first tracking interval after initialization, the closest detector to the predicted location of the target is waked up to track the target. Current node, which detected the target at the current time, predicts the next location of the target using a linear model and wakes up the closest node to the predicted location of the target at the next tracking interval. Current sensor node alternates between sleep and awake even if it predicts the target to remain at its visibility for the next tracking interval. After waking the next predicted node up for tracking, current detector node sends the last estimated location coordinates of the target to the new waked up sensor. Current detector uses the last location information of the target and its current location to predict the next location of the target using a linear model.

Current node also sends the desirable sensing radius to wake up with to the next sensor node. This desirable sensing radius is based on the closeness of the predicted location of target to the selected sensor node. Reducing the visibility area of a sensor node at the time of activation can save a considerable amount of energy as the energy is proportional to the cube of the sensing radius. See Equation 5.1 for more details.

4.2.3 Sensing Radius Adjustment State

We have suggested to initiate a sensing radius adjustment technique after detecting the target. Sensing radius adjustment aims at decreasing the sensing energy consumption by tuning the sensing radius in real time based on the closeness of the target to the sensor node while maintaing the target observability. The proposed approach is performed in three steps. At the beginning, the radius of the covered area around the sensor is divided by ω . At the next step, if the target is still in the visibility area of the sensor node, the sensing radius is decreased again dividing by ω. After each of the sensing radius

Algorithm 1: Sensing Radius Adjustment().

```
begin
    switch Radius Adjustment Step Status do
        case First Step
            Rs_new ← Rs_current / w
        end
        case Second Step
            if Target is detected then
                Rs_new ← Rs_current / w
            else
                Rs_new ← Rs_current + (Rs_max − Rs_current) / w
            end
        end
        case Third Step
            if Target is detected then
                Report the location coordinates to server
            else
                Rs_new ← Rs_current + (Rs_max − Rs_current) / w
                Report the location coordinates to server
            end
        end
    endsw
end
```

deduction, the current sensor node tries to locate the target again and if the target is not detected, sensing radius is increased using Equation 4.1.

$$Rs_{new} = Rs_{current} + \frac{Rs_{max} - Rs_{current}}{\omega} \tag{4.1}$$

where the new calculated sensing range is denoted by Rs_{new} and current sensing radius of the sensor node is $Rs_{current}$ and Rs_{max} represents the maximum sensing radius which can be covered by a sensor node. We denote ω as the sensing radius adjustment rate. After adjusting the sensing radius, the algorithm reports the coordinates of the current detector node as the estimated coordinates of the target to the sink node. If the target is not located in this step, VARSA increases the radio covered area in the network. The proposed real time sensing radius tuning is illustrated in Algorithm 1.

An example can clarify the proposed radius adjustment algorithm further. In Figure 4.1, when the detector detects the target, it divides its visibility range by ω. If it detects the target again, the current sensor decreases its visibility area again, dividing it by ω. After a short interval, the target is not detected anymore by this sensor node; hence, the detector increase its visibility area using Equation 4.1 to include the target and reports the target location to the server.

After this sensing radius adjustment, the sensing area of the detector is much less than the initial visibility area which saves a considerable amount of energy as the time passes. The energy efficiency can be even more in wildlife monitoring considering movement behaviours of animals which might remain in a location for a long time. Current studies in [37] reveal that a deer sleeps 4.5 hours on average during a day. Preventing the sensor nodes from covering that area aims at reducing the probability of depleting their energy. Decreasing the consumed energy for sensing during these hours can improve the missing rate considerably over time. As the rest location of the animal gets closer to the sensor node, more energy can be saved during tracking.

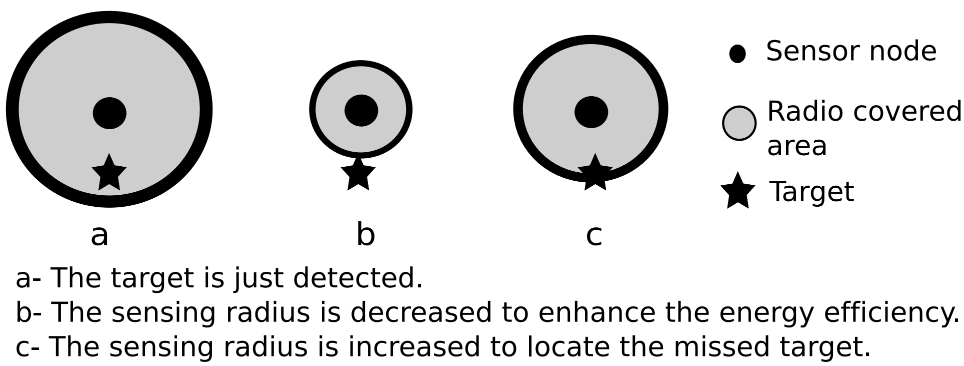

Figure 4.1: An Example for Sensing Radius Adjustment

After radius adjustment state, the algorithm state is again changed to tracking. Thus, current detector predicts the next location of the target and wakes up the closest sensor among all sensor nodes covering the predicted location of the target. Current detector sends the advisable sensing radius, which the next detector is able to wake up with in order to locate the target efficiently, and the last estimated location of the target to the awaked sensor node. Afterwards, current detector switches to sleep mode. The proposed tracking algorithm keeps tracking the target during the tracking time by predicting next location of target, activating the closest node to that location with an appropriate sensing radius and adjusting its sensing radius in real-time.

4.2.4 Recovery State

A rewarding tracking algorithm should be able to overcome node failures and prediction errors. A recovery mechanism is initiated in case of failing to detect the target after a predefined *recovery time* after the current node wakes up as the next predicted sensor node. The recovery process is either inaugurated by the predicted sensor node or by the server node if it does not receive a report within a predefined time. We have deployed the *destination recovery mechanism* as it is proved to outperform source recovery and all neighbour mechanisms in [100]. So, this is the next predicted node‘s responsibility to initiate the recovery; however, the recovery can also be initiated from the sink node to ensure the survivability and error resiliency of the algorithm.

Our recovery mechanism has been designed in four steps. In the full sensing radius recovery state, the current sensor node aims at locating the target by increasing its visibility area to its maximum. Full sensing radius recovery helps the network to locate the target without waking more nodes up and it also improves the energy efficiency of the algorithm. All the neighbour nodes in the transmission range of the current predicted detector are waked up in the first step recovery. If the network was not able to detect the target, all these neighbour nodes activate the nodes within their transmission range with their maximum sensing radius to further increase the radio covered area in the network in the second step recovery. The established lookup table at the initialization step aims at recognizing the neighbour nodes in order to wake them up using a low power communication channel. In the worst case, all sensor nodes are activated in the full recovery step to locate the target. Activating the entire sensor nodes ensure that the target is located unless the current location of the target is not covered by any nodes due to node failures or imperfect deployment of the sensor nodes.

4.2.5 VARSA State Transition

To clarify different states of VARSA, we have provided a transition diagram of this algorithm in Figure 4.2. This figure shows in which conditions, the state of VARSA changes. As the tracking starts, the algorithm is in initialization state until $t_{initialization}$. Then, VARSA switches to the tracking state when only one node in the AoI is responsible for tracking the target and the rest of the sensor nodes are in sleep mode. If the target is detected, the algorithm goes to the sensing radius adjustment state where the algorithm tries to locate the target with a smaller radio covered area. After adjusting the sensing radius of the sensor node, the algorithm returns to the tracking state and continues tracking the target.

If the target is lost during the tracking state and the awaked sensor node could not locate the target, the algorithm starts the recovery process. The first state of the recovery is full radius recovery when the algorithm aims at locating the target by expanding its sensing radius to its maximum. If the target is detected, the algorithm switches to sensing radius adjustment state and if the target was not detected, the first step recovery starts. All the one hop neighbour nodes of the predicted sensor to detect the target are waked up in first step recovery to locate the target. If the target is detected, VARSA starts adjusting the sensing radius and if the sensor was not able to detect the target, the second step recovery is initiated. In this step, all the one hop neighbours of the sensor nodes which were waked up in the first recovery step are awaked to detect the target. In case the target is detected, the algorithm goes to sensing radius adjustment and the sensors which did not detect the target are sent to sleep. In the last step of recovery, the entire sensor nodes in the network are waked up to locate the target. Once the target is detected, the algorithm transits to the sensing radius adjustment state.

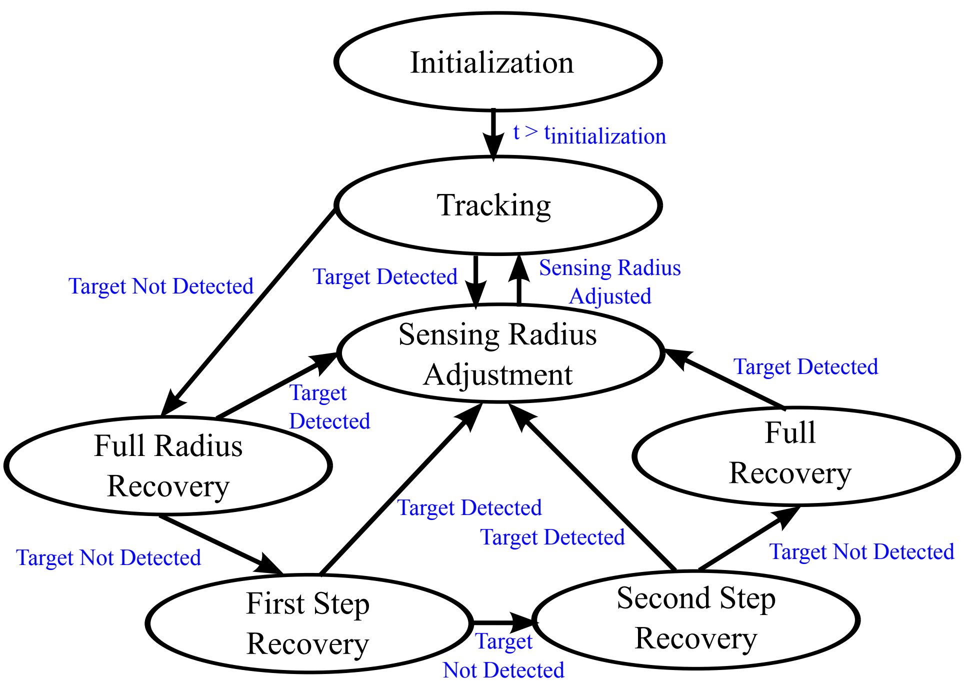

Figure 4.2: State Transition Diagram of VARSA

4.2.6 Prediction method and movement model

Several movement models have been elaborated in Section 2.6.5. Linear model of the target movement assumes that the target velocity and its direction is constant in any short interval of times. While linear movement model predicts next location of the target using only the last estimated velocity of the target and its coordinates, average and exponential movement models require the target velocity information to be transmitted toward the next sensor nodes at the time of activation to provide them enough information to estimate the current velocity of the target. Hence, the energy consumption of linear model is much less than the other two techniques. In the other hand, absolute models are application specific and are not reliable for all targets as some targets might not follow the same pattern. In addition, the prediction inaccuracy which might lead to the problem of losing the target can be solved by a recovery mechanism which has been elaborated lately.

In this study, assuming that the target velocity is constant during a short time interval, we used a linear model to predict the location of the target after each interval to simplify the calculations and to avoid unnecessary processes at the sensor nodes. In this model, velocity of the target is assumed to be constant in a very short period of time and it can be calculated using 4.2 and 4.3.

$$V_{x,t_i} = \frac{x_{t_i} - x_{t_{i-1}}}{t_i - t_{i-1}} \tag{4.2}$$

$$V_{y,t_i} = \frac{y_{t_i} - y_{t_{i-1}}}{t_i - t_{i-1}} \tag{4.3}$$

Using linear estimation, next location of target can be estimated as:

$$x_{t_{i+1}} = V_{x,t_i} \times (t_{i+1} - t_i) + x_{t_i} \tag{4.4}$$

$$y_{t_{i+1}} = V_{y,t_i} \times (t_{i+1} - t_i) + y_{t_i} \tag{4.5}$$

We used spatio-temporal prediction technique. However, we only take the information from current detector and last detector since this technique leads to a more accurate prediction using less transferred data which can enhance the energy efficiency. One node is activated at each interval of tracking and network resiliency is assured through the recovery mechanism. As far as the tracking interval time is short enough, this model provides acceptable prediction accuracy considering the limited processing abilities of tiny sensor nodes.

4.2.7 Localization

Lots of localization techniques for binary proximity tracking sensor networks are proposed in recent studies. We have considered the average locations of the sensors, which has detected the target in each interval as the location of the target. The most rewarding advantage of this technique is its simplicity, which leads to increasing the speed of running a tracking algorithm and provides a better live performance. In addition, the processor's energy consumption is less. Let the set of n sensors S_i that detect the target at time t_0 to be $Detectors(t_0)$, Centriod localization estimates the location coordinates of the target at time t_0 as:

$$x_{target,t_0} = \frac{\sum_{i \in Detectors(t_0)} x_i}{n}, \tag{4.6}$$

$$y_{target,t_0} = \frac{\sum_{i \in Detectors(t_0)} y_i}{n}, \tag{4.7}$$

where (x_i, y_i) denotes the location coordinates of detector S_i. In this localization technique, there is no constraint in sensor deployment. While some other algorithms need a specific number of nodes to cover each place in order to localize the target, one proximity sensor node can approximate the location of the target using this technique.

During the initialization and recovery state, more than one node is activated to locate the target; hence, Equation 4.6 and Equation 4.7 are used to estimate the location of the target. However, only one node is activated in tracking state and the location of the target is estimated as the location of this sensor if it can detect the target. In this approach, the bound on the location error is the sensing radius of the sensor. In applications of target tracking using binary proximity sensors, the accuracy of the localization technique is not the main objective of the application; therefore, this error is acceptable for tracking. The main aim of binary proximity tracking sensor networks is to find the area of the presence of the target and not the exact location of the target. For instance, the closeness of an animal to a dangerous area such as a road is crucial in wildlife monitoring applications; while, in military applications, the exact coordinates of the target is important. Hence, binary proximity sensors are not suitable for the applications which require the exact position of the target.

4.2.8 Routing

The information derived from the sensor nodes should be aggregated at a sink node to estimate the trajectory of the target. There are several routing protocols in the literature to aggregate the sensed data at a sink node [48] [106] [13] [15] [28]. In VARSA, the location reports are sent to the sink node using a traditional geographical routing called greedy routing. This routing protocol does not affect the performance of the proposed technique to improve the sensing energy consumption in the network; however, an efficient routing protocol can decrease the communication energy consumption.

Greedy routing protocol is a location-based routing which aims at forwarding the packet in the right direction to the destination node [55]. Each sensor node forwards the packet to the closest node to the destination in its transmission range. In OTSN, sensor nodes are always aware of the location of the sink node. In addition, the location of the sensor nodes in the transmission range of a sensor is gathered at the initialization state and can be used during the routing. As discussed in Section 4.2.1, each sensor node broadcasts a location request packet at the initialization state. Each node which receives this request replies the source node with a packet containing its address and location. This sensor also updates its routing table with the received information from the neighbouring nodes. In this way, all the sensor nodes create a routing table containing the location coordinates and address of their neighbour sensors during the initialization time.

A thorough geometric and simulation study in [113] has shown that the greedy routing protocol can effectively route the location reports through a short path. The intrinsic distributed characteristic of greedy routing protocol makes it suitable for large scale sensor networks. In addition, this geographical routing algorithm aims at forwarding the packets in a short path to guarantee the energy efficiency of the communication protocol. Hence, we take advantage of this scalable and efficient routing protocol to aggregate the location reports at the server node. Taking to account the limited computation capabilities and memory resources of a sensor node, greedy protocol is a practical routing protocol for OTSNs since it is working in a distributed fashion and it neither needs to maintain large routing tables nor perform complex computations.

4.2.9 Summary

In this chapter, we have proposed a novel sensor activation algorithm to decrease the sensing energy consumption of the network. The proposed algorithm, VARSA, takes advantage of the ability of the sensor nodes to adjust their sensing range in real-time. In VARSA, the current node predicts the new location of the target and wakes up the next predicted sensor node to detect the target. It also sends the appropriate sensing radius which the next sensor node requires to wake up with to the next predicted sensor. This advisable sensing radius is based on the closeness of the predicted location of the target and the location of the next predicted sensor node to detect the target. VARSA also adjusts the sensing radius after the initial detection of the target to further decrease the sensing energy consumption of the network. We argue that this reduction of the energy consumption provides more alive sensor nodes to locate the target as the network ages. Hence, VARSA does not only prolong the life time of the network but it also decreases the missing rate over time.

Chapter 5

Simulation Studies

In this chapter, we have evaluated VARSA using extensive simulations. The tracking parameters including the tracking time interval, duty cycle of tracking and sensing radius adjustment rate (ω) are tuned in this chapter. Then, we have compared VARSA with other two promising tracking algorithms PRA and PPRA. Afterwards, the effect of changing the density of the deployed sensor nodes is studied by altering the number of deployed sensor nodes in the tracking area. We have considered two movement scenarios for the target movement. We have evaluated VARSA using a RandomWay point mobility model for target to examine and compare its performance for wildlife monitoring applications. Then, VARSA is evaluated using Pursue mobility model for the target to demonstrate its performance for high-mobility applications.

The rest of this chapter is organized as follows. First, we have provided the simulation setups. Then, the evaluation metrics are elaborated. Afterwards, the simulation study is given and the algorithms comparisons are provided. In addition, the effect of the node density and the movement model is investigated.

5.1 Simulation Setup

There are some user defined parameters for the simulation which could be adjusted based on the requirements of application, while network settings are unalterable after the network deployment. We have examined several intervals for tracking and the best interval time of tracking to prolong the network lifetime and achieve a less missing rate was 0.2 second as shown in Section 5.3.1. This tracking interval is selected after parameter tuning experiments to achieve the best quality of tracking and energy efficiency. We have also evaluated the possible choices for the duty cycle of tracking and we have shown that 20 percent is the best duty cycle as shown in Section 5.3.1. Location reports are sent to the sink node after each 5 consecutive intervals or when the current detector has not detected the target in the previous tracking interval. If the target were not detected in the first 30 percent of each interval, the recovery mechanism is initiated. In addition, we have conducted several experiments to find the best ω in the radius adjustment algorithm. Sensing radius adjustment rate, ω, has been shown to be 4 to achieve the best quality of tracking and network lifetime as elaborated in Section 5.3.1. User defined parameters are summarized in table 5.1.

Table 5.1: User Defined Parameters

Parameters	Values
Tracking Interval	0.2 s
Tracking Duty Cycle	20 %
Report Interval	5 s
Recovery Timer	30 %
Initial Time	0.02 s
ω	4

5.2 Evaluation Metrics

As energy limitation is the most challenging constraint of WSNs, we have evaluated the proposed algorithm measuring the network lifetime, consumed energy for sensing and the remaining energy after some intervals of time. In the other hand, we have compared the missing rate of all algorithms to see their accomplished tracking quality.

5.2.1 Sensing Energy Consumption

Recent advances in active sensing technologies provided the sensors the ability to adjust their sensing range in real-time. Some of these sensors are introduced in [76]. The energy consumption of these sensor nodes depends on their sensing coverage area. Even though the exact energy consumption as a function of sensing radius depends on the type of the sensor and the applied technology for manufacturing the sensor node; an approximate model is given at [9] as formulated in Equation 5.1.

$$SensingEnergyCosumption = a \times R_s^c + b, \tag{5.1}$$

where parameter R_s represents the sensing radius of the sensor and a, b and c are constants depended on the applied technology at the time of manufacturing the sensor node and $2 < c < 4$.

5.2.2 Remaining Energy

The remaining energy in the network after different intervals of time can effectively represent the energy efficiency of the target tracking algorithms as it shows both the sensing and communication energy consumption. Besides the sensing energy, a considerable amount of energy is needed for packet transmission and receiving. This energy could be consumed for sensing, transmission, computation and switching between sleep and active modes. We have taken the total remaining energy in the network into consideration, which shows the total energy consumed for both communication and sensing even though the proposed approach is for decreasing the consumed energy for sensing. The communication energy consumption is also related to the transmission radius; however, transmission range is always constant in our algorithm. we consider the energy cost model from [120].

5.2.3 Lifetime

There are several definitions of lifetime for WSNs. Network life time is defined in [40] as "Network lifetime is the time span from the deployment to the instant when the network is considered non-functional. When a network should be considered non-functional is, however, application specific". A pessimistic definition of the network lifetime is the time that the first node depletes its energy and fails to continue its operations [86]. In our simulation study, we have considered the lifetime of the network as the time that 75 percent of the sensor nodes are dead as suggested in [120].

5.2.4 Number of Dead Nodes

The deployed sensor nodes deplete their energy during the tracking depending on the time they were active for sensing and/or communication. Some of the energy might also be used for switching between sleep and active modes. Number of dead sensors represents the possibility of network holes, where no sensor can cover that area, which affects the tracking quality of the algorithm. In addition, it reveals the ability of the algorithm to locate the target more precisely and efficiently.

5.2.5 Missing Rate

Evaluating a tracking algorithm in terms of missing rate represents the accomplished tracking quality. Missing rate is defined as the sum of tracking intervals that the target could not be located without increasing the radio covered area in the network divided by total number of tracking intervals during the last tracking segment. Each tracking segment is a duration of 12 hours. After each tracking segment, the current state of the system and all parameters are recorded to evaluate the tracking algorithm.

5.2.6 Recovery Rate

Recovery rate is denoted as the proportion of the number of the performed recovery to the number of tracking intervals. Recovery rate reveals the demand of an algorithm to wake up more sensors to assure the quality of tracking and can be drawn for different recovery steps. The radio covered area of the network is increased in each recovery step which leads to increasing energy consumption of the network. Recovery rate also represents the movement prediction accuracy and tracking quality of the network. The recovery rate can be recorded for each recovery step based on the applied recovery mechanism.

5.3 Simulation Study

We run the proposed algorithm for 10 days of simulation time. The results are the average of 50 iterations to provide 95% confidence interval. The proposed algorithm is compared with other two promising algorithms which are PRediction-based Activation (PRA) and Periodic PRediction-based Activation (PPRA).

The proposed tracking algorithm has been evaluated through extensive simulation using Network Simulator 2 (NS-2.35). To better understand the target tracking applications, we draw actual trajectory of the target and the estimated trajectory at the sink node using VARSA tracking algorithm for the first 12 hours in Figure 5.1. In addition, we have shown the estimated and actual location of the target for the first hour and ten minutes of the simulation in Figures 5.2 and 5.3. In these diagrams, it is shown how the estimated trajectory of the target at the sink node is close to the actual geographical location of the target. The parameters that affect the performance of the algorithm must be tuned to improve the performance of a novel algorithm. We run several simulations to find a suitable value for each parameter in the following section.

5.3.1 Parameter Tuning

In this section, we have found a suitable value for tracking time interval and the duty cycle of the algorithm. Then, we also examined several values of ω in the radius adjustment state of the algorithm to find the most rewarding value.

Tracking Interval Tuning

We run several simulations to find the appropriate tracking interval for the proposed algorithm. We have changed the tracking interval of the tracking algorithm in the range of $[0.06, 5]$ seconds. We have examined 13 different tracking intervals to find the best tracking interval between these numbers which suits the tracking algorithm.

The main aim of the proposed algorithm is to decrease the energy consumption for sensing in order to prolong the network lifetime while ensuring the tracking quality. Hence, we have studied the energy consumption and localization accuracy of the proposed activation technique.

Figure 5.5 shows the remaining energy of the network over time. As the tracking interval is increased from 0.06, the remaining energy of the network is also increased until the TI reaches 0.2. As the TI was increased to more than 0.2, the remaining energy of the system is less over time. So, this figure shows that the 0.2 is the best value for TI between the examined values.

The consumed energy for sensing shows to be in reverse of the remaining energy of the network as shown in Figure 5.6. Sensing energy consumption is decreased after we increased the TI from 0.06 to 0.2. It differs from the expectation that by increasing the TI, energy consumption does not change as the duty cycle of the system is the same. If the target is not missed, the active time of the network is the same. However, the occasions that the target is missed is more and the network needs to activate more sensor nodes in the recovery state to locate the target. By increasing the TI to 0.2, the tracking quality of the network was good enough to avoid energy consumptions in several recovery states. Thus, the energy consumption is increased after we increased the TI more than

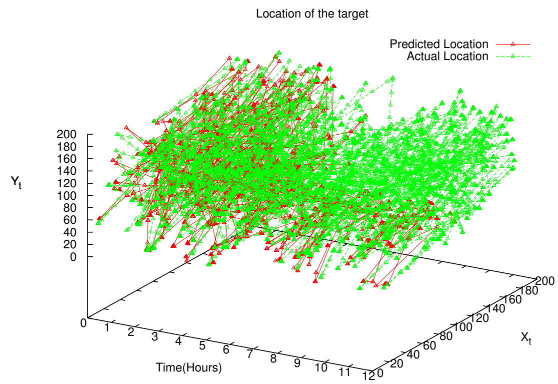

Figure 5.1: The estimated and actual location of the target in the first 12 hours

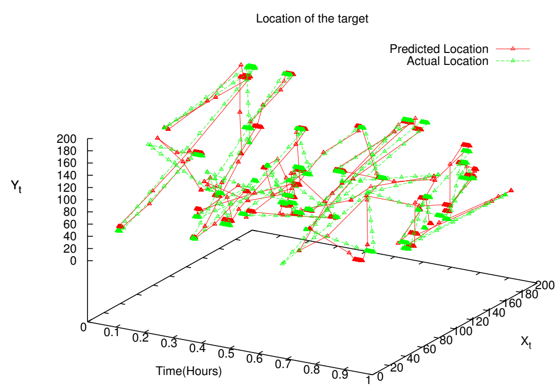

Figure 5.2: The estimated and actual location of the target in the first 1 hour

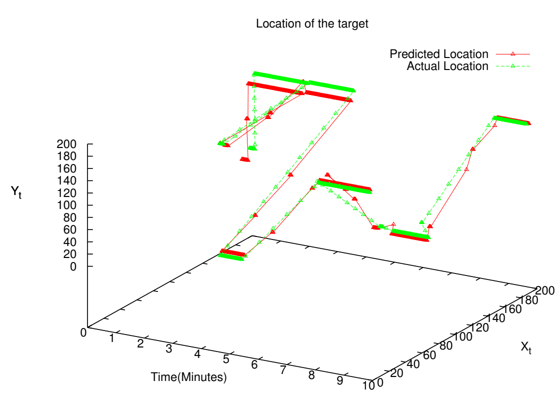

Figure 5.3: The estimated and actual location of the target in the first 10 minutes

Figure 5.4: The estimated and actual location of the target

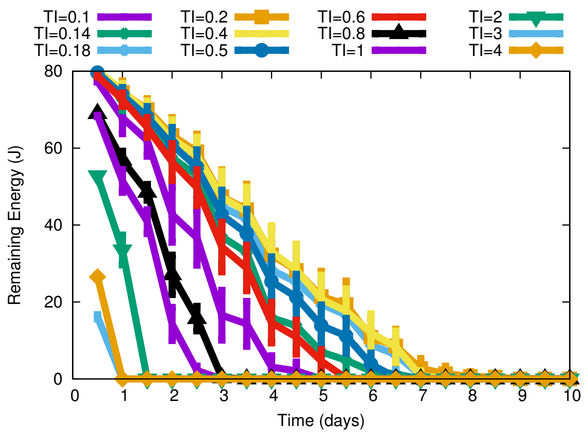

Figure 5.5: Total Remaining Energy

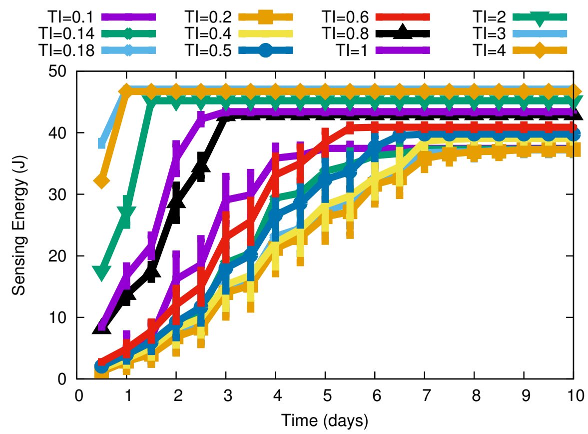

Figure 5.6: Sensing Energy Consumption

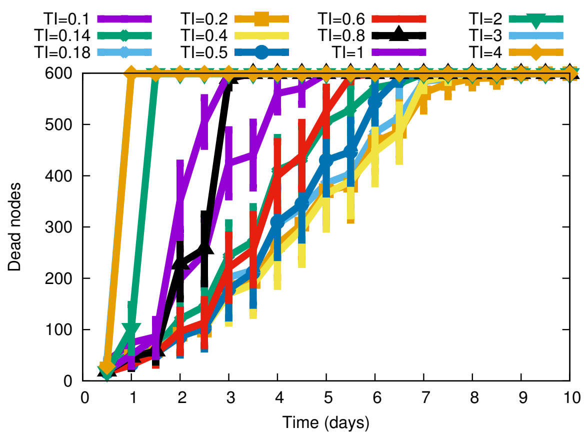

Figure 5.7: Number of Dead Nodes

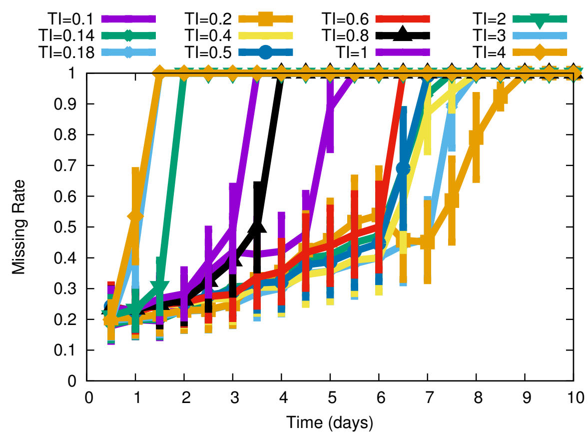

Figure 5.8: Missing Rate

Figure 5.9: Analysis of VARSA using different Tracking Intervals

0.2.

The number of dead nodes in the network over time is shown in Figure 5.7. The number of dead nodes decreased as we increased the TI from 0.06 to 0.2. However; after we increased the TI more than 0.2, the number of dead nodes increased. Thus, this figure also reveals that the best value for TI is 0.2 between the examined values for TI.

Figure 5.8 represents the missing rate of the target. As the figure shows, the minimum missing rate over time which is the best tracking quality is achieved for TI=0.2. The missing rate is increased after we further increased the TI of the tracking algorithm.

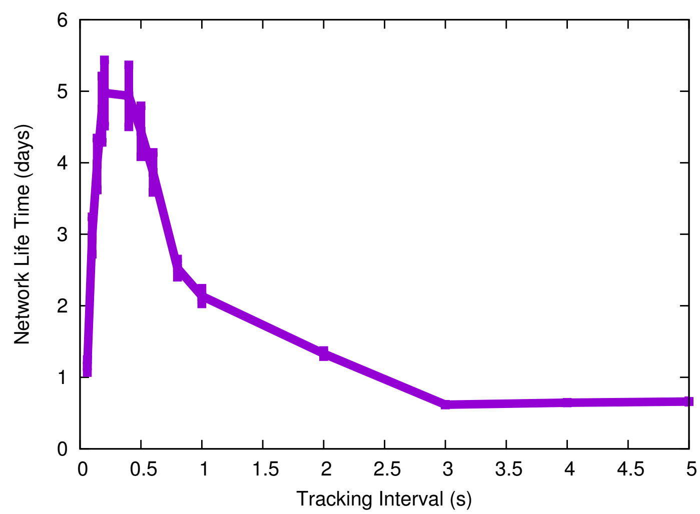

Figure 5.10: Network Life Time using different Tracking Intervals

Life time of the network reveals the ability of the network to locate the target as the network ages. Figure 5.10 confirms our previous observations that TI=0.2 is the best choice between the examined tracking intervals for VARSA.

To better analyze the algorithm, first step recovery, second step recovery, full recovery and full radius recovery rates are illustrated in Figure 5.15. When the tracking interval is 4 seconds, more first step recoveries are performed. As the tracking interval is decreased to 0.2 seconds, less first step recovery is needed to locate the missed target. However, as the tracking interval is further decreased, the number of the performed first step recoveries are higher.

In Figure 5.13, the second step recovery rate is represented. As the network ages, the number of second step recoveries are closer to the number of first step recoveries since the first step recoveries are not able to locate the target anymore. The least second step recovery rate is for tracking interval of 0.2 seconds.

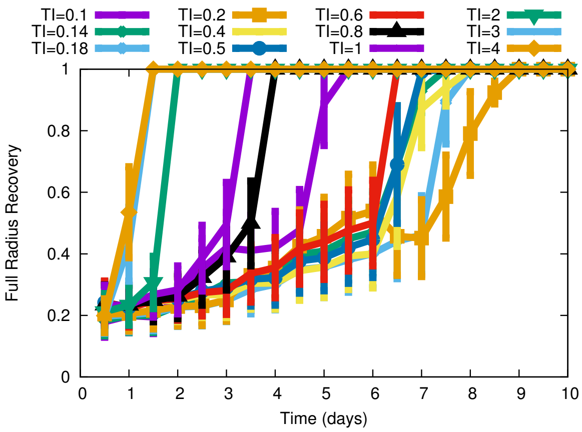

Figure 5.11: Full Radius Recovery Rate

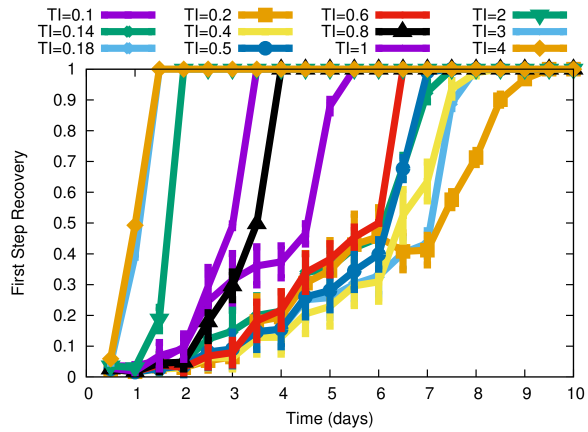

Figure 5.12: First Step Recovery Rate

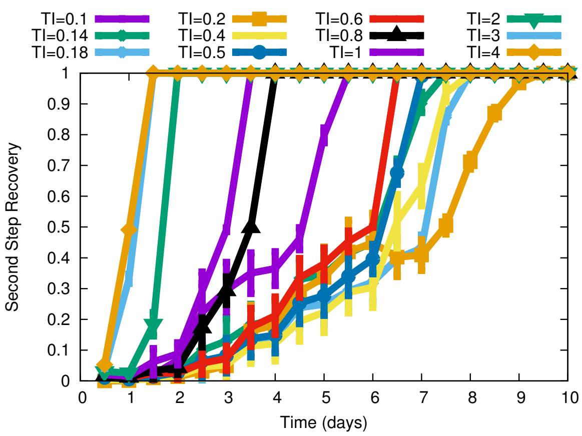

Figure 5.13: Second Step Recovery Rate

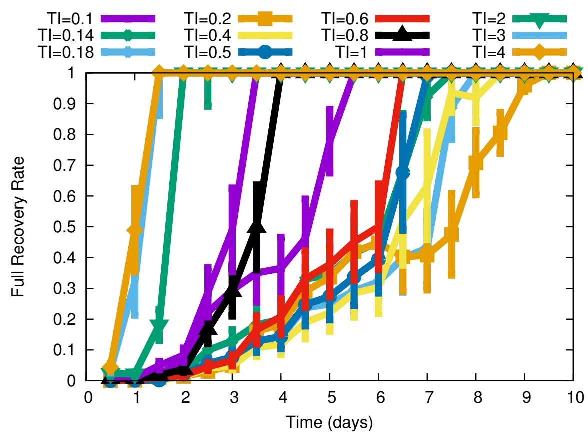

Figure 5.14: Full Network Recovery Rate

Figure 5.15: Recovery Analysis of VARSA using different Tracking Intervals

Full network recovery rate is shown in Figure 5.14. This graph shows the need of the algorithm to wake up the whole sensor nodes to locate the target. Tracking interval of 0.2 seconds achieves the minimum full recovery rate.

Full radius recovery rate is shown in Figure 5.11. This figure reveals that the tracking interval of 0.2 seconds needs the least number of full radius recoveries to locate the target.

Tracking Duty Cycle Tuning

Duty cycle of tracking is the percentage of one interval of tracking in which the network is active. One interval of tracking is the total time that the network is activated and then deactivated for detecting the target. In the first predetermined percentage of each tracking time interval, some sensor nodes are active to locate the target and in the rest of the tracking interval, all the sensor nodes are inactive. The duty cycle can be calculated using Equation 5.2.

$$DC = \frac{AT}{TI} \tag{5.2}$$

where AT is the active time of sensing in each interval of tracking and TI is the tracking time interval.

Several simulations are conducted to find the best duty cycle for tracking. We have examined 10, 20, 35, 50, 65 and 80. The remaining energy in the network is shown in Figure 5.16. Overall, as the duty cycle is increased, the remaining energy is decreased over time; however, when the duty cycle is increased to 80, the remaining energy is increased over time since the missing rate was less and consequently, less recovery is needed. Hence, less expansion of the radio coverage area in the network is required and the remaining energy of the network is increased over time.

Figure 5.17 represents the consumed energy for sensing. Sensing energy consumption is increased as the duty cycle is increased. However, we can see that the sensing energy consumption is less for duty cycles more than 65% which shows that increasing the duty cycle causes the missing rate to be decreased and since less recovery instances are needed, less energy is consumed for sensing.

Number of dead nodes in the network is shown in Figure 5.18. This figure shows how more sensor nodes are dead after increasing the duty cycle; but after increasing the duty cycle more than 65%, again the number of dead nodes is decreased due to the less recovery needed during the tracking.

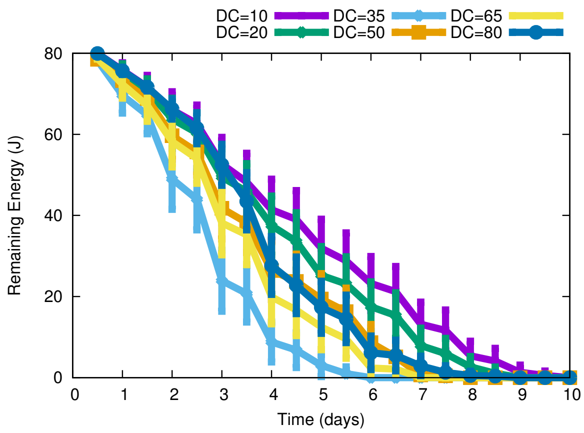

Figure 5.16: Total Remaining Energy

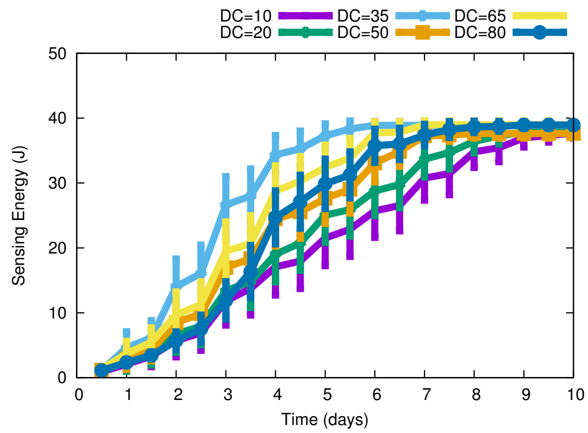

Figure 5.17: Sensing Energy Consumption

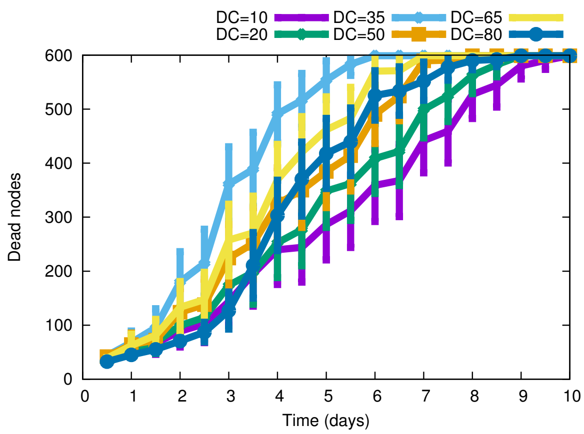

Figure 5.18: Number of Dead Nodes

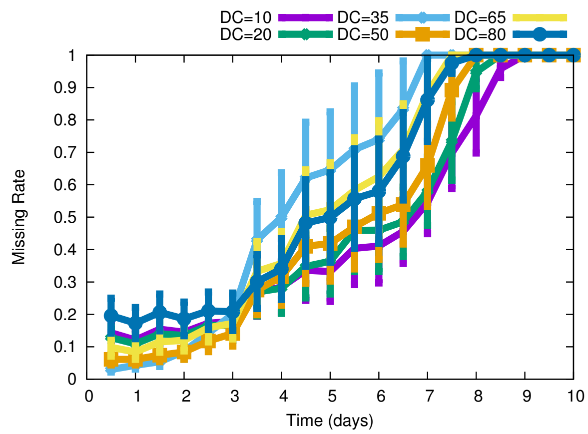

Figure 5.19: Missing Rate

Figure 5.20: Analysis of VARSA using different Duty Cycles

Figure 5.19 represents the missing rate of the proposed algorithm. Overall, the missing rate is increased as the network ages. Duty cycle 10% and 20% provide the least missing rate compared to the other duty cycles. Duty cycle 35% causes the highest missing rate after the third day as more nodes are dead at this time. Due to the increase of missing rate, more recovery states are initiated and the network depletes its energy faster.

Life time of the network is shown in Figure 5.21. This figure represents that the best lifetime is for the duty cycle of 10% and after that for 20%. The life time decreases as the duty cycle is increased to 35%. After this point, the life time increases.

We have selected the duty cycle to be 20% as the network life time of VARSA using duty cycle 20% is the best after the network life time of VARSA using duty cycle 10%

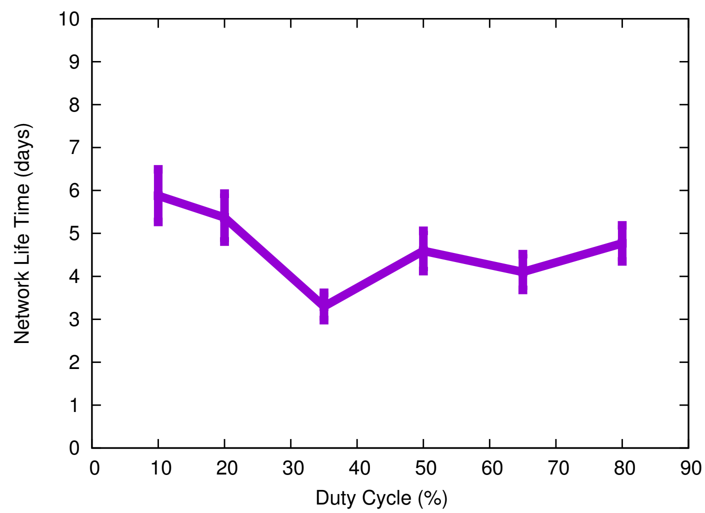

Figure 5.21: Network Life Time using different Duty Cycles

and it provides less missing rate when the network is tracking the target in the first days of tracking when less sensor nodes are dead.

Figure 5.26 can further show the performance of the algorithms. These figures are the rate of different recovery steps, performed in the network. This figure reveals the moments that the target was missed and the ability of the network to overcome the target missed instances.

Figure 5.23 reveals the number of first step recoveries performed during the tracking time. In the first step of recovery, all the one hop neighbours of the current detector are activated to locate the target. This figure shows a similar behaviour as the missing rate. As the duty cycle is increased, the first step recovery rate is also increased but when the duty cycle is 35%, the performed first step recoveries are more which explains the behaviour of the consumed energy for sensing and the remaining energy of the network.

Figures 5.24 and 5.25 reveal the second step recovery rate and the full recovery rate in the network relatively. The behaviour of these diagrams are similar to the first step recovery rate in the rate of increase and decrease.

Finally, Figure 5.22 shows the full radius recovery. This diagram is similar to the missing rate as this is the first recovery performed after missing a target. This figure reveals how VARSA can locate the target by extending its sensing radius to its maximum.

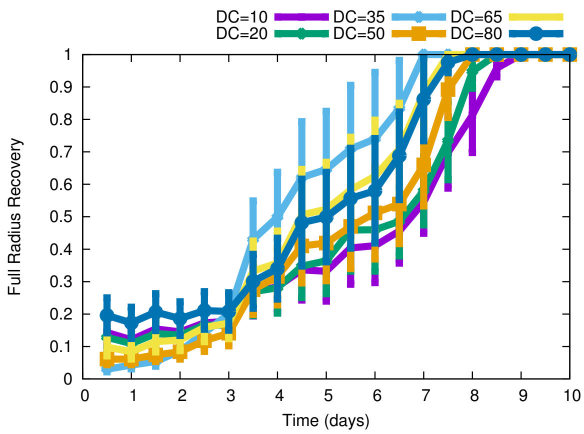

Figure 5.22: Full Radius Recovery Rate

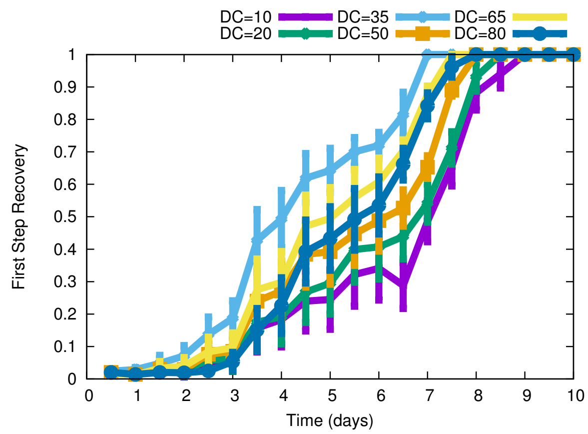

Figure 5.23: First Step Recovery Rate

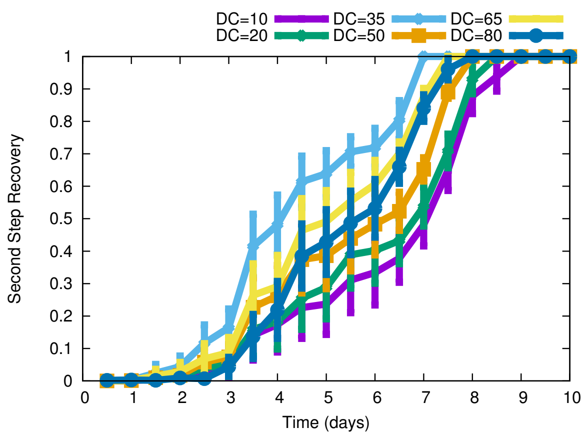

Figure 5.24: Second Step Recovery Rate

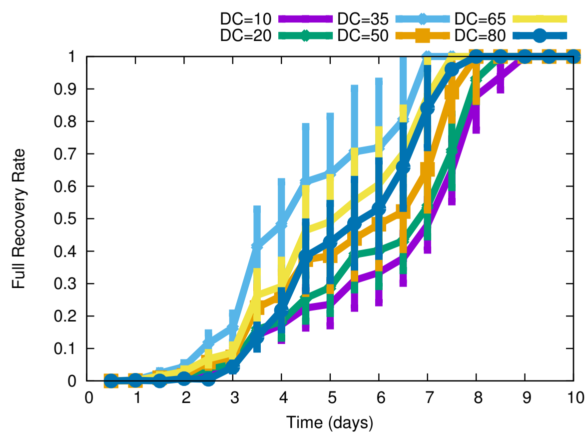

Figure 5.25: Full Network Recovery Rate

Figure 5.26: Recovery Analysis of VARSA using different Duty Cycles

Sensing Radius Adjustment Rate (ω) Tuning

Parameter ω in the radius adjustment algorithm plays a significant role in the performance of the algorithm. Parameter ω can be adjusted based on the requirements of the application. In this study, extensive simulations are conducted to adjust this parameter.

Figure 5.27 represents the sum of the remaining energy of sensor nodes in the network. It is shown that as the ω is increased, the remaining energy in the network increases. However, the increase is not substantial from $\omega = 4$ to $\omega = 10$.

Figure 5.28 represents the sensing energy consumption of the network. The sensing energy consumption decreases as the ω is increased but the decrease is negligible when we alter ω from $\omega = 4$ to $\omega = 10$.

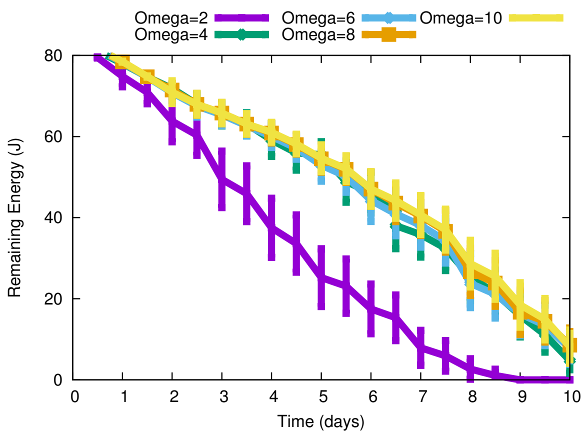

Figure 5.27: Total Remaining Energy

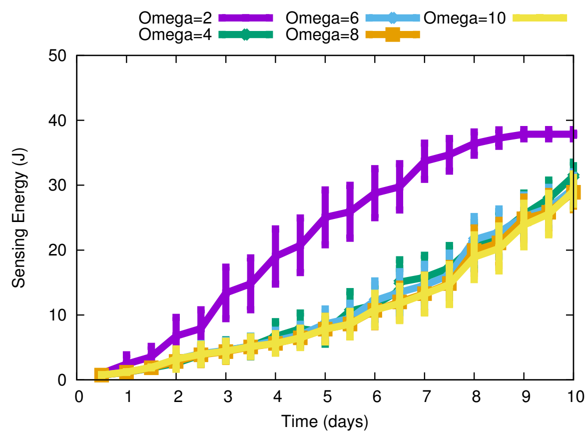

Figure 5.28: Sensing Energy Consumption

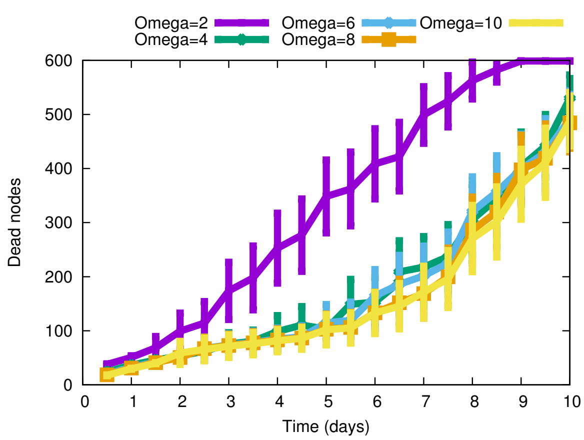

Figure 5.29: Number of Dead Nodes

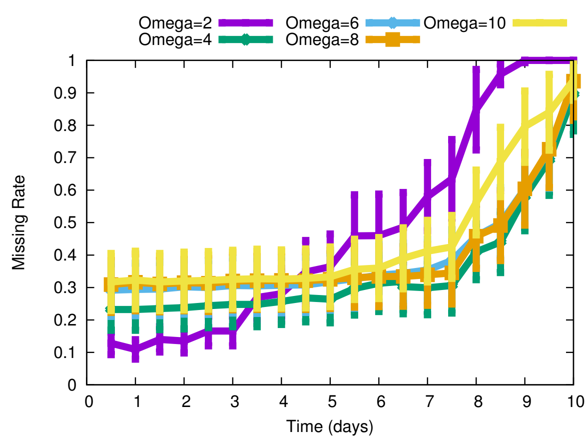

Figure 5.30: Missing Rate

Figure 5.31: Analysis of VARSA using different sensing radius adjustment rate (ω)

Number of dead nodes in the network over time is shown in Figure 5.29. As the parameter ω is set to 10, the minimum number of dead nodes over time is achieved. Maximum number of dead nodes occured when using $\omega = 2$.

Missing rate is shown in Figure 5.30. Before the fourth day of the tracking, the best missing rate is achieved for $\omega = 2$. However, after the fourth day, the missing rate is increased exponentially since there are more dead nodes and consequently coverage holes in the network. Overall, the minimum missing rate is achieved for $\omega = 4$. This ω also provides a more stable results for the missing rate over time.

The life time of the network is shown in Figure 5.32. The life time is increased as the ω is increased. However, the increase is not substantial when $4 < \omega < 10$. However, the life time for $\omega = 2$ is considerably less than the other instances.

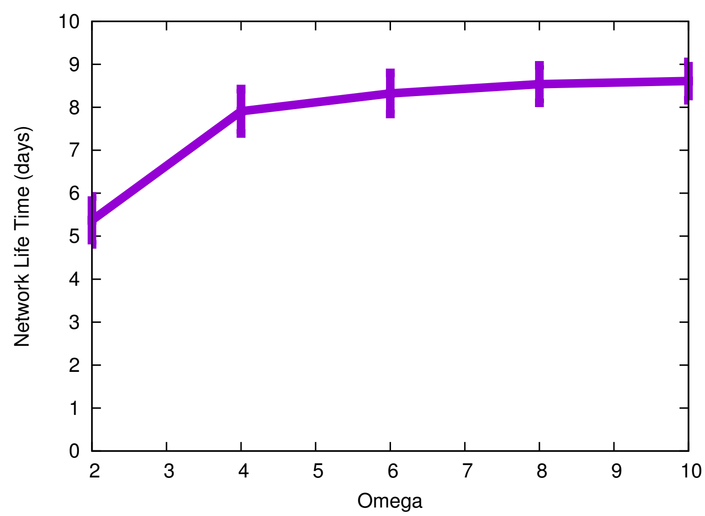

Figure 5.32: Network Life Time using different ω

As the simulation results reveal, the best value of ω is 4 as it provides the minimum missing rate over time and its life time is within 10 percent of the best life time which if for $\omega = 10$.

Recovery steps analysis, given in Figure 5.37, can also help to analyze the proposed algorithm. First step recovery rate, as provided in Figure 5.34, shows how the ω equals to 10 needs the least first step recoveries; while as the ω is decreased, the number of required first step recoveries increase. The second step recovery rate and full recovery rate behaviour is identical to the first step recovery rate as shown in Figures 5.35 and 5.36 respectively.

The weakness of using $\omega = 10$ can be shown using the required full radius recovery

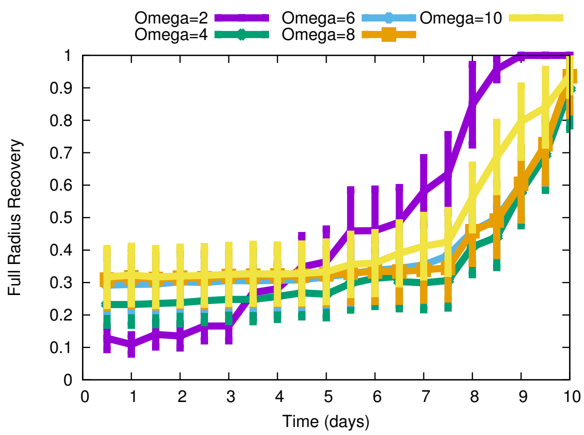

Figure 5.33: Full Radius Recovery Rate

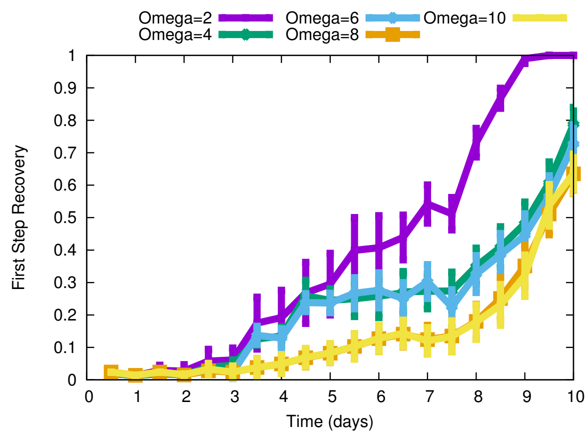

Figure 5.34: First Step Recovery Rate

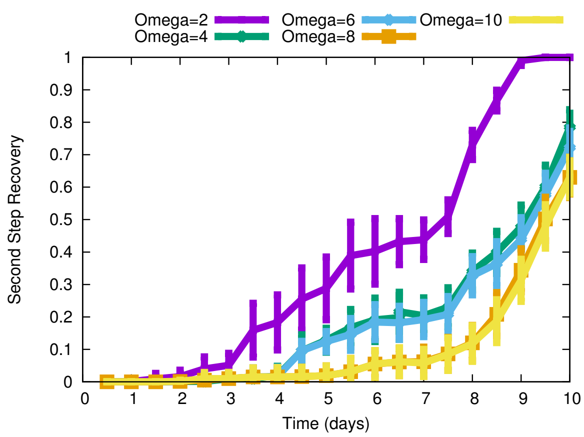

Figure 5.35: Second Step Recovery Rate

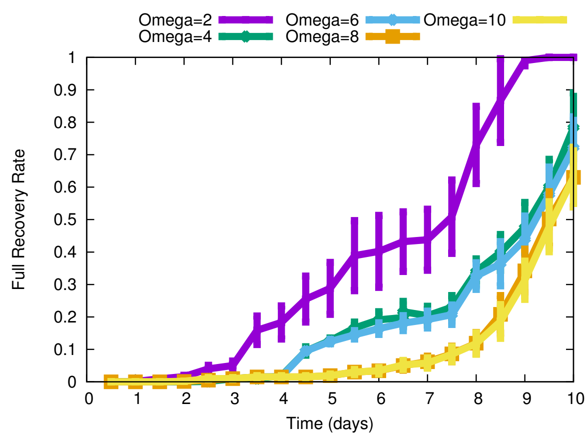

Figure 5.36: Full Network Recovery Rate

Figure 5.37: Recovery Analysis of VARSA using different sensing radius adjustment rate (ω)

given in Figure 5.33. This figure shows that the increase of ω leads to increasing the required number of full radius recoveries as the rate of decreasing the sensing range is higher. Thus, the decrease in the radio covered of the network does not improve the energy efficiency as it is required to expand the sensing range quickly to locate the target in most of the tracking time intervals.

5.3.2 Simulation Results

Energy Consumption Analysis

Energy efficiency of the algorithm is crucial in all the algorithms using WSNs. The goal of minimizing the energy consumption of the network is achieved as it is shown in Figure 5.38. In PRA, all the sensor nodes are dead at the fourth day and there is no more residual energy to be used for tracking. Thus, VARSA and PPRA sensing energy consumption goes higher than PRA after the first and ninth days respectively, as they are still tracking the target. VARSA consumes much less energy for sensing compared to the other two algorithms and it keeps tracking the target to the ninth day.

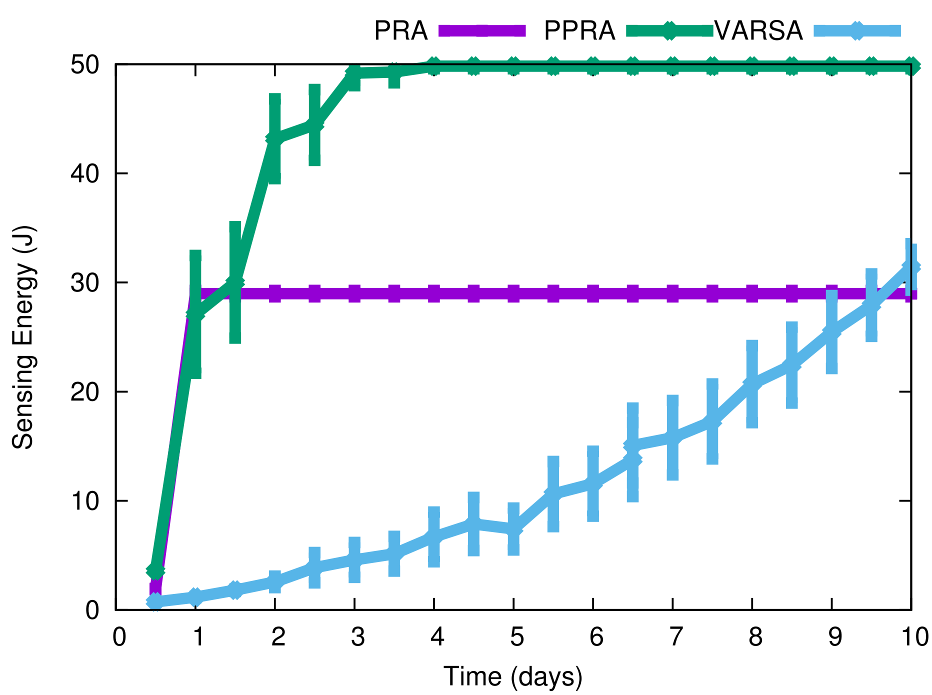

Figure 5.38: Sensing Energy Consumption

Figure 5.39 shows the total remaining energy in the network which represents the energy consumption for both sensing and communication. Even though PPRA and PRA are competitive in the first day but PPRA outperforms PRA thereafter. VARSA consumes more energy for communication as it needs to send an advisable sensing radius to wake up with to the next sensor; however, this overhead is negligible and the total consumed energy is less over the time using VARSA.

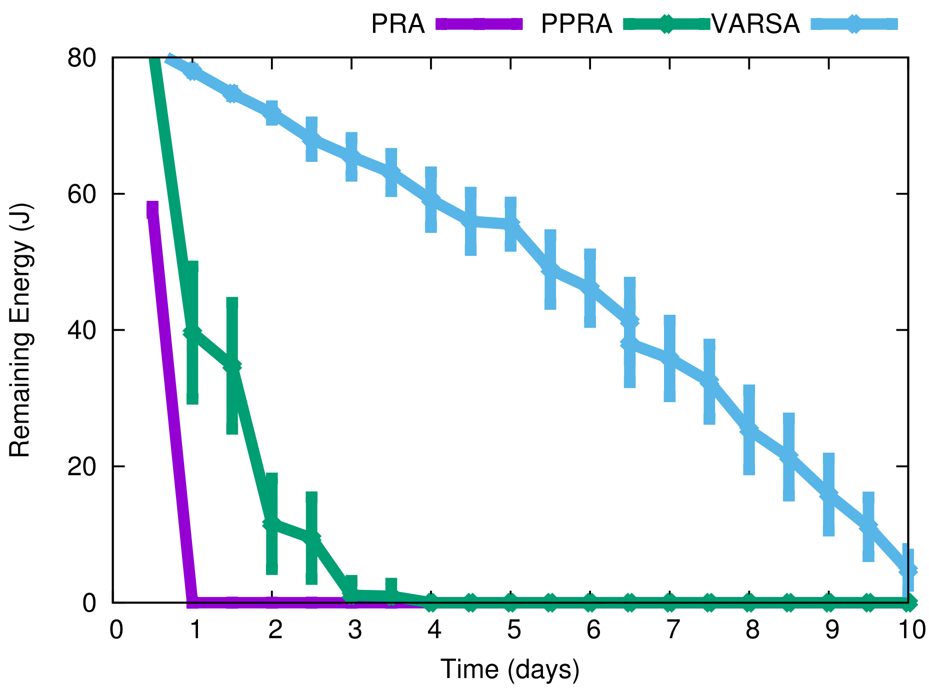

Figure 5.39: Remaining Energy of the Network

The number of dead nodes in the network as a function of time is shown in Figure 5.40. Generally, number of dead nodes increases with time but the speed of this increase is much less for VARSA. For instance, VARSA exhaust the total energy of 10 percent of nodes after 3 days, while the other two algorithms consumed almost all the available energy in the network at this time. Number of dead nodes shows the network holes or uncovered geographical places which might cause the target lost or tracking error. In addition, when more sensors are dead, target lost instances increases and the network expands its monitoring area to locate the target which leads to exhausting the energy of more sensor nodes and eventually an exponentially increase in the number of dead nodes. The results also reveal that when around 20 percent of the network is dead, number of dead nodes increase exponentially due to the high increase in the number of performed recoveries.

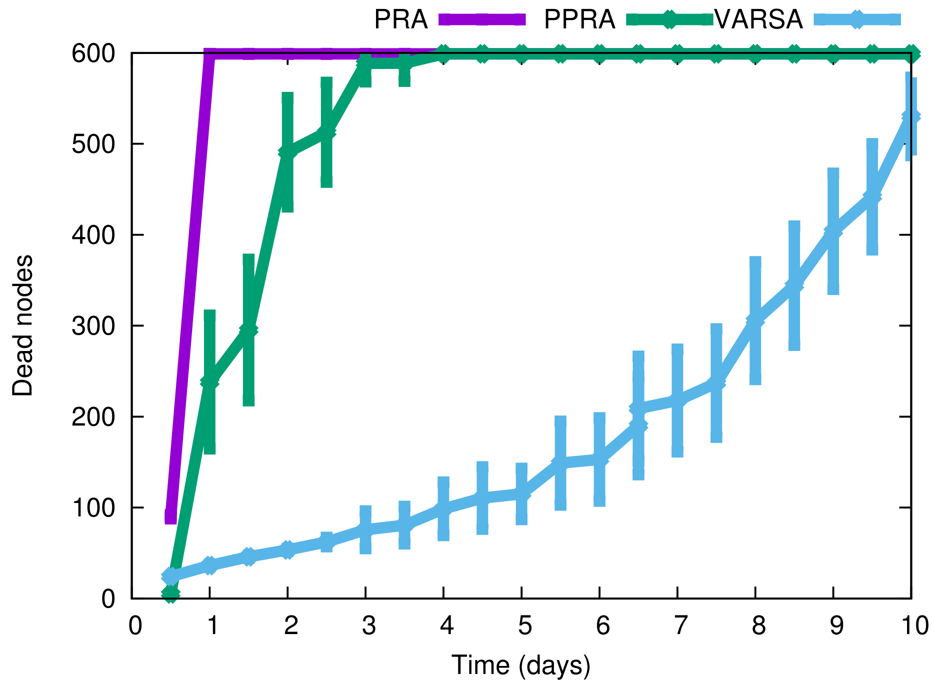

Figure 5.40: Number of Dead Nodes in the Network

Network life time results are summarized in Figure 5.41. Network life time is 7.9 days for VARSA; while the other two algorithms deplete the energy of the network much faster. While around 100 nodes are dead after 12 hours of tracking using PRA and in first day of tracking using PPRA; VARSA depletes the energy of 100 nodes after the fifth day. Thus, the network life time is prolonged due to the significant energy saved in sensing.

Tracking Quality Analysis

Missing rate is illustrated in Figure 5.42. VARSA does not only provide less missing rate as the time passes compared to the other two algorithms but it also initiates a full radius recovery which might cause to assist the system to locate the target without awaking more sensor nodes. Missing rate represents the tracking quality and VARSA significantly outperforms the other two algorithms after the first 36 hours. Thus, saving energy caused to improve the tracking quality over the time. By analyzing Figures 5.40 and 5.42, it can be concluded that when the number of dead nodes increases, missing rate increases drastically.

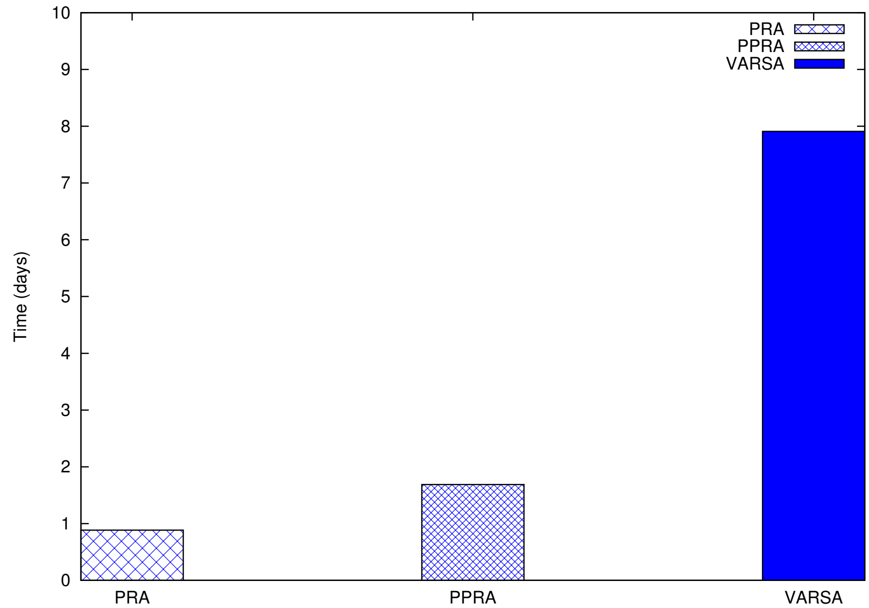

Figure 5.41: Network Life Time

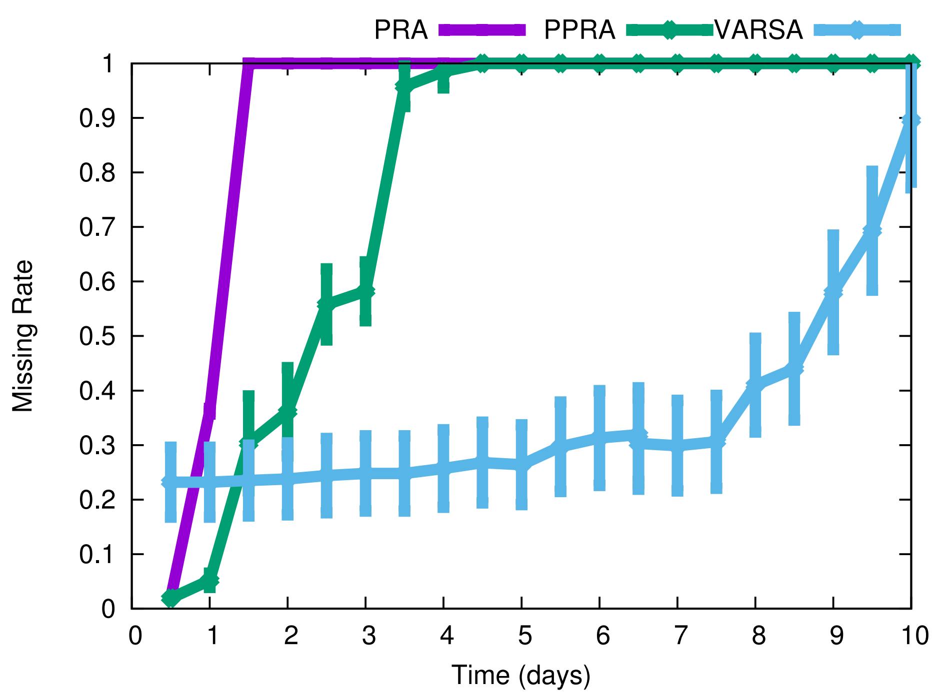

Figure 5.42: Missing Rate

Recovery Mechanism Analysis

First step, second step and full recovery rates are shown in Figures 5.43, 5.44 and 5.45 respectively. These figures demonstrate high performance of VARSA as it needs to turn on less sensors than the other two algorithms. However, VARSA needs to adjust its sensing radius to cover the missing target in the first recovery step.

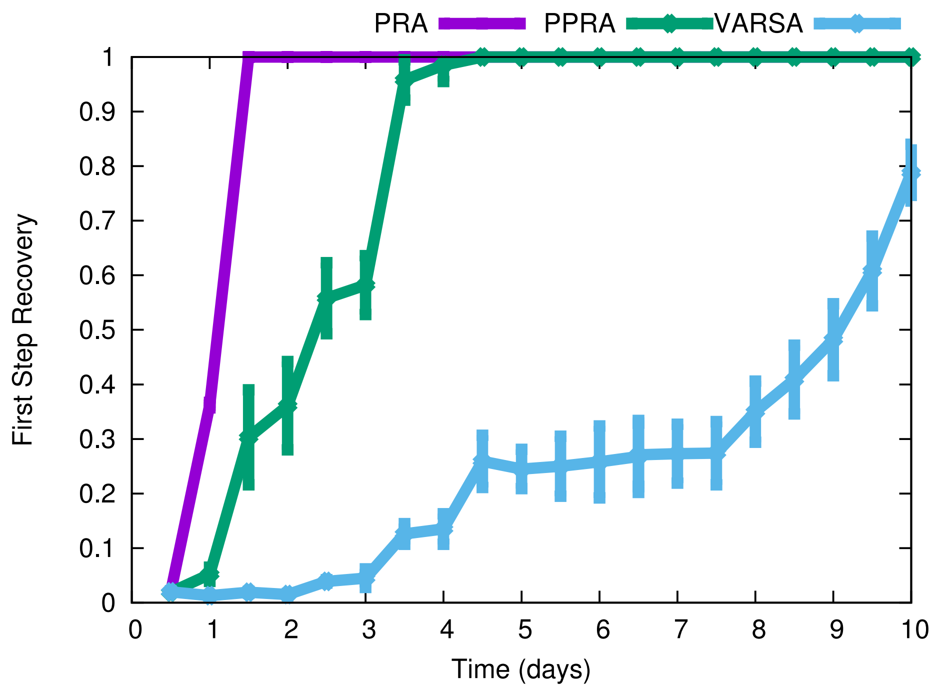

Figure 5.43: First Step Recovery Rate

Figure 5.43 shows that PRA requires more first step recoveries compared to PPRA and VARSA in order to locate the target. As all the sensor nodes are dead after 1.5 days using PRA, the first step recovery rate is always one after this time. It shows that the system were not able to locate the target after the first 1.5 days since there were not alive sensor nodes to locate the target.

VARSA requires the least second step recoveries compared to PRA and PPRA as shown in Figure 5.44. While PRA and PPRA need to perform the second step recovery in each tracking interval after the 1.5 and 4.5 days respectively, VARSA requires to initiate the second step recovery in 80 percent of the tracking intervals in the 10th day.

Full recovery rates for VARSA, PPRA and PRA are shown in Figure 5.45. VARSA needs less full recovery to locate the target which leads to saving more energy and helps to prolong the life time of the network.

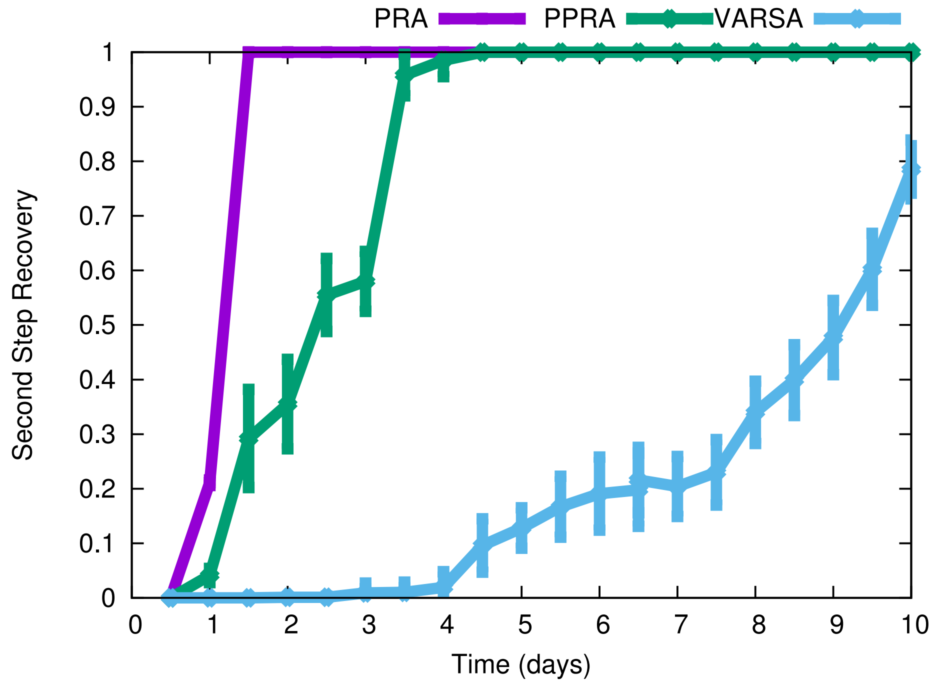

Figure 5.44: Second Step Recovery Rate

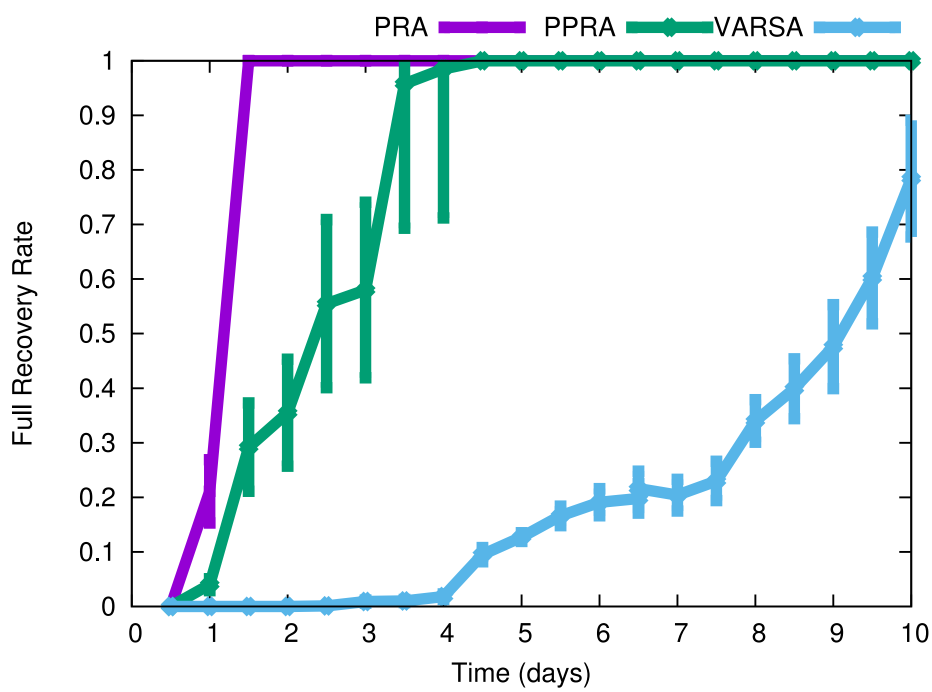

Figure 5.45: Full Network Recovery Rate

Even though VARSA performed less first step, second step and full recoveries during the tracking, it performs a full radius recovery step. Full radius recovery aims at increasing the sensing range of the current sensor node to its maximum to locate the target. The full radius recovery rate for VARSA is shown in Figure 5.46

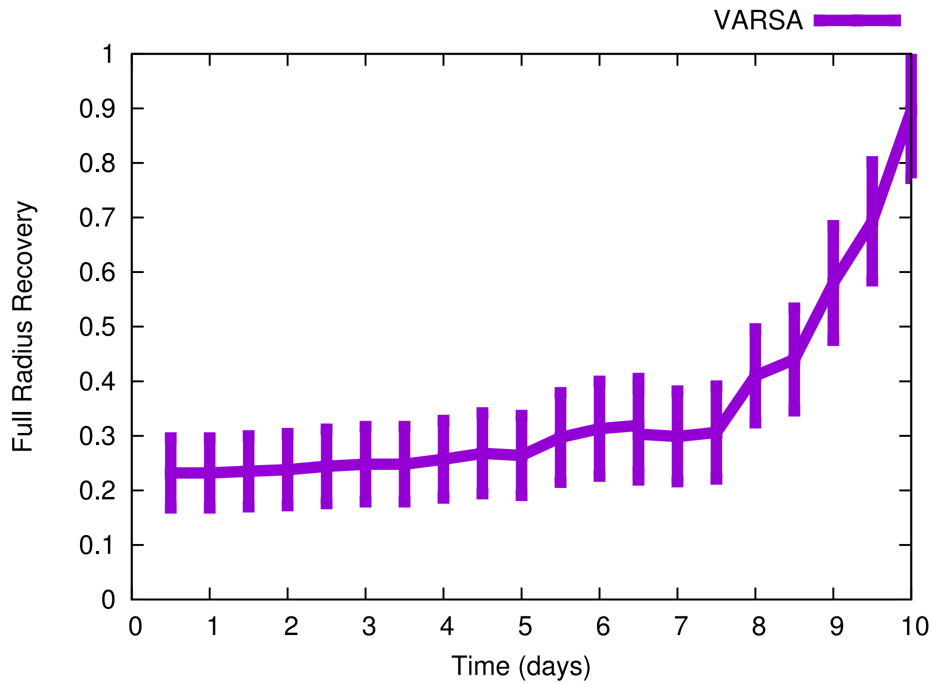

Figure 5.46: Full Radius Recovery Rate

5.3.3 Sensor Deployment Density

In this section, we have studied the effect of the number of nodes in the network on the performance of PRA, PPRA and VARSA. In the previous experiments, we have evaluated these protocols by deploying 600 nodes with a uniform random distribution in the tracking field. The remaining energy of the network, sensing energy consumption, number of dead nodes in the network, missing rate and the life time of the network, when 200, 400 and 800 nodes are deployed in the tracking area, are evaluated in this section.

Figure 5.51 summarizes the performance evaluation of PRA, PPRA and VARSA. As deploying 200 nodes in an area of 200×200 is a very sparse deployment, the performance of none of the algorithms is good. VARSA performs marginally better in terms of missing rate and the remaining energy of the network but even for VARSA, there is no alive sensor node to locate the target after the first day of tracking.

Sensing energy consumption of PRA, PPRA and VARSA has been illustrated in Figure 5.48. This figure shows that VARSA consumes more energy for sensing after the first day of tracking compared to PRA since PRA already used all its energy for communication and sensing and it does not have energy resources to use for sensing after the first day. Thus, the fact that PRA used less sensing energy after the first day does not show its good performance as there is no more energy resources in the network to be used for sensing.

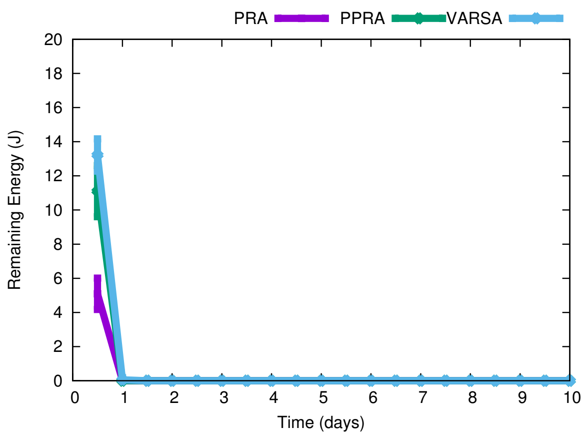

Figure 5.47: Total Remaining Energy

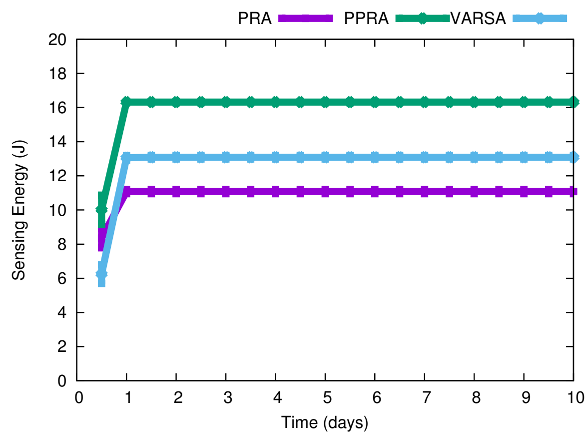

Figure 5.48: Sensing Energy Consumption

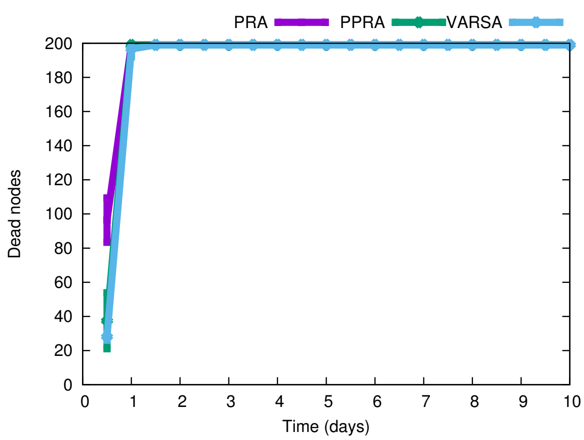

Figure 5.49: Number of Dead Nodes

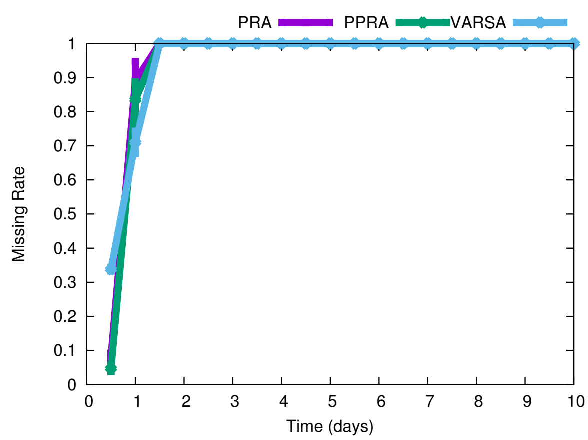

Figure 5.50: Missing Rate

Figure 5.51: Analysis of VARSA with 200 nodes

The evaluation of target tracking using VARSA, PPRA and PRA when 400 nodes have been deployed in the tracking area is provided in Figure 5.56. VARSA outperforms PPRA and PRA significantly in terms of sensing energy consumption, remaining energy in the network, number of the dead nodes and missing rate. After the fifth day of using VARSA, almost all of the sensor nodes are dead and VARSA is not able to locate the target anymore. However, PRA and PPRA lose their functionality after the first day of tracking.

Overall, the performance of VARSA when 400 nodes have been deployed is considerably better than deploying 200 nodes. Hence, the performance improvement of PRA and PPRA is negligible when more nodes are used. Comparing Figures 5.51 and 5.56, it can be seen that PPRA and PRA cannot keep their tracking functionality after the first day independent of the density of the deployed sensor nodes.

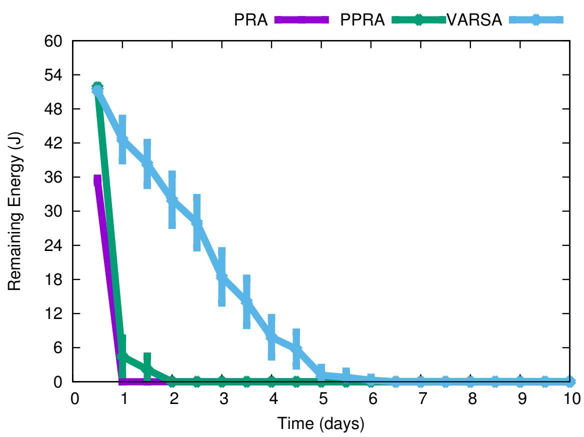

Figure 5.52: Total Remaining Energy

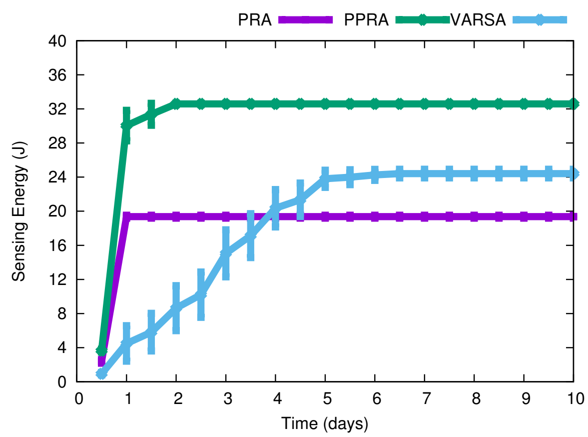

Figure 5.53: Sensing Energy Consumption

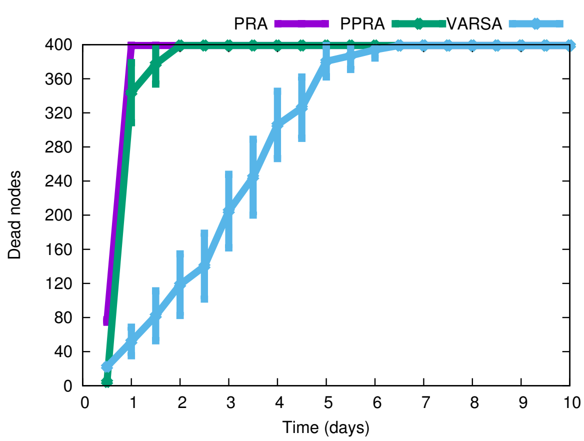

Figure 5.54: Number of Dead Nodes

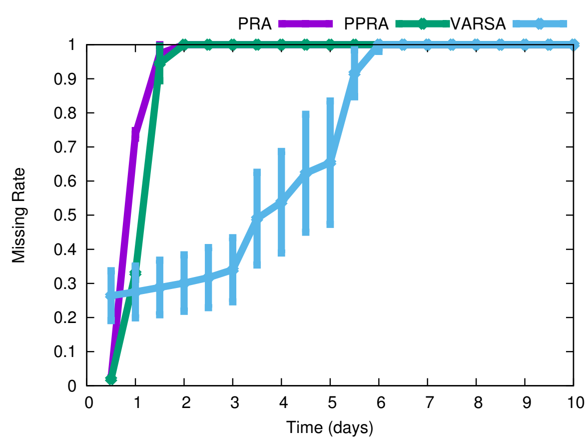

Figure 5.55: Missing Rate

Figure 5.56: Analysis of VARSA with 400 nodes

Figure 5.61 illustrates the performance of VARSA, PPRA and PRA when 800 nodes are deployed in the tracking area. The performance of VARSA improves significantly by using 800 nodes in the tracking area. As the proximity sensor nodes are cost-effective, the dense deployment of the sensor nodes are practical.

The remaining energy in the network is shown in Figure 5.57. This figure reveals that even after 10th day from the start of tracking, VARSA still preserves almost 30% of the energy that it had at the beginning of tracking. PRA depletes all of the available energy of the network within the first day. PPRA outperforms PRA in terms of remaining energy in the network as PPRA depletes the energy of the sensor nodes in six days.

Figure 5.58 reveals the sensing energy consumption of VARSA using 800 nodes. This figure shows that VARSA consumes less energy for sensing compared to PRA and PPRA. This figure shows the effect of VARSA in decreasing the sensing energy consumption. This decrease in the sensing energy consumption provides more alive sensor nodes in the network over time to detect the target. Thus, VARSA can also improve the tracking quality over time.

Number of dead nodes in the network is shown in Figure 5.59. This figure shows that VARSA provides around half of the deployed sensors to detect the target even after ten days. PRA and PPRA deplete the energy of all the sensor nodes after the first and sixth days of tracking respectively.

Figure 5.60 represents the missing rate of the target over time. Missing rate of the target is around 10% even after the tenth day of tracking. The improvement of the missing rate is due the decrease in sensing energy consumption which provided more alive sensors to detect the target over time. In addition, VARSA considerably outperforms PRA and PPRA in terms of tracking quality as PRA and PPRA are not able to track the target after first and sixth days of tracking respectively.

By comparing Figures 5.61 and 5.56, it can be seen that the performance of VARSA using dense sensor deployment is remarkably higher. However, VARSA always outperforms PRA and PPRA over time due the efficiency of the proposed sensing radius tuning regardless of the deployed sensor nodes density. In the other hand, a dense deployment of nodes using VARSA can provide even a better tracking accuracy over time and it can also prolong the network life time.

The life time of the network when VARSA is used for tracking a single target with 200, 400, 600 and 800 nodes is illustrated in Figure 5.62. This figure reveals that VARSA always outperforms PRA and PPRA in terms of the network life time. In addition, the life time of the network is increased as the number of deployed nodes in the tracking area is increased. This increase is negligible for PRA; thus, deploying a dense sensor network

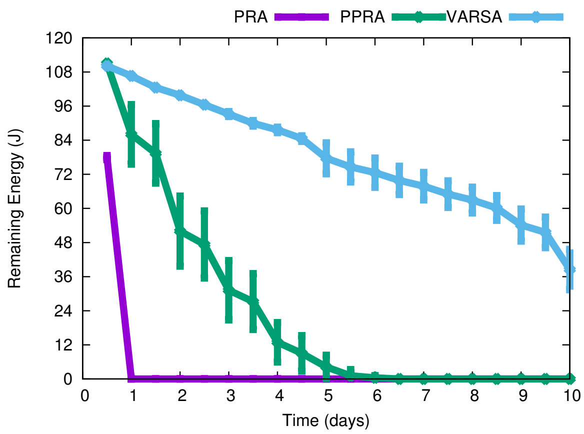

Figure 5.57: Total Remaining Energy

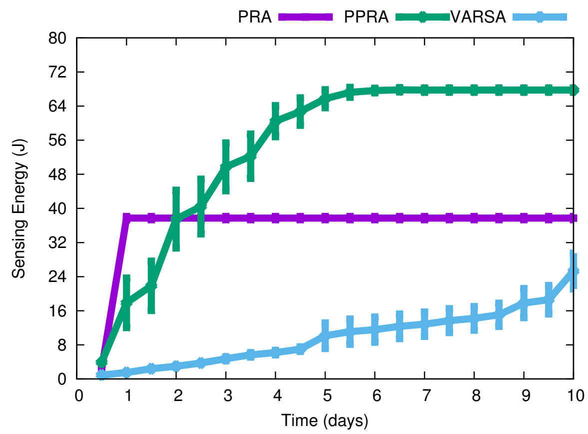

Figure 5.58: Sensing Energy Consumption

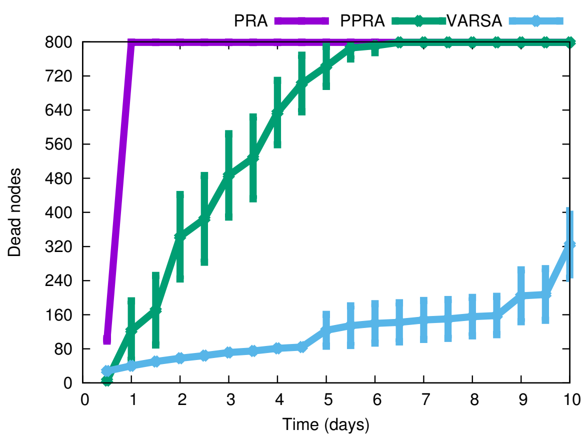

Figure 5.59: Number of Dead Nodes

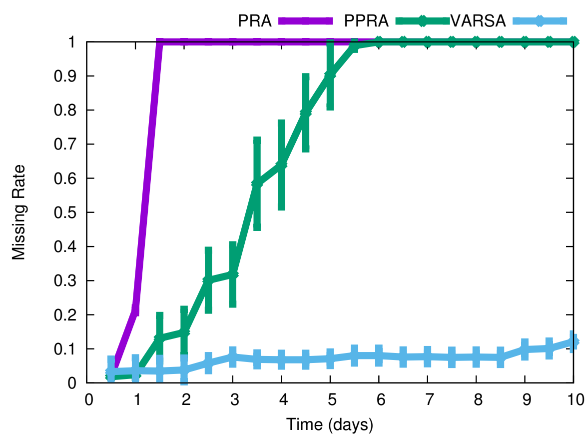

Figure 5.60: Missing Rate

Figure 5.61: Analysis of VARSA with 800 nodes

can not improve the performance of PRA considerably. Life time of PPRA is increased significantly as more sensor nodes are deployed in the tracking area. However, VARSA still outperforms PPRA even when more sensor nodes are deployed in the network. The slope of increase of the lifetime is much higher for VARSA and the dense deployment is able to remarkably improve the life time of VARSA.

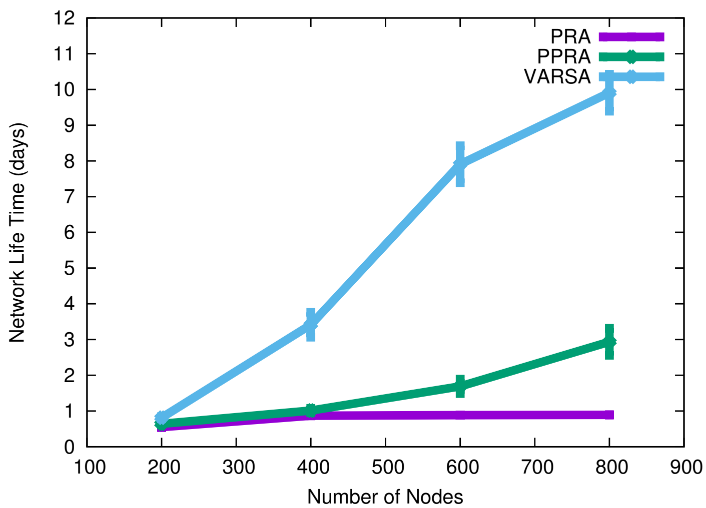

Figure 5.62: Network Life Time using different Number of Nodes

5.3.4 VARSA Evaluation using Pursue Mobility Model

The performance of VARSA for tracking a target with Random Waypoint mobility has been studied in previous sections. In this section, we have evaluated VARSA for tracking a target with Pursue mobility model. Pursue mobility model represents the mobility model of a criminal escaping from cops or a deer escaping from the hunters. Thus, the movements are quick and there is no pause during the movements. We have evaluated VARSA using Pursue mobility model for one hour of tracking, since the escape and pursue movements are not as long as the ordinary movement patterns. This simulation also reveals the functionality of the algorithms when almost all the sensor nodes are alive. The performance of VARSA is compared with PRA and PPRA in Figure 5.67.

The total remaining energy in the network when tracking a target using VARSA, PPRA and PRA for one hour is provided in Figure 5.63. This figure shows that VARSA and PPRA considerably outperform PRA; however, the remaining energy of the network is almost identical for PPRA and VARSA.

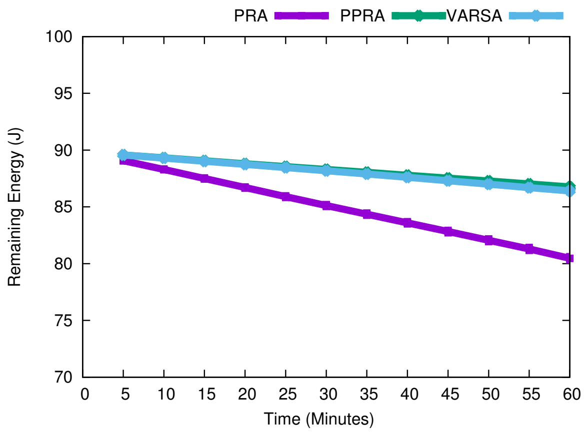

Figure 5.63: Total Remaining Energy

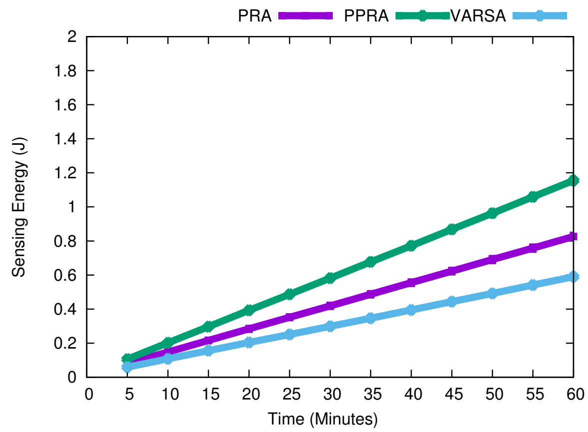

Figure 5.64: Sensing Energy Consumption

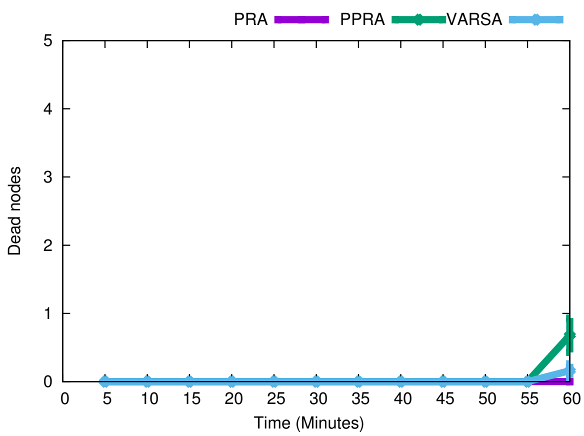

Figure 5.65: Number of Dead Nodes

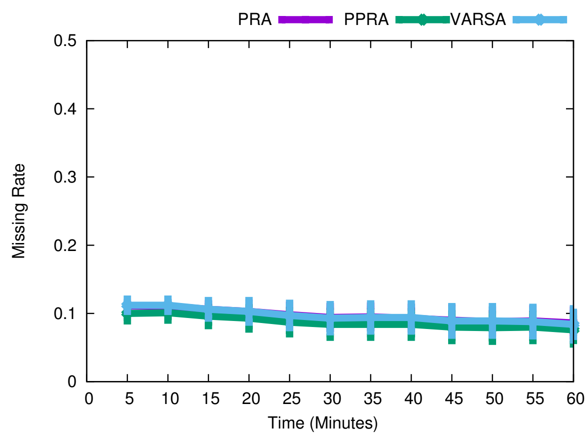

Figure 5.66: Missing Rate

Figure 5.67: Analysis of VARSA for Pursue Mobility Model

Figure 5.64 reveals the sensing energy consumption of the network over time. VARSA consumes less energy for sensing compared to PRA and PPRA. By comparing Figures 5.63 and 5.64, it can be seen that even though the remaining energy of the network was comparable using VARSA and PPRA but VARSA consumes less energy for sensing. Thus, the consumed energy for communication was more for VARSA since VARSA requires to send the advisable sensing radius to wake up with to the next predicted sensor node. This control packets overhead cause VARSA to consume more energy for sensing; however, this energy is compensated by the energy saved in sensing.

Figure 5.65 shows the number of dead nodes in the network. This figure shows that almost all of the sensor nodes are alive during the tracking as the average number of dead nodes for 50 runs is less than one node for PRA, PPRA and VARSA. Thus, this simulation shows the performance of the evaluated algorithms when all the sensor nodes are available for sensing.

Figure 5.66 illustrates the missing rate of the target over time. This figure can reveal the tracking quality of PRA, PPRA and VARSA. Overall, the missing rate of VARSA is almost the same as the missing rate of PPRA and PRA in a short tracking time. However, it is shown in the previous sections that VARSA outperforms PPRA and PRA as the network ages. Since the performance of VARSA is comparable to PRA and PPRA in the first hour of tracking and VARSA outperforms the other two algorithms significantly over time; thus using VARSA is beneficial for target tracking applications, specially for the ones which take longer times.

5.3.5 Summary

This chapter is an aim to evaluating the performance of VARSA. Three effective parameters of the algorithms are examined in a wide range using simulations to find the most appropriate values. Tracking interval, duty cycle of activation and the rate of changing the radio covered area in sensing radius adjustment state, ω, are tuned in this chapter. Then, VARSA is compared to two other promising sensor activation algorithms, PRediction-based Activation (PRA) and Periodic PRediction-based Activation (PPRA). VARSA is shown to prolong the network life time and decrease the missing rate of the target over time. In addition, the impact of the deployed sensor nodes density and the mobility model of the target is investigated in this chapter. The results reveal that dense deployment of sensor nodes can improve the tracking quality and network life time.

Chapter 6

Conclusion and Future Work

In this chapter, we have concluded this thesis with summarizing some of our important results. Then, some of the possible directions for future studies are highlighted.

6.1 Conclusion

In this thesis, the proposed algorithms for target tracking using WSNs are analyzed and compared. We have elaborated and categorized the sensor deployment techniques, node and target localization techniques and sensor activation algorithms for target tracking using WSNs. Then, we have designed a tracking algorithm including the sensor deployment, prediction model, activation mechanism, recovery technique and the localization.

A novel tracking algorithm for WSNs to decrease the consumed energy for sensing and prolong the network lifetime through real-time sensing radius adjustment is proposed in this study. Our aim is to decrease the sensing energy consumption over time by adjusting the sensing radius of the sensor nodes to only cover the AoI. VAriable Sensor Activation algorithm, *VARSA*, decreases the sensing energy consumption by sending the advisable sensing radius to wake up with to the next predicted sensor node in the AoI. VARSA also performs a sensing radius adjustment mechanism to further decrease the radio covered area of the network to minimize the sensing energy consumption.

Through extensive simulations, we have shown that *VARSA* consumes less sensing energy and achieves a lower missing rate compared to PRediction based Activation, *PRA*, and Periodic PRediction based activation, *PPRA*, algorithms using Random Waypoint mobility model for the target. In addition, VARSA outperforms the other two algorithms in terms of percentage of alive sensors over time and the total consumed energy for sensing and communication. VARSA improves the tracking quality as the network ages due to more available alive sensor nodes.

The performance of PRA, PPRA and VARSA using Pursue mobility model is also investigated in this thesis. While Random Waypoint model is used to evaluate the algorithms for longer tracking times and to show the performance of the tracking algorithm as the network ages; Pursue model is used for showing the performance of the algorithm in a short time of tracking. A criminal escaping form cops or a deer escaping from the hunter are two instances of the Pursue model. The simulation results has shown that VARSA consumes more energy for communication but it is compensated by the considerable energy saved for sensing.

6.2 Future Work

In this study, we have studied and analyzed the current state of art for target tracking from the sensor deployment, sensor nodes localization techniques, target localization techniques to sensor activation mechanisms. We have also proposed a tracking algorithm to decrease the sensing energy consumption. However, there are still open research directions that requires to be investigated as discussed below.

In VARSA, the source decides the next node to wake up; however, the current node can start an auction and asks all the candidates to communicate with each other and agree on one node to continue the tracking. This might increase the communication energy consumption but it can help the algorithm to preserve the sensors with less available energy alive for a longer time.

In some tracking applications, the tracker requires to be mobile and pursue the target. The tracker aggregates the location information of the target from the sensor nodes in order to decide which direction to go in order to catch the target. The functionality of VARSA for mobile trackers can be investigated in a future research study.

Tracking multiple targets using proximity sensor nodes confronts severe challenges due the limited information provided by these sensors. Hence, the problem of multiple target tracking using sensor networks requires to be investigated in another research study.

Network resilience, or the ability of network to recover from node failures, should be investigated as a future study. In addition, a real testbed of the tracking network can be implemented to better evaluate VARSA.

Bibliography

[1] Ameer Ahmed Abbasi and Mohamed Younis. A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30(14):2826–2841, 2007.

[2] Osama Abumansoor and Azzedine Boukerche. A secure cooperative approach for nonline-of-sight location verification in vanet. *IEEE Transactions on Vehicular Technology*, 61(1):275–285, 2012.

[3] Joe Albowicz, Alvin Chen, and Lixia Zhang. Recursive position estimation in sensor networks. In *Network Protocols, 2001. Ninth International Conference on*, pages 35–41. IEEE, 2001.

[4] Youngwon Kim An. *Multiple targets detection and tracking with Doppler effect in noisy acoustic wireless sensor networks*. PhD thesis, THE UNIVERSITY OF ALABAMA IN HUNTSVILLE, 2013.

[5] Youngwon Kim An, Seong-Moo Yoo, Changhyuk An, and B Earl Wells. Doppler effect on target tracking in wireless sensor networks. *Computer Communications*, 36(7):834–848, 2013.

[6] Anish Arora, Prabal Dutta, Sandip Bapat, Vinod Kulathumani, Hongwei Zhang, Vinayak Naik, Vineet Mittal, Hui Cao, Murat Demirbas, Mohamed Gouda, et al. A line in the sand: a wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46(5):605–634, 2004.

[7] Nils Aschenbruck, Raphael Ernst, Elmar Gerhards-Padilla, and Matthias Schwamborn. Bonnmotion: a mobility scenario generation and analysis tool. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, page 51. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010.

[8] Athanasios Bamis, Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks. *Computer Networks*, 52(1):130–154, 2008.

[9] Novella Bartolini, Tiziana Calamoneri, Tom La Porta, Chiara Petrioli, and Simone Silvestri. Sensor activation and radius adaptation (sara) in heterogeneous sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 8(3):24, 2012.

[10] Sania Bhatti and Jie Xu. Survey of target tracking protocols using wireless sensor network. In *Wireless and Mobile Communications, 2009. ICWMC'09. Fifth International Conference on*, pages 110–115. IEEE, 2009.

[11] Alledine Boukerche, HAB Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Secure localization algorithms for wireless sensor networks. *Communications Magazine, IEEE*, 46(4):96–101, 2008.

[12] Azzedine Boukerche. *Handbook of algorithms for wireless networking and mobile computing*. CRC Press, 2005.

[13] Azzedine Boukerche. *Algorithms and protocols for wireless sensor networks*, volume 62. John Wiley & Sons, 2008.

[14] Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. *Computer communications*, 29(4):477–489, 2006.

[15] Azzedine Boukerche, Xiuzhen Cheng, and Joseph Linus. Energy-aware data-centric routing in microsensor networks. In *Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pages 42–49. ACM, 2003.

[16] Azzedine Boukerche, Xuzhen Cheng, and Joseph Linus. A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks*, 11(5):619–635, 2005.

[17] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Swimnet: a scalable parallel simulation testbed for wireless and mobile networks. *Wireless Networks*, 7(5):467–486, 2001.

[18] Azzedine Boukerche, Sajal K Das, Alessandro Fabbri, and Oktay Yildiz. Exploiting model independence for parallel pcs network simulation. In *Thirteenth Workshop on Parallel and Distributed Simulation, 1999. Proceedings.*, pages 166–173. IEEE, 1999.

[19] Azzedine Boukerche and Xin Fei. A coverage-preserving scheme for wireless sensor network with irregular sensing range. *Ad hoc Networks*, 5(8):1303–1316, 2007.

[20] Azzedine Boukerche and Xin Fei. A voronoi approach for coverage protocols in wireless sensor networks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 5190–5194. IEEE, 2007.

[21] Azzedine Boukerche, Xin Fei, and Regina B Araujo. An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange. *Computer Communications*, 30(14):2708–2720, 2007.

[22] Azzedine Boukerche, Kathia Regina Lemos Jucá, João Bosco Sobral, and Mirela Sechi Moretti Annoni Notare. An artificial immune based intrusion detection model for computer and telecommunication systems. *Parallel Computing*, 30(5):629–646, 2004.

[23] Azzedine Boukerche and Xu Li. An agent-based trust and reputation management scheme for wireless sensor networks. In *Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE*, volume 3, pages 5–pp. IEEE, 2005.

[24] Azzedine Boukerche, Anahit Martirosyan, and Richard Pazzi. An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks. *Mobile Networks and Applications*, 13(6):614–626, 2008.

[25] Azzedine Boukerche and Mirela Sechi M Annoni Notare. Behavior-based intrusion detection in mobile phone systems. *Journal of Parallel and Distributed Computing*, 62(9):1476–1490, 2002.

[26] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Localization systems for wireless sensor networks. *IEEE Wireless Communications*, 14(6):6–12, 2007.

[27] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Localization systems for wireless sensor networks. *Algorithms and Protocols for Wireless Sensor Networks*, page 307, 2009.

[28] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina B Araujo. Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol. In *The IEEE Conference on Local Computer Networks. 30th Anniversary.*, pages 560–567. IEEE, 2005.

[29] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina Borges Araujo. Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments. *Journal of Parallel and Distributed Computing*, 66(4):586–599, 2006.

[30] Azzedine Boukerche and Yonglin Ren. A trust-based security system for ubiquitous and pervasive computing environments. *Computer Communications*, 31(18):4343–4351, 2008.

[31] Azzedine Boukerche and Yonglin Ren. A secure mobile healthcare system using trust-based multicast scheme. *IEEE Journal on Selected Areas in Communications*, 27(4):387–399, 2009.

[32] Azzedine Boukerche and Samer Samarah. A novel algorithm for mining association rules in wireless ad hoc sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(7):865–877, 2008.

[33] Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad Z Ahmad, Ladislau Bölöni, and Damla Turgut. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13):3032–3080, 2011.

[34] Olutayo Boyinbode, Hanh Le, and Makoto Takizawa. A survey on clustering algorithms for wireless sensor networks. *International Journal of Space-Based and Situated Computing*, 1(2):130–136, 2011.

[35] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

[36] Andre N Campos, Efren L Souza, Fabiola G Nakamura, Eduardo F Nakamura, and Joel JPC Rodrigues. On the impact of localization and density control algorithms in target tracking applications for wireless sensor networks. *Sensors*, 12(6):6930–6952, 2012.

[37] NORMA CHAPMAN. Deer, the animal answer guide. *Zoological Journal of the Linnean Society*, 166(2):464–464, 2012.

[38] Jiming Chen, Kejie Cao, Keyong Li, and Youxian Sun. Distributed sensor activation algorithm for target tracking with binary sensor networks. *Cluster Computing*, 14(1):55–64, 2011.

[39] Wei-Peng Chen, Jennifer C Hou, and Lui Sha. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 3(3):258–271, 2004.

[40] Yunxia Chen and Qing Zhao. On the lifetime of wireless sensor networks. *Communications Letters, IEEE*, 9(11):976–978, 2005.

[41] Peng Cheng, Fan Zhang, Jiming Chen, Youxian Sun, and Xuemin Shen. A distributed tdma scheduling algorithm for target tracking in ultrasonic sensor networks. *IEEE Transactions on Industrial Electronics*, 60(9):3836–3845, 2013.

[42] B Chithra and NJR Muniraj. A mutual information based sensor selection and information controlled transmission power adjustment. In *Optical Imaging Sensor and Security (ICOSS), 2013 International Conference on*, pages 1–10. IEEE, 2013.

[43] Shung Han Cho, Kyung Hoon Kim, and Sangjin Hong. Effective object identification and association by varying coverage through rfid power control. *Journal of Computer Science and Technology*, 29(1):4–20, 2014.

[44] Hung-Chi Chu and Rong-Hong Jan. A gps-less self-positioning method for sensor networks. In *11th International Conference on Parallel and Distributed Systems, 2005. Proceedings.*, volume 2, pages 629–633. IEEE, 2005.

[45] Horacio ABF de Oliveira, Azzedine Boukerche, E Freire Nakamura, and Antonio Alfredo Ferreira Loureiro. An efficient directed localization recursion protocol for wireless sensor networks. *IEEE Transactions on Computers*, 58(5):677–691, 2009.

[46] Fatemeh Deldar and Mohammad Hossien Yaghmaee. Designing a prediction-based clustering algorithm for target tracking in wireless sensor networks. In *2011 International Symposium on Computer Networks and Distributed Systems (CNDS)*, pages 199–203. IEEE, 2011.

[47] Gordana Dodig-Crnkovic. Scientific methods in computer science. In *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Skövde, Suecia*, pages 126–130, 2002.

[48] Arash Nasiri Eghbali, Nastooh Taheri Javan, Amir Dareshoorzadeh, and Mehdi Dehghan. An energy efficient load-balanced multi-sink routing protocol for wireless sensor networks. In *10th International Conference on Telecommunications, 2009. ConTEL 2009.*, pages 229–234. IEEE, 2009.

[49] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. Diagnosing mobile ad-hoc networks: two distributed comparison-based self-diagnosis protocols. In *Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access*, pages 18–27. ACM, 2006.

[50] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. Performance analysis of a distributed comparison-based self-diagnosis protocol for wireless ad-hoc networks. In *Proceedings of the 9th ACM International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pages 165–172. ACM, 2006.

[51] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. A distributed fault identification protocol for wireless and mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 68(3):321–335, 2008.

[52] Kevin Fall and Kannan Varadhan. The ns manual (formerly ns notes and documentation). *The VINT project*, 47, 2005.

[53] Mohsin Fayyaz. Classification of object tracking techniques in wireless sensor networks. *Wireless Sensor Network*, 3:121, 2011.

[54] Xin Fei, Azzedine Boukerche, and Richard Yu. A pomdo based k-coverage dynamic scheduling protocol for wireless sensor networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[55] Roland Flury, Sriram V Pemmaraju, and Roger Wattenhofer. Greedy routing with bounded stretch. In *INFOCOM 2009, IEEE*, pages 1737–1745. IEEE, 2009.

[56] Samir Goel and Tomasz Imielinski. Prediction-based monitoring in sensor networks: taking lessons from mpeg. *ACM SIGCOMM Computer Communication Review*, 31(5):82–98, 2001.

[57] Karthika Gopal and Ramalakshmi Krishnamoorthy. Analysis of cluster based target tracking in wireless sensor networks. *International Journal of Computer Applications (0975–8887) Volume*, 2013.

[58] Lin Gu and John A Stankovic. Radio-triggered wake-up capability for sensor networks. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 27–37, 2004.

[59] Chao Gui and Prasant Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 129–143. ACM, 2004.

[60] Zhen Guo, Mengchu Zhou, and Lev Zakrevski. Optimal tracking interval for predictive tracking in wireless sensor network. *Communications Letters, IEEE*, 9(9):805–807, 2005.

[61] Guangjie Han, Huihui Xu, Trung Q Duong, Jinfang Jiang, and Takahiro Hara. Localization algorithms of wireless sensor networks: a survey. *Telecommunication Systems*, 52(4):2419–2436, 2013.

[62] Alfred O Hero and Douglas Cochran. Sensor management: Past, present, and future. *Sensors Journal, IEEE*, 11(12):3064–3075, 2011.

[63] H Jamali-Rad, B Abolhassani, and M Abdizadeh. Mathematical analysis of optimal tracking interval management for power efficient target tracking wireless sensor networks. *Iranian Journal of Electrical & Electronic Engineering*, 8(3):195, 2012.

[64] Hadi Jamali Rad, Mahdi Azarafrooz, H Shahriar Shahhoseini, and Bahman Abolhassani. A new adaptive power optimization scheme for target tracking wireless sensor networks. In *IEEE Symposium on Industrial Electronics & Applications, 2009. ISIEA 2009.*, volume 1, pages 307–312. IEEE, 2009.

[65] Hadi Jamali-Rad, Andrea Simonetto, and Geert Leus. Sparsity-aware sensor selection: Centralized and distributed algorithms. *Signal Processing Letters, IEEE*, 2014.

[66] Nauman Javed and Tilman Wolf. Multiple object tracking in sensor networks using distributed clique finding. In *2013 International Conference on Computing, Networking and Communications (ICNC).*, pages 1139–1145. IEEE, 2013.

[67] Deepak Jeswani, Ankit Kesharwani, Sneha Chaudhari, Vaishali P Sadaphal, and Ratan K Ghosh. A practical approach for target tracking in sparsely deployed binary sensor network. In *2012 IEEE 20th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS).*, pages 153–160. IEEE, 2012.

[68] Deepak Jeswani, Ankit Kesharwani, Sneha S Chaudhari, Vaishali P Sadaphal, and Ratan K Ghosh. Efficient target tracking through binary-detection in sparsely deployed wsn. In *2011 Third International Conference on Communication Systems and Networks (COMSNETS).*, pages 1–10. IEEE, 2011.

[69] Feng Juan, Baowang Lian, and Zhao Hongwei. Hierarchically coordinated power management for target tracking in wireless sensor networks. *International Journal of Advanced Robotic Systems*, 10, 2013.

[70] B Kalpana and R Sangeetha. A collaborative target tracking framework using particle filter. In *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, pages 1–4. IEEE, 2013.

[71] Gaurav S Kasbekar, Yigal Bejerano, and Saswati Sarkar. Lifetime and coverage guarantees through distributed coordinate-free sensor activation. *IEEE/ACM Transactions on Networking*, 19(2):470–483, 2011.

[72] Natallia Katenka, Elizaveta Levina, and George Michailidis. Tracking multiple targets using binary decisions from wireless sensor networks. *Journal of the American Statistical Association*, 108(502):398–410, 2013.

[73] Wooyoung Kim, Kirill Mechitov, Jeung-Yoon Choi, and Soo Ham. On target tracking with binary proximity sensors. In *Fourth International Symposium on Information Processing in Sensor Networks, 2005. IPSN 2005.*, pages 301–308. IEEE, 2005.

[74] Mikkel Baun Kjærgaard, Sourav Bhattacharya, Henrik Blunck, and Petteri Nurmi. Energy-efficient trajectory tracking for mobile devices. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, pages 307–320. ACM, 2011.

[75] Lasse Klingbeil and Tim Wark. A wireless sensor network for real-time indoor localisation and motion monitoring. In *International Conference on Information Processing in Sensor Networks, 2008. IPSN'08.*, pages 39–50. IEEE, 2008.

[76] Costis Kompis and Simon Aliwell. Energy harvesting technologies to enable remote and wireless sensing. *Sensors and Instrumentation-Knowledge Transfer Network*, 2008.

[77] Santosh Kumar, Ten H Lai, and József Balogh. On k-coverage in a mostly sleeping sensor network. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 144–158. ACM, 2004.

[78] Sung-Min Lee, Hojung Cha, and Rhan Ha. Energy-aware location error handling for object tracking applications in wireless sensor networks. *Computer Communications*, 30(7):1443–1450, 2007.

[79] Hongbin Li, Di Miao, Jiming Chen, Youxian Sun, and Xuemin Shen. Networked ultrasonic sensors for target tracking: an experimental study. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6. IEEE, 2009.

[80] Yanjun Li. Real-time surveillance performance under different sensing models in duty-cycled sensor networks. *Sensors & Transducers, IFSA*, 2013.

[81] Yan Liang, Xiaoxue Feng, Feng Yang, Lianmeng Jiao, and Quan Pan. The distributed infectious disease model and its application to collaborative sensor wakeup of wireless sensor networks. *Information Sciences*, 223:192–204, 2013.

[82] Jianyong Lin, Wendong Xiao, Frank L Lewis, and Lihua Xie. Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks. *IEEE Transactions on Instrumentation and Measurement*, 58(6):1886–1896, 2009.

[83] Guiyun Liu and Bugong Xu. Novel sensor scheduling and energy-efficient quantization for tracking target in wireless sensor networks. *Journal of Control Theory and Applications*, 11(1):116–121, 2013.

[84] Yonggui Liu, Bugong Xu, and Linfang Feng. Energy-balanced multiple-sensor collaborative scheduling for maneuvering target tracking in wireless sensor networks. *Journal of Control Theory and Applications*, 9(1):58–65, 2011.

[85] Stuart MacLean and Suprakash Datta. Energy constrained positioning in mobile wireless ad hoc and sensor networks. *Procedia Computer Science*, 19:321–329, 2013.

[86] Nok Hang Mak and Winston Khoon Guan Seah. How long is the lifetime of a wireless sensor network? In *International Conference on Advanced Information Networking and Applications, 2009. AINA'09.*, pages 763–770. IEEE, 2009.

[87] Goutham Mallapragada, Yicheng Wen, Shashi Phoha, Doina Bein, and Asok Ray. Tracking mobile targets using wireless sensor networks. In *2010 Seventh International Conference on Information Technology: New Generations (ITNG).*, pages 873–878. IEEE, 2010.

[88] Andrew Markham, Niki Trigoni, Stephen A Ellwood, and David W Macdonald. Revealing the hidden lives of underground animals using magneto-inductive tracking. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 281–294. ACM, 2010.

[89] Asis Nasipuri and Kai Li. A directionality based location discovery scheme for wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 105–111. ACM, 2002.

[90] Songhwai Oh, Shankar Sastry, and Luca Schenato. A hierarchical multiple-target tracking algorithm for sensor networks. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.*, pages 2197–2202. IEEE, 2005.

[91] Sundeep Pattem, Sameera Poduri, and Bhaskar Krishnamachari. Energy-quality tradeoffs for target tracking in wireless sensor networks. In *Information Processing in Sensor Networks*, pages 32–46. Springer, 2003.

[92] Bartłomiej Płaczek and Marcin Bernaś. Optimizing data collection for object tracking in wireless sensor networks. In *Computer Networks*, pages 485–494. Springer, 2013.

[93] Wint Yi Poe and Jens B Schmitt. Node deployment in large wireless sensor networks: coverage, energy consumption, and worst-case delay. In *Asian Internet Engineering Conference*, pages 77–84. ACM, 2009.

[94] K Ramya, K Praveen Kumar, and V Srinivas Rao. A survey on target tracking techniques in wireless sensor networks. *International Journal of Computer Science and Engineering Survey*, 3(4):93–108, 2012.

[95] Yonglin Ren and Azzedine Boukerche. Modeling and managing the trust for wireless and mobile ad hoc networks. In *IEEE International Conference on Communications, 2008. ICC'08.*, pages 2129–2133. IEEE, 2008.

[96] Yonglin Ren, Richard Werner Nelem Pazzi, and Azzedine Boukerche. Monitoring patients via a secure and mobile healthcare system. *Wireless Communications, IEEE*, 17(1):59–65, 2010.

[97] Hosam Rowaihy, Sharanya Eswaran, Matthew Johnson, Dinesh Verma, Amotz Bar-Noy, Theodore Brown, and Thomas La Porta. A survey of sensor selection schemes in wireless sensor networks. In *Defense and Security Symposium*, pages 65621A–65621A. International Society for Optics and Photonics, 2007.

[98] Jason Ryder, Brent Longstaff, Sasank Reddy, and Deborah Estrin. Ambulation: A tool for monitoring mobility patterns over time using mobile phones. In *International Conference on Computational Science and Engineering, 2009. CSE'09.*, volume 4, pages 927–931. IEEE, 2009.

[99] Mahmood Salehi and Azzedine Boukerche. Trust-aware opportunistic routing protocol for wireless networks. In *Proceedings of the 10th ACM symposium on QoS and security for wireless and mobile networks*, pages 79–86. ACM, 2014.

[100] Samer Samarah, Muhannad Al-Hajri, and Azzedine Boukerche. A predictive energy-efficient technique to support object-tracking sensor networks. *IEEE Transactions on Vehicular Technology*, 60(2):656–663, 2011.

[101] Andreas Savvides, Heemin Park, and Mani B Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 112–121. ACM, 2002.

[102] Nisheeth Shrivastava, R Mudumbai U Madhow, and S Suri. Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, pages 251–264. ACM, 2006.

[103] Biao Song, Wendong Xiao, and Zhaohui Zhang. Multi-step sensor scheduling for energy-efficient high-accuracy collaborative target tracking in wireless sensor networks. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pages 1341–1345. IEEE, 2013.

[104] Efren Lopes Souza, Eduardo Freire Nakamura, and Horacio Antonio De Oliveira. On the performance of target tracking algorithms using actual localization systems for wireless sensor networks. In *Proceedings of the 12th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 418–423. ACM, 2009.

[105] Leandro Aparecido Villas, Azzedine Boukerche, Heitor S Ramos, Horacio ABF de Oliveira, Regina Borges de Araujo, and Antonio Alfredo Ferreira Loureiro. Drina: A lightweight and reliable routing approach for in-network aggregation in wireless sensor networks. *IEEE Transactions on Computers*, 62(4):676–689, 2013.

[106] Guojun Wang, Md Bhuiyan, Zakirul Alam, and Li Zhang. Two-level cooperative and energy-efficient tracking algorithm in wireless sensor networks. *Concurrency and Computation: Practice and Experience*, 22(4):518–537, 2010.

[107] Xue Wang, Junjie Ma, Sheng Wang, and Daowei Bi. Distributed energy optimization for target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(1):73–86, 2010.

[108] Zhibo Wang, Wei Lou, Zhi Wang, Junchao Ma, and Honglong Chen. A hybrid cluster-based target tracking protocol for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[109] Zijian Wang, Eyuphan Bulut, and Boleslaw K Szymanski. Distributed energy-efficient target tracking with binary sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 6(4):32, 2010.

[110] Greg Welch and Gary Bishop. An introduction to the kalman filter. 2006. *University of North Carolina: Chapel Hill, North Carolina, US*, 2006.

[111] Wendong Xiao, Jiankang Wu, Lihua Xie, and Liang Dong. Sensor scheduling for target tracking in networks of active sensors. *Acta Automatica Sinica*, 32(6):922–928, 2006.

[112] Wendong Xiao, Sen Zhang, Jianyong Lin, and Chen Khong Tham. Energy-efficient adaptive sensor scheduling for target tracking in wireless sensor networks. *Journal of Control Theory and Applications*, 8(1):86–92, 2010.

[113] Guoliang Xing, Chenyang Lu, Robert Pless, and Qingfeng Huang. On greedy geographic routing algorithms in sensing-covered networks. In *Proceedings of the 5th ACM International Symposium on Mobile ad hoc Networking and Computing*, pages 31–42. ACM, 2004.

[114] Yingqi Xu and Wang-Chien Lee. On localized prediction for power efficient object tracking in sensor networks. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pages 434–439. IEEE, 2003.

[115] Yingqi Xu, Julian Winter, and Wang-Chien Lee. Prediction-based strategies for energy saving in object tracking sensor networks. In *2004 IEEE International Conference on Mobile Data Management, 2004. Proceedings.*, pages 346–357. IEEE, 2004.

[116] Hua Yang and Biplab Sikdar. A protocol for tracking mobile targets using sensor networks. In *2003 IEEE International Workshop on Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE.*, pages 71–81. IEEE, 2003.

[117] Fan Zhang, Jiming Chen, Hongbin Li, Youxian Sun, and Xuemin Sherman Shen. Distributed active sensor scheduling for target tracking in ultrasonic sensor networks. *Mobile Networks and Applications*, 17(5):582–593, 2012.

[118] Jian Zhang, Cheng-dong Wu, Yun-zhou Zhang, and Peng Ji. Energy-efficient adaptive dynamic sensor scheduling for target monitoring in wireless sensor networks. *ETRI journal*, 33(6):857–863, 2011.

[119] Shuting Zhang, Guojun Li, Lan Xiao, Linhong Wang, and Xiao-na Zhou. Distributed targets tracking with dynamic power optimization for wireless sensor networks. In *Informatics and Management Science IV*, pages 221–229. Springer, 2013.

[120] Jin Zheng, Md Zakirul Alam Bhuiyan, Shaohua Liang, Xiaofei Xing, and Guojun Wang. Auction-based adaptive sensor activation algorithm for target tracking in wireless sensor networks. *Future Generation Computer Systems*, 2013.

[121] Zongheng Zhou, Samir R Das, and Himanshu Gupta. Variable radii connected sensor cover in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(1):8, 2009.